PNYX: ROBUST COMMUNICATION AND DYNAMIC AUTHORITY FOR COLLABORATIVE VLA AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models provide a language-conditioned interface between visual reasoning and closed-loop control, but are typically deployed as egocentric policies without mechanisms for sharing observations, distributing subtasks, or scheduling around task conflicts. To demonstrate the benefit of collaborative VLA agents, we study VLA-driven multi-agent systems (VLA-MAS), augmenting the conventional dual-system VLM and action expert pairing with a collaborative layer operating between user prompting and the model's existing natural language interface. To enable free-form decentralized planning, we introduce a novel system PNYX, combining a structured, fault-tolerant communication protocol with dynamic task leadership. The improved robustness is realized through a protocol tracking details such as message purpose, freshness, receipt, and agreement, along with the leader-based architecture to temporarily centralize proposal formation while preserving local review, execution, and fallback. We evaluate PNYX across six closed-loop cooperative-driving tasks and show improvement in nominal joint success from 51.2% for the strongest canonical baseline to 58.3%, while increasing route completion from 75.9% to 80.7%. Under severely degraded communication, PNYX reaches 55.4% joint success rate, compared with 42.5% for centralized and 53.8% for free-form decentralized planning. Our work highlights the insight that effective VLA collaboration depends not only on communication volume alone, but also on explicit commitment and network-aware adaptation in information and decision authority.

1 INTRODUCTION

Vision-language-action (VLA) models connect visual perception and language-guided reasoning with physical action (Renz et al., 2025; Wang et al., 2025; Xiong et al., 2026). For instruction-following VLA systems, language provides an interface through which a planner can specify goals and constraints without modifying the execution policy. This interface, however, does not by itself coordinate independently controlled agents. In cooperative driving, one vehicle may observe a hazard occluded from another, while merging or overtaking requires vehicles to agree on who proceeds and who yields. Collaboration therefore requires both sharing missing information and assigning an order to conflicting actions.

We study VLA-driven multi-agent systems (VLA-MAS) by adding a collaboration system above the dual-system pairing of a vision-language model and an action expert. Building on language-mediated embodied planning (Chen et al., 2024; Mandi et al., 2024), this system receives task prompts, exchanges information among agents, and produces coordinated instructions for their local VLAs. Collaboration operates through the existing language interface without requiring modification or collaborative fine-tuning of the execution policies.

We instantiate three baseline organizations drawn from embodied multi-agent planning (Chen et al., 2024): independent planning without inter-agent communication, centralized planning in which one shared planner assigns all agents' actions, and free-form decentralized planning in which local planners exchange unconstrained natural-language messages. Our analysis identifies two challenges (Sec. 4). First, sharing observations and intentions does not reliably establish who should act or which assignment participants have accepted. This finding concerns the evaluated free-form dialogue, not decentralized planning in general. Second, coordination must complete before the physical task's

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.

decision deadline. Centralized planning depends on communication with one fixed planner, while free-form dialogue can leave peers with different assumptions about agreement. Limited bandwidth restricts message content, and delay and packet loss can prevent agreement before the deadline.

To address both challenges, we introduce PNYX[1], a unified coordination system with two coupled mechanisms. The first is a *structured, deadline-bounded communication protocol*. It adapts FIPA-ACL message typing to natural-language payloads, separating information sharing from action negotiation and message receipt from proposal acceptance. Each joint proposal assigns an action to every participant and requires their explicit acceptance before commitment. To handle limited and unreliable communication, PNYX prioritizes facts within a message budget, tracks observation freshness, and bounds retransmission by an attempt limit and coordination deadline. An expired exchange returns agents to local execution rather than blocking them indefinitely.

The second mechanism is *deterministic, task-scoped leader election*. PNYX assigns proposal formation to one participating agent while retaining local review and execution for all participants. The election favors the task initiator, then the candidate with fresher information, using a stable identifier to break ties rather than language-model deliberation. Terms, heartbeats, and replacement mechanisms adapted from Raft allow leadership to change when the leader becomes unavailable or its state becomes stale. Together, these mechanisms establish one source of joint proposals without permanent dependence on a fixed planner or allowing a leader to bypass participant acceptance (Sec. 5).

We evaluate PNYX on six cooperative-driving scenarios from an extended TalkingVehicles benchmark (Cui et al., 2026) in CARLA (Dosovitskiy et al., 2017). Each architecture–condition cell contains 240 episodes pooled across two planners, with a fixed VLA-emulation executor to isolate collaborative planning. Nominal, mild, and severe profiles jointly vary communication and sensing conditions, including bandwidth, latency, packet loss, sensing noise, and range (Sec. 6.1). PNYX achieves the highest aggregate joint success and route completion among the evaluated architectures at all three degradation levels (Sec. 6). Under nominal communication, joint success—all focal vehicles succeeding without collision—reaches 58.3%, compared with 51.2% for both centralized and free-form decentralized planning. Mean route completion reaches 80.7%, compared with 75.9% for centralized planning, the strongest baseline on this metric. Under severe degradation, joint success remains 55.4%, compared with 42.5% for centralized and 53.8% for free-form decentralized planning.

The contributions of this work are as follows:

- We formulate VLA-MAS with an explicit collaboration system above language-conditioned execution and analyze the communication, computation, and authority tradeoffs of three baseline organizations.
- We develop the PNYX protocol, combining typed natural-language exchanges and explicit proposal acceptance with payload budgets, freshness tracking, and bounded retransmission.
- Within the same system, we introduce deterministic task-scoped leader election and management, preserving participant review while supporting failure detection and replacement.
- We evaluate PNYX across six closed-loop driving scenarios, two planners, and three degradation profiles, with ablations of state content, proposal rejection, and leader selection.

2 RELATED WORK

Vision-Language-Action Models. Vision-language-action models connect visual reasoning and language-conditioned planning to robot control. Generalist policies such as π_0 operate across manipulation tasks and embodiments (Black et al., 2024), while driving models generate closed-loop actions or trajectories. SimLingo aligns language with vision-only driving (Renz et al., 2025); Alpamayo-R1 couples causal reasoning with diffusion-based trajectory generation (Wang et al., 2025); and ReCogDrive separates cognitive reasoning from trajectory planning (Xiong et al., 2026). Dual-system designs similarly separate semantic planning from fast action generation (Cui et al., 2025; Tan et al., 2025). Most of these systems remain centered on one ego agent, while joint-action policies generally control multiple effectors within one model rather than independently deployed

[1] PNYX (pronounced "niks"; Ancient Greek: *Pnýx*) is named after the hill in Athens where the Athenian democratic assembly met.

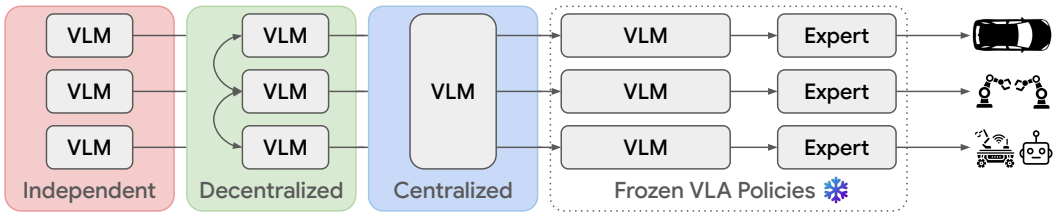

Figure 1: Comparison of the three multi-agent VLA coordination architectures.

agents with private observations and intermittent communication. CHORUS studies decentralized execution with a shared VLA policy but no inference-time communication (Doshi et al., 2026). Other work shows that correct language-level reasoning may not produce the intended low-level action and improves frozen policies through runtime action selection (Wu et al., 2025; Nakamoto et al., 2024). We therefore hold the execution interface fixed to isolate multi-agent planning and communication from action decoding. Under this abstraction, existing VLA work provides the language-conditioned action interface but does not specify how independently deployed policies should share private state, assign interdependent actions, or transfer decision authority.

Embodied Multi-Agent Planning. LLM-based systems decompose tasks, allocate roles, exchange observations, and revise multi-robot plans. RoCo combines agent dialogue with environment and motion-planning feedback (Mandi et al., 2024), while CoELA integrates communication with modular perception, memory, planning, and execution (Zhang et al., 2024); other systems use centralized planning and feedback for heterogeneous or long-horizon tasks (Liu et al., 2025; Nayak et al., 2024). Chen et al. (Chen et al., 2024) compare centralized, decentralized, and hybrid organizations, showing that planner topology affects success and token efficiency as team size increases. Additional dialogue is not always beneficial: goal-oriented or calibrated communication can reduce redundant messages (Ying et al., 2024; Zhang et al., 2026), and dialogue may align agents' world models without improving task success (Dongre & Hakkani-Tür, 2026). Work on communication-restricted robot teams further studies intermittent connectivity and deviations from shared plans (Bramblett et al., 2023; Zhang et al., 2023). These systems address organizational questions largely absent from single-agent VLA research, but often assume predefined skills or collaboration-specific executors and rarely evaluate controlled degradation of inter-agent communication. PNYX connects the two settings by coordinating otherwise unmodified, black-box language-conditioned executors through typed inter-agent messages and dynamic task leadership, while directly varying message loss, latency, corruption, and range.

3 PRELIMINARIES

We consider N embodied agents, each assigned a local objective and controlled through an instruction-following vision-language-action system. We extend the VLA with an additional VLM planning layer, which receives the task prompt and performs task assignment, orchestration, and scheduling to decompose the long-horizon task into individual commands for each agent VLA. Fig. 1 shows the three fundamental multi-agent architectures as presented in embodied LLM literature (Chen et al., 2024). We present these three architectures as our baseline, and extend them to use end-to-end VLA policies for execution in lieu of hard-coded control. *Independent planning* uses one local planner per agent without communication, serving as our non-collaborative baseline. *Centralized planning* places team state and assignment authority in one shared planner, but makes every participant dependent on that planner and its communication paths. *Free-form decentralized planning* retains local planners and allows unconstrained peer dialogue.

4 CHALLENGES

We compare the architectures from Sec. 3 to examine two barriers to collaborative planning: dependence on a fixed planner and failure to convert shared observations into compatible actions. Sec. 6 reports the controlled experiments, and Appendix A analyzes scaling and communication dependencies. Centralized and free-form decentralized planning achieve the same nominal joint success but form decisions differently (Fig. 3). Centralized planning colocates task-relevant information and au-

thority to produce one complete assignment. Free-form decentralized planning instead requires each local planner to infer whether peer information is current, whether a statement is an intention or an accepted proposal, and when to act. Aggregate success does not distinguish scene misunderstanding from failure to form a compatible joint decision; the trace analysis below examines this distinction. Centralized and decentralized planning also expose different dependencies to communication failure. Centralized planning requires observations and joint assignments to traverse a fixed planner's communication paths. Free-form decentralized planning preserves local execution, but delay, loss, and truncation can leave agents with different dialogue histories and assumptions about agreement. Local execution therefore removes dependence on a shared planner without ensuring consistent joint decisions. The degraded profiles jointly perturb communication and sensing (Sec. 6.3); their non-monotonic results do not isolate the effects of communication loss or local fallback. Across 2,539 free-form messages, 99.1% state the sender's intent, and communicated scene descriptions agree with the sender's sensed observations, yet only 0.8% explicitly assign precedence. Agreement with local observations does not establish a correct shared scene interpretation, but the traces expose a separate failure: agents announce intentions without establishing one accepted joint assignment. Across the three negotiation scenarios, decentralized planning produces compatible GO/YIELD assignments in 22% of decision cycles and instructs both agents to proceed in 76%; centralized planning reaches 63% compatibility, while independent planning reaches 37%. These measurements concern planner instructions, not executed actions. In 53% of decentralized negotiation episodes, both planners claim within the same two-second window that the other has yielded, and instructions reverse between GO and YIELD an average of 2.3 times per agent-episode. Observations, intentions, proposals, and accepted assignments share one free-form representation, leaving commitment to be inferred from prose. For an extended dialogue trace illustrating this phenomenon, see App. C. These findings motivate explicit message purpose and freshness, a distinction between receipt and acceptance, and one joint assignment accepted by all participants. Task-scoped proposal authority that can be reassigned addresses dependence on a fixed planner. Sec. 5 develops these mechanisms within PNYX while leaving the executor unchanged.

5 METHOD

PNYX addresses the challenges in Sec. 4 through two layers above an unmodified VLA system: a structured communication protocol (Sec. 5.1) and dynamic task leadership (Sec. 5.2). The protocol governs information sharing and agreement; the leadership mechanism assigns proposal formation to one agent through deterministic election, with failure detection and replacement.

5.1 STRUCTURED COMMUNICATION PROTOCOL

PNYX adopts FIPA-ACL communicative-act typing (Poslad, 2007) with natural-language payloads to distinguish information sharing from action negotiation. Truncated automatic repeat request (ARQ) (Liu et al., 2004) bounds retransmission of unacknowledged messages, while age-of-information (AoI) accounting (Yates et al., 2021) tracks observation freshness. Commitment requires every participant to accept the same proposal. Exchanges run asynchronously over AutoGen (Wu et al., 2024b), which provides message addressing and transport. Fig. 2 summarizes the protocol state machine.

Act types and exchange protocol. The protocol for exchanges between agents evolves along one of two paths. Agents may engage in an *information-sharing* exchange, where the goal is to request and obtain information outside of the ego agent's observation space, or they may engage in an *action-negotiation* exchange, where the goal of the interaction is to derive a joint action plan to be executed by each agent in parallel. Interactions begin with a common QUERY-REF act type, which may request either an action proposal or information from another specified agent. The remaining act type set consists of INFORM, PROPAGATE, PROPOSE, ACCEPT-PROPOSAL, and REJECT-PROPOSAL, with varying availability depending on current exchange state. Each message $m = (h, x)$ consists of a header h and act-specific content x. We define the header as

$$h = (\kappa, s, r, c), \tag{1}$$

where κ is act type, s and r identify the sender and intended receiver, and c is a globally unique conversation identifier that associates all messages belonging to the same exchange.

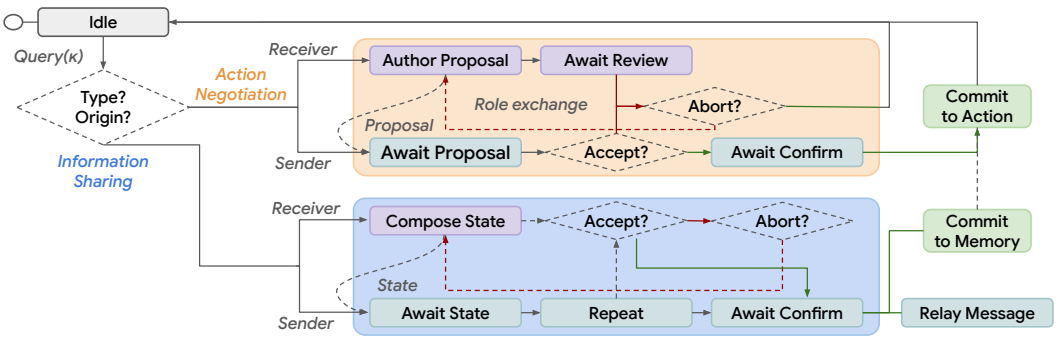

Figure 2: The PNYX$_{\text{Proto}}$ protocol from one agent's perspective. Information sharing records source and freshness before state enters memory. Action negotiation releases a joint assignment only after all participants accept the same proposal and complete confirmation. Rejection permits bounded revision; timeout returns each agent to local execution.

Information sharing. When a planner's local observation is insufficient, it sends a QUERY-REF whose content identifies the desired information. The recipient answers with an INFORM response that shares the most relevant current observation. When the complete answer would exceed the message budget, the responder ranks candidate facts by relevance and urgency and transmits the highest-ranked subset that fits within the allocated budget. An agent may additionally use PROPAGATE to relay an INFORM to all additional participants within range, thus improving performance under communication range limitations. The embedded content retains the original source and observation time so that forwarding does not obscure information freshness.

Action negotiation and commitment. Action negotiation exchanges begin when a QUERY-REF is issued requesting a joint-action proposal from a designated agent. The queried agent becomes the proposer and responds with PROPOSE, where the content specifies the complete joint action plan, and a proposal identifier. Each participant in the exchange evaluates its proposed action against its local observation, task objective, and execution constraints, then returns ACCEPT-PROPOSAL for that exact proposal or REJECT-PROPOSAL with a description of the limiting constraint. A rejection may additionally suggest a corrected assignment, but that assignment enters the protocol as a new PROPOSE. The roles of proposer and reviewer cycle in a round-robin form, within a set number of rounds. A given action proposal is only committed and executed when

$$\text{Commit}(z) = \bigwedge_{i \in \mathcal{C}_q} A_i(z), \tag{2}$$

where $A_i(z)$ indicates that participant i sent ACCEPT-PROPOSAL for the proposal identifier z. Once this condition holds, the leader uses INFORM to announce the committed assignment and common execution time to the participants. A rejection or expired deadline discards the uncommitted proposal, so every negotiation exchange terminates as committed, rejected, or expired.

Bounded retransmission. To improve robustness to partial and total message loss, PNYX applies truncated ARQ (Liu et al., 2004) beneath the communicative-act layer. The sender retains each outgoing message and, if delivery is not acknowledged within a given timeout, retransmits the same message up to a set maximum number of attempts. Retransmission terminates when delivery is acknowledged, the attempt limit is reached, or the coordination deadline expires, preventing an unavailable peer or stale message from blocking the exchange indefinitely. This transport-level acknowledgement confirms receipt only and confirmation of proposed action still requires issuance of ACCEPT-PROPOSAL.

5.2 DYNAMIC LEADER-BASED COORDINATION

Leader-based organization has been previously established as an effective technique in embodied LLM planning. HMAS-2 (Chen et al., 2024) uses a central planner whose assignments are reviewed by local agents, CAMON (Wu et al., 2024a) transfers leadership to the agent that requests coordination, and DynaHMRC (Yu et al., 2026) elects a leader through generated campaign statements and voting.

PNYX retains the core aspect of having a single, authoritative agent in a distributed multi-agent system but removes language-model deliberation from leader selection. Instead, it chooses leadership using a deterministic rank over task and communication state, then manages leader tenure using terms, heartbeats, and replacement mechanisms adapted from the Raft consensus algorithm (Ongaro & Ousterhout, 2014).

Leader-based exchange protocol. Our leader-based exchange protocol primarily modifies the original protocol with an asymmetric variant which restricts permission to send PROPOSE act types based on agent role. For each task q, the active agents form a fixed participant set $\mathcal{C}_q$. A follower that requires a coordinated action sends a QUERY-REF requesting a joint-action proposal from the current leader. The leader gathers missing participant state through the information-sharing protocol and responds with PROPOSE, whose content assigns one action to every participant. The commitment condition in Eq. 2 remains unchanged: the leader cannot release the proposal unless every affected participant accepts it.

Deterministic leader election. An election begins when a task has no active leader or when the current leader is unavailable. Election transaction z_{e} opens a new task term, and each participant shares an INFORM containing its candidate rank

$$\rho_i = (r_i, -a_i, \mathrm{ID}_i), \tag{3}$$

where r_i indicates whether planner i initiated the task, a_i is the age of its newest task-relevant state, and ID_i is a stable tie-breaker. The rank first favors the task initiator, then the candidate with fresher information, and finally the fixed identifier. The age term reuses the AoI measure applied during information sharing, so the same quantity that determines whether state is usable also contributes to the selection of the agent that will aggregate that state. After collecting one candidate record from every participant, each agent independently computes

$$\ell^* = \arg\max_{i \in \mathcal{C}_q} \rho_i. \tag{4}$$

Because all agents apply the same rule to the same candidate set, leader selection does not require a model vote or negotiation round. Each participant then sends an INFORM containing its vote for ℓ^* and a digest of the task, participant set, election transaction, and candidate records. The selected candidate becomes leader only after receiving matching votes from every participant; the complete vote set forms the election certificate $\Gamma_{z_{\mathrm{e}}}$. Missing candidate records, inconsistent digests, or an incomplete certificate terminate the election in local fallback.

Failure detection and handoff. The leader periodically sends an INFORM heartbeat containing its identifier, task term, latest proposal, and timestamp. Followers suspect leader loss after a set number of missed heartbeat deadlines. Replacement may also begin when the leader's shared state becomes stale, its proposals are repeatedly rejected, or a follower requests replacement after the minimum leader tenure.

6 RESULTS

6.1 EXPERIMENT SETUP

We evaluate our framework in CARLA 0.9.14 (Dosovitskiy et al., 2017) on an extended TalkingVehicles (Cui et al., 2026) benchmark featuring six multi-agent driving scenarios under randomized weather, lighting, and vehicle appearances, with identical seed-paired routes across conditions. The benchmark splits into three *negotiation* scenarios (highway exit, merge, and overtake), where two focal vehicles are in direct physical conflict, and three *collaborative-perception* scenarios (unprotected left turn, occluded overtake, and red-light violation), which pair a focal vehicle with an occluded stationary sensor truck.

To separate model capacity from system design, we test two VLM planners operating at 2 Hz: Qwen3.5-27B and Gemma-4-E4B. Each produces scheduled high-level natural-language instructions for a fixed, closed-loop executor. Because current VLAs struggle with instruction following, particularly under distribution shift (Wu et al., 2025; Nakamoto et al., 2024), we use a controlled VLA emulation to isolate collaborative planning: a frozen Gemma vision-language model maps the

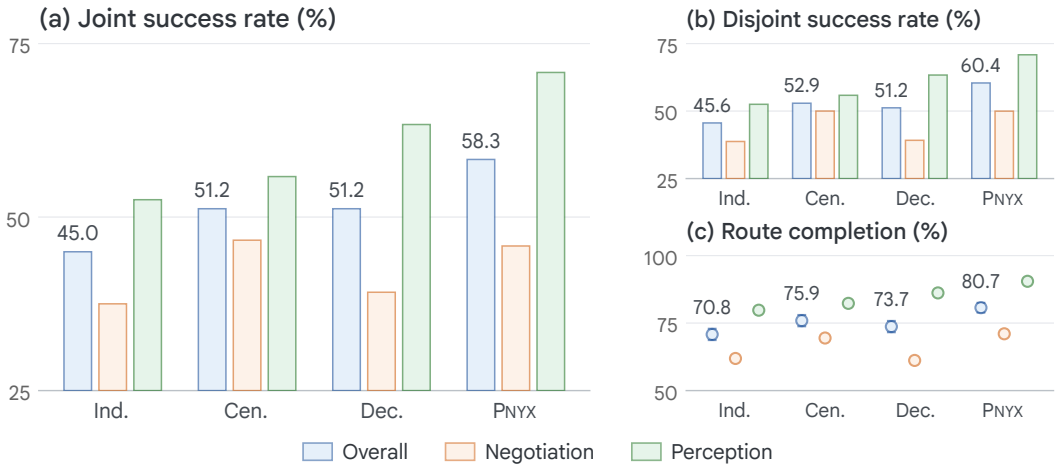

Figure 3: Nominal performance overall and by task group. Results pool both planners (n=240 episodes per architecture) on seed-paired routes and initial conditions. Negotiation tasks use joint/disjoint success; collaborative-perception tasks use focal-vehicle success. Overall success pools both task groups. Route completion is mean ± standard error. PNYX uses full STATE with REJECT available.

current observation and planner instruction to an atomic command from the TalkingVehicles action set, which the existing command follower executes at higher frequency. The executor, command vocabulary, sensing configuration, and controller are fixed across all architectures and communication conditions, so only the information and decision authority available to the planning layer change.

Inter-agent communication is managed asynchronously via AutoGen (Wu et al., 2024b). We report performance across five primary dimensions: (1) *joint success rate*, the percentage of episodes where all focal vehicles succeed without collision; (2) *disjoint success rate*, the percentage where at least one focal vehicle succeeds; (3) continuous *route completion percentage*; (4) *route efficiency*, the elapsed simulated time for vehicles that complete their route; and (5) *coordination overhead*, measured through planner prompt tokens, completion tokens, and call rates. Unless otherwise stated, each architecture–condition cell contains 240 episodes pooled over both planners, with 10 risky and 10 randomized trials per scenario and planner. System resilience is evaluated under nominal, mild, and severe network conditions, where we vary bandwidth, latency, packet loss, sensing noise, and range.

6.2 PERFORMANCE UNDER NOMINAL COMMUNICATION

PNYX improves aggregate joint success and route completion over both centralized and free-form decentralized planning (Fig. 3). It reaches 58.3% joint success, compared with 51.2% for both baselines and 45.0% for independent planning. Route completion reaches 80.7%, compared with 75.9% for the strongest baseline, centralized planning. The gains over the strongest baselines are therefore 7.1 and 4.8 percentage points, respectively.

The improvement is not uniform across scenarios. PNYX exceeds all baselines in joint success at highway exit, unprotected left turn, and red-light violation, and matches the strongest baseline at merge. Negotiation overtake remains difficult: joint/disjoint success reaches 10.0/21.2%, compared with 5.0/5.0% for free-form decentralized planning, but joint success remains below centralized planning. The aggregate gain is consistent with the failure analysis in Sec. 4: PNYX assigns one proposer and requires acceptance of a specific joint assignment rather than leaving agreement implicit in dialogue. This comparison evaluates those mechanisms together, not the effect of commitment alone.

6.3 ROBUSTNESS TO COMMUNICATION DEGRADATION

PNYX retains the highest aggregate joint and disjoint success under both degraded profiles (Fig. 4). Joint/disjoint success is 59.6/61.9% under mild degradation and 55.4/59.0% under severe degradation; route completion remains above 80% at every level. Under severe degradation, its joint-success

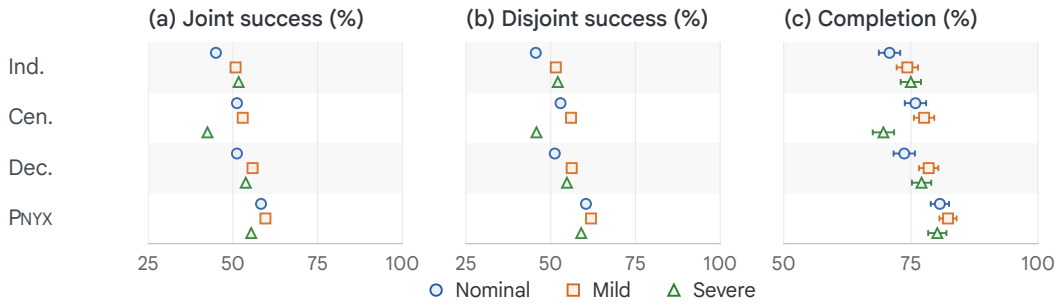

Figure 4: Aggregate performance under nominal, mild, and severe degradation (n=240 episodes per architecture–condition cell, pooled across planners). Profiles jointly vary communication and sensing conditions. PNYX uses full STATE with REJECT available throughout; the severe configuration additionally includes grounded assertions in election candidacy messages. Route completion is mean $\pm$ standard error.

advantage is 12.9 points over centralized planning, 1.6 over free-form decentralized planning, and 3.7 over independent planning. From nominal to severe conditions, joint success decreases by 2.9 points for PNYX and 8.7 points for centralized planning.

A plausible explanation for this smaller decrease is that PNYX bounds retransmission and agreement by a coordination deadline and retains local execution when an exchange fails, whereas centralized planning depends on communication with one shared planner. These results do not isolate those mechanisms: the profiles jointly perturb communication and perceived state, and independent and decentralized planning both improve relative to nominal conditions. Moreover, the severe PNYX configuration includes grounded assertions in election candidacy messages. The comparison therefore measures the configured system under each profile, not the effect of channel degradation alone or the benefit of leader replacement.

6.4 ABLATION STUDIES

Within PNYX, we vary three design choices: full versus semantic STATE, availability of REJECT, and ranked election versus a random leader fixed at task start. The protocol study crosses the two payloads with rejection enabled or disabled while retaining ranked election. The election study holds full STATE and rejection availability fixed. Each cell contains 240 episodes pooled across planners, using seeds disjoint from the main evaluation.

6.4.1 PROTOCOL ABLATION

Full STATE improves average performance, whereas enabling REJECT provides no consistent gain (Fig. 5). Full state transmits grounded assertions in relevance order, followed by sensed text and images as the message budget permits; semantic state transmits planner-selected assertions and a one-sentence summary. Across paired PNYX configurations, full state improves joint success by 3.1 points and route completion by 2.6 points overall. The differences are concentrated in nominal and mild conditions; under severe degradation, joint success is approximately 51–52% for all four configurations. The severe 436 B message budget limits full state to approximately ten assertions without images, so "full" denotes the payload policy, not an unbounded observation transfer.

Enabling rejection permits a participant to return a conflicting local constraint for proposal revision, but the results do not establish an average benefit from that additional exchange. This ablation tests rejection availability, not the requirement to accept a proposal before commitment. We use full state for its average gain and retain rejection as a revision path, without attributing a success improvement to it. The boxed configuration in Fig. 5 is used in both main comparisons (Figs. 3 and 4).

6.4.2 LEADER-ELECTION ABLATION

Ranked election outperforms random leader assignment under nominal and mild conditions, but not under severe degradation (Table 6). Its pooled gains in joint success/route completion are 3.3/2.9 points nominally and 6.2/3.8 points under mild degradation. The gains are larger for Qwen3.5-27B;

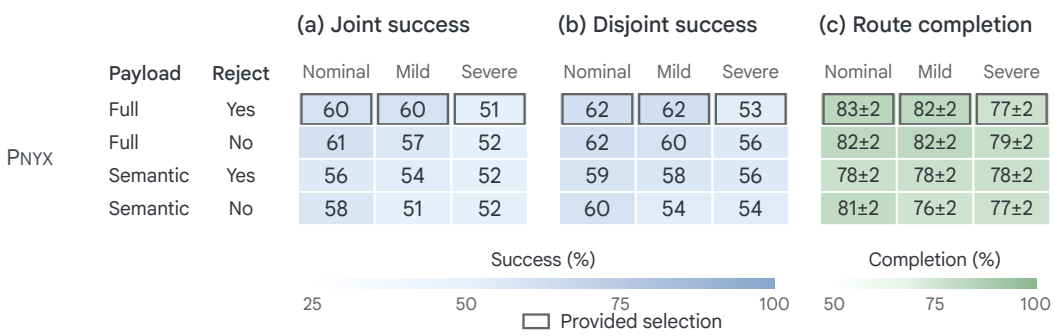

Figure 5: Payload and rejection ablation for PNYX with ranked election. Each cell pools 240 episodes across both planners on seeds disjoint from the main evaluation. Panels report joint success, disjoint success, and route completion (%); completion includes standard error. Boxes mark full STATE with REJECT available, the payload and review configuration used in the main evaluation.

Gemma-4-E4B has unchanged nominal joint success and a 2.5-point increase under mild degradation. These results favor the task-initiator-first election over random assignment in the first two profiles, but do not establish a uniform benefit across planners and conditions.

Under severe degradation, ranked election reaches 50.8% joint success and 77.0% route completion, below the random baseline's 56.7% and 79.4%. The random baseline receives its leader assignment before execution and sends no election messages, so this comparison changes both leader selection and startup communication. We hypothesize that exchanging candidacy and vote records at 16 kbps with high delay and loss delays the first proposal or causes election timeout. Establishing this explanation requires election latency and timeout measurements; the success rates alone do not distinguish startup cost from leader choice. Because the random leader is fixed and provides no failure detection or replacement, the comparison also does not isolate the benefit of leadership handoff.

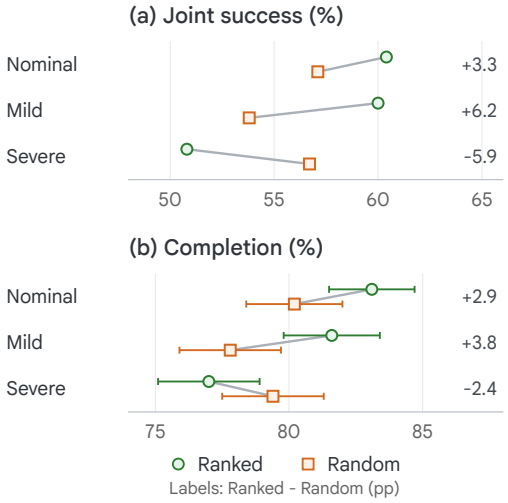

Figure 6: Leader-election ablation for PNYX: ranked election versus a seeded random leader fixed at task start without election messages. Joint success / route completion (%); n=120 per planner and cell, n=240 pooled, on ablation seeds. Both use full STATE with REJECT available.

7 CONCLUSION

We study how planner organization and communication determine whether VLA agents convert complementary observations into coordinated actions. PNYX combines a structured asynchronous protocol with dynamic task leadership to form explicit joint commitments within communication budgets and deadlines. The protocol distinguishes receipt from proposal acceptance, while task-scoped leadership assigns proposal formation to one agent without permanent dependence on a fixed planner. Each participant retains local review, execution, and fallback. Across six closed-loop cooperative-driving tasks, it achieves the highest aggregate joint success and route completion among the evaluated architectures under nominal, mild, and severe degradation. Future work will evaluate larger agent counts, household and industrial tasks involving mobile robots and manipulators, and fine-tuned VLA models as alternative executors.

DISCLOSURE OF AI USE

In this work, we used generative AI tools to polish the writing and to assist with debugging, editing, and understanding existing code. We have reviewed the AI-assisted text and checked the affected code. We take responsibility for the final content of the paper and accompanying code.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/html/2410.24164v1.

Lauren Bramblett, Shijie Gao, and Nicola Bezzo. Epistemic prediction and planning with implicit coordination for multi-robot teams in communication restricted environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5744–5750, 2023. doi: 10.1109/ICRA48891.2023.10161553. URL https://arxiv.org/abs/2302.10393.

Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas Roy, and Chuchu Fan. Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4311–4317. IEEE, 2024. URL https://arxiv.org/abs/2309.15943.

Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, et al. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. *arXiv preprint arXiv:2505.03912*, 2025. URL https://arxiv.org/abs/2505.03912.

Jiaxun Cui, Chen Tang, Jarrett Holtz, Janice Nguyen, Alessandro G. Allievi, Hang Qiu, and Peter Stone. CoopReflect: Towards natural language communication for cooperative autonomous driving via multi-agent learning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026)*, Paphos, Cyprus, 2026. IFAAMAS. doi: 10.65109/MOAV6406. URL https://arxiv.org/abs/2505.18334.

Vardhan Dongre and Dilek Hakkani-Tür. Embodied multi-agent coordination by aligning world models through dialogue. In *Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 293–306, Atlanta, Georgia, USA, August 2026. Association for Computational Linguistics. URL https://arxiv.org/abs/2605.12920.

Ria Doshi, Tian Gao, Annie Chen, Chelsea Finn, and Jeannette Bohg. Chorus: Decentralized multi-embodiment collaboration with one vla policy. *arXiv preprint arXiv:2606.12352*, 2026. URL https://arxiv.org/abs/2606.12352.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017. URL https://arxiv.org/abs/1711.03938.

Kehui Liu, Zixin Tang, Dong Wang, Zhigang Wang, Xuelong Li, and Bin Zhao. Coherent: Collaboration of heterogeneous multi-robot system with large language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10208–10214, 2025. doi: 10.1109/ICRA55743.2025.11127808. URL https://arxiv.org/abs/2409.15146.

Qingwen Liu, Shengli Zhou, and G.B. Giannakis. Cross-layer combining of adaptive modulation and coding with truncated arq over wireless links. *IEEE Transactions on Wireless Communications*, 3 (5):1746–1755, 2004. doi: 10.1109/TWC.2004.833474. URL https://ieeexplore.ieee.org/document/1343909.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 286–299, 2024. doi: 10.1109/ICRA57147.2024.10610855. URL https://arxiv.org/abs/2307.04738.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *Conference on Robot Learning*, 2024. doi: 10.48550/arXiv.2410.13816. URL https://arxiv.org/abs/2410.13816.

Siddharth Nayak, Adelmo Morrison Orozco, Marina Ten Have, Jackson Zhang, Vittal Thirumalai, Darren Chen, Aditya Kapoor, Eric Robinson, Karthik Gopalakrishnan, James Harrison, et al. Long-horizon planning for multi-agent robots in partially observable environments. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2407.10031.

Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pp. 305–319, Philadelphia, PA, June 2014. USENIX Association. ISBN 978-1-931971-10-2. URL https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro.

Stefan Poslad. Specifying protocols for multi-agent systems interaction. *ACM Trans. Auton. Adapt. Syst.*, 2(4):15–es, November 2007. ISSN 1556-4665. doi: 10.1145/1293731.1293735. URL https://dl.acm.org/doi/10.1145/1293731.1293735.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2503.09594.

Huajie Tan, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Yaoxu Lyu, Mingyu Cao, Zhongyuan Wang, and Shanghang Zhang. Roboos: A hierarchical embodied framework for cross-embodiment and multi-agent collaboration. *arXiv preprint arXiv:2505.03673*, 2025. URL https://arxiv.org/abs/2505.03673.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025. URL https://arxiv.org/abs/2511.00088.

Pengying Wu, Yao Mu, Kangjie Zhou, Ji Ma, Junting Chen, and Chang Liu. Camon: Cooperative agents for multi-object navigation with llm-based conversations. *arXiv preprint arXiv:2407.00632*, 2024a. URL https://arxiv.org/abs/2407.00632.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang (Eric) Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Ahmed Awadallah, Ryen W. White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. In *COLM 2024*, August 2024b. URL https://arxiv.org/abs/2308.08155.

Yilin Wu, Anqi Li, Tucker Hermans, Fabio Ramos, Andrea Bajcsy, and Claudia Perez-D'Arpino. Do what you say: Steering vision-language-action models via runtime reasoning-action alignment verification. *arXiv preprint arXiv:2510.16281*, 2025. doi: 10.48550/arXiv.2510.16281. URL https://arxiv.org/abs/2510.16281.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026. URL https://arxiv.org/abs/2506.08052.

Roy D. Yates, Yin Sun, D. Richard Brown, Sanjit K. Kaul, Eytan Modiano, and Sennur Ulukus. Age of information: An introduction and survey, 2021. URL https://arxiv.org/abs/2007.08564.

Lance Ying, Kunal Jha, Shivam Aarya, Joshua B Tenenbaum, Antonio Torralba, and Tianmin Shu. Goma: Proactive embodied cooperative communication via goal-oriented mental alignment. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7099–7106. IEEE, 2024. URL https://arxiv.org/abs/2403.11075.

Wenhao Yu, Yu'ang Xie, Yifan Duan, Jie Peng, Guanting Ye, Ka-Veng Yuen, Yanyong Zhang, and Jianmin Ji. Dynahmrc: Decentralized heterogeneous multi-robot collaboration for dynamic tasks with large language models. *arXiv preprint arXiv:2606.14882*, 2026. URL https://arxiv.org/abs/2606.14882.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *International Conference on Learning Representations*, volume 2024, pp. 19373–19401, 2024. URL https://arxiv.org/abs/2307.02485.

Qingzhao Zhang, Xumiao Zhang, Ruiyang Zhu, Fan Bai, Mohammad Naserian, and Z. Morley Mao. Robust real-time multi-vehicle collaboration on asynchronous sensors. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399906. doi: 10.1145/3570361.3613271. URL https://doi.org/10.1145/3570361.3613271.

Xiaopan Zhang, Zejin Wang, Zhixu Li, Jianpeng Yao, and Jiachen Li. Commcp: Efficient multi-agent coordination via llm-based communication with conformal prediction. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026. URL https://arxiv.org/abs/2602.06038.

A Architecture Analysis

The three canonical architectures differ in where they place communication dependency, planner computation, context, and decision authority. Table 1 summarizes their nominal scaling for N agents, where R is the number of dialogue rounds, $|\mathcal{E}|$ is the number of directed peer links, s is the bounded context for one agent, m is the mean peer-message length, and $d \leq N-1$ is the mean communication degree. These relations characterize structural resource requirements and failure exposure rather than task success, which also depends on the planner, executor, embodiment, and environment.

Table 1: Nominal architecture scaling.

Architecture	Calls	Deliveries	Context
Ind.	N	0	$\Theta(s)$
Cen.	1	$2N$	$\Theta(Ns)$
Dec.	NR	$R\|\mathcal{E}\|$	$\Theta(s+Rdm)$

Communication dependency. Independent planning requires no inter-agent communication. Centralized planning requires each agent to send its state to the shared planner and receive an assignment, giving $2N$ logical deliveries per planning step. If these deliveries are independent, homogeneous, and all required before a common deadline, the probability that the complete round arrives on time is $(p_\Delta)^{2N}$, where p_Δ is the timely-delivery probability for one link. This expression describes communication exposure rather than task success, because the centralized planner may still act on partial or stale inputs. Free-form decentralized planning exchanges $R|\mathcal{E}|$ messages over R rounds. Its traffic is $O(RN^2)$ under all-to-all communication and $O(RN)$ when each agent communicates with a bounded number of neighbors. A failed decentralized link can separate the beliefs or commitments of the affected agents without making the remaining planners unavailable.

Context and inference scaling. Centralized planning uses one planner call per step, but its context grows as $\Theta(Ns)$ because one model must represent the participating agents and produce their joint assignment. Independent planning uses N calls with $\Theta(s)$ context per planner. Free-form decentralized planning distributes computation across N local contexts but may require up to NR calls and $\Theta(s+Rdm)$ context per planner. It therefore avoids concentrating the complete team state in one context only when dialogue rounds, message length, and communication degree remain bounded. Under all-to-all communication, $d = N-1$, and aggregate received context grows as $\Theta(RN^2m)$. These expressions are asymptotic resource relations rather than latency estimates; batching, caching, parallel inference, and hardware scheduling affect measured runtime.

Table 2: Route efficiency and planner overhead under nominal communication. The analysis uses n=60 episodes per architecture, averaged across scenarios; entries are mean ± standard error. Route efficiency is the elapsed simulated time for successful routes. Token and call rates are measured per vehicle per simulated second and include planner traffic only.

Architecture	Route Eff. (s)	Prompt Tok/s	Completion Tok/s	Planner Calls/s
Independent	9.5 ±0.5	4,367.1 ±35.7	38.4 ±0.5	1.91 ±0.01
Centralized	11.2 ±0.4	4,169.4 ±48.7	45.9 ±0.6	0.93 ±0.01
Decentralized	11.2 ±0.5	5,208.6 ±59.1	169.8 ±2.7	1.88 ±0.02
PNYX$_{\text{Proto}}$	13.2 ±0.5	7,320.8 ±132.7	68.8 ±1.3	1.79 ±0.03
PNYX	12.8 ±0.6	5,253.5 ±88.0	79.2 ±1.8	1.79 ±0.03

Free-form decentralized planning — collision at t=4.75 s

t (s)	**car1** (blocked by the truck)	**car2** (oncoming)
0.5	"I am accelerating … to overtake by passing on the left. **Please hold your position**." →`lane follow`	"No conflict with your position in the opposite lane." →`lane follow`
1.5	"I am accelerating … I will move into the left lane (your lane)." →`slow down`	"Acknowledged. **I am slowing and holding** to let you overtake." →`speed up`
3.0	"Acknowledged. **I am holding my position** until you have cleared." →`change left`	"I will maintain my speed and lane to let you overtake." →`stop`
4.0	"**I am accelerating now to overtake.** The lane is clear ahead." →`stop`	"I am accelerating … **You may proceed** with your overtake." →`speed up`

4.75 s: head-on collision in the oncoming lane.

Centralized planning, same scenario — both vehicles succeed, 100% route completion

1.5	"**Stop** behind the stopped vehicle **until car2 has passed**, then overtake on the left." →`stop`	"Accelerate and continue forward in your lane." →`speed up`
2.5–5.5	"Stop and wait in lane 1 until car2 has passed" *(restated every cycle)* →`stop`	"Continue forward in lane −1 at cruise speed." →`lane follow`

Figure 7: Selected steps from a negotiation-overtake trace. Free-form planners make conflicting statements about who should proceed; the centralized planner maintains an explicit order. Arrows show executor commands, which sometimes differ from the accompanying language. The trace therefore does not isolate coordination failure as the sole cause of collision.

B Overhead and Efficiency Metrics

Table 2 reports planner usage and route efficiency under nominal communication. Independent planning provides the no-communication reference, using 4,367.1 prompt tokens/s, 38.4 completion tokens/s, and 1.91 planner calls/s. Centralized planning uses 0.93 calls/s because one invocation produces both assignments; it processes 4,169.4 prompt tokens/s and 45.9 completion tokens/s, with a mean successful-route time of 11.2 s. Free-form decentralized planning uses 1.88 calls/s and generates 169.8 completion tokens/s, 3.7× the centralized rate, while matching its 11.2 s route time and remaining 10.0 percentage points lower in overall success. In this setting, additional dialogue increases generated language without producing a comparably reliable joint decision.

The structured systems use 1.79 planner calls/s, slightly fewer than free-form decentralized planning, and generate substantially less completion traffic. PNYX protocol only produces 68.8 completion tokens/s and PNYX produces 79.2, reductions of 59.5% and 53.4%, respectively. Their prompt cost differs because protocol state, acknowledgements, and shared-state payloads re-enter the planner context: PNYX protocol only uses 7,320.8 prompt tokens/s, whereas PNYX uses 5,253.5, close to the 5,208.6 used by free-form decentralized planning. Mean successful-route time is 13.2 s for PNYX protocol only and 12.8 s for PNYX. The success improvements are therefore not accompanied by higher planner-call rates or shorter successful routes; they incur additional input context and more conservative completion times.

C AGENT DIALOG

D PROTOCOL PARAMETERS

PNYX derives its runtime communication limits from the available channel profile and the time remaining before the coordination deadline. The message budget β_{msg} limits the serialized payload admitted to one transmission. The acknowledgement timeout τ_{ack} determines when an unacknowledged message becomes eligible for retransmission, and $R_{\max}$ bounds the number of attempts. The commitment deadline T_{commit} bounds the complete information-sharing and action-negotiation exchange so that an otherwise valid agreement is not released after the relevant task decision. Revision and read-back are optional rounds and are removed when their expected duration would exceed the remaining deadline.

The mapping from channel conditions to β_{msg}, τ_{ack}, and $R_{\max}$ is fixed using dedicated tuning episodes that are excluded from both the protocol ablations and the main architecture comparisons. During evaluation, the system is provided the active degradation profile and its associated channel parameters, which we assume can be estimated during deployment, and applies the corresponding runtime limits deterministically. This runtime derivation is distinct from the payload and rejection configurations in Sec. 6.4: those configurations are selected offline for each architecture and degradation level using separate ablation runs and are then evaluated on independent main runs.

E LEADER-ELECTION ALGORITHM

Algorithm 1 gives the task-scoped election used by PNYX. The participant set $\mathcal{C}_q$ is fixed for one election transaction, and each non-Byzantine participant contributes at most one candidate record and one vote. All participants compute the same lexicographic ranking from the collected candidate records and bind their votes to a digest of the task, participant set, election transaction, and records. A candidate activates only with one matching vote from every participant. This rule guarantees that two candidates cannot obtain complete certificates for the same transaction and participant set; it does not guarantee availability, because a missing participant causes the election to terminate in local fallback.

Algorithm 1 Task-scoped leader election at planner i

Require: Task q, participants $\mathcal{C}_q$, election transaction z_{e}
Ensure: Active leader ℓ^*, or FALLBACK

Broadcast CANDIDATE$(q, z_{\mathrm{e}}, i, \rho_i)$
Collect one candidate record from every $j \in \mathcal{C}_q$
if any record is missing at timeout **then**
    return FALLBACK
end if
$\ell^* \leftarrow \arg\max_{j \in \mathcal{C}_q} \rho_j$
$d \leftarrow H(q, z_{\mathrm{e}}, \mathcal{C}_q, \{\rho_j\}_{j \in \mathcal{C}_q})$
Send VOTE$(q, z_{\mathrm{e}}, \ell^*, d)$ to candidate ℓ^*
if $i = \ell^*$ **then**
    Collect matching votes from every $j \in \mathcal{C}_q$
    if the vote set is incomplete at timeout **then**
        return FALLBACK
    end if
    $\Gamma_{z_{\mathrm{e}}} \leftarrow$ complete matching vote set
    Broadcast ACTIVATE$(q, z_{\mathrm{e}}, \ell^*, d, \Gamma_{z_{\mathrm{e}}})$
    return LEADER
else
    Wait for ACTIVATE$(q, z_{\mathrm{e}}, \ell^*, d, \Gamma_{z_{\mathrm{e}}})$ until timeout
    if a valid ACTIVATE arrived and $\Gamma_{z_{\mathrm{e}}}$ contains one matching vote from every participant **then**
        return FOLLOWER
    else
        return FALLBACK
    end if
end if