Waggle: LEARNING ONE ANONYMOUS LOCAL LAW FOR SELF-ORGANIZING LLM SWARMS

Anonymous authors
Paper under double-blind review

ABSTRACT

As LLM agents increasingly collaborate on complex tasks, how to organize their interactions becomes a central design question. Existing multi-agent systems typically learn or adapt explicit roles, hierarchies, routing policies, or communication topologies. We shift the learning target to a reusable local law that can be shared across interchangeable agents and adapt coordination as populations or interaction conditions change, without redefining a global organization. We introduce **Waggle**, a shared anonymous policy over bounded local views that jointly selects task actions, semantic communication, and local commitment updates. Repeated execution of the same law allows coordination to form, persist, and reorganize online without explicit roles or global topology. To learn this law across interchangeable agents and evolving coordination, we develop **Swarm-Consistent Distillation (SCD)**, combining anonymous-orbit consistency with rollout-grounded prediction of the next local coordination field, with no added inference-time components. Across diverse coordination settings, the same learned law remains effective as populations and interaction budgets change, retains over 96% of substrate-specific oracle quality, and transfers without retraining; SCD further improves reorganization after counterevidence. Together, these results show that LLM-agent organization can emerge and adapt through repeated execution of a learned local law.

1 INTRODUCTION

LLM agents can reason, use tools, and revise plans from feedback, enabling flexible collaborative systems (Yao et al., 2023). Multi-agent systems organize these capabilities through roles, managers, routes, and conversation graphs (Li et al., 2023; Hong et al., 2024; Qian et al., 2024; Wu et al., 2023a; Zhuge et al., 2024). Recent methods further design task-dependent communication graphs, jointly generate agent compositions and topologies, or adapt connectivity and agent activation online (Zhang et al., 2025b; Li et al., 2026b; Yang et al., 2025b; Wu et al., 2026). Despite this flexibility, the learned object remains an explicit organization—the graph, routing policy, or agent composition.

Swarm intelligence suggests a different organizing principle: persistent yet revisable population-level coordination can emerge from repeated local reactions to local signals, without requiring a population-wide plan (Bonabeau et al., 1999; Theraulaz & Bonabeau, 1999; Seeley et al., 2012). Related decentralized and swarm-inspired learning systems provide computational precedents for this local-rule view (Hüttenrauch et al., 2019; Shaw et al., 2022; Yang et al., 2025b; Li et al., 2025). Inspired by this local-rule view, we shift the learning target from explicit organization to the reusable local law that generates and revises it. This law can be shared across interchangeable agents and reused as populations or interaction conditions change.

Yet effective local laws require more than individually reasonable actions. Proposals based on private observations can become redundant, conflicting, or stale when composed across a population. This failure is pronounced in SILO-BENCH: at $N = 64$, transient-exchange and fixed-rule baselines reach only about 15% success, compared with 58.2% for our method (Table 1), and additional interaction alone does not close the gap (Figure 3). The bottleneck is therefore not merely local competence or message availability, but how a shared local rule composes proposals into evolving collective state.

We therefore treat proposals as inputs to coordination and learn the shared policy that determines each local decision. This leads to our central question: **can a shared anonymous local law generate,**

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.

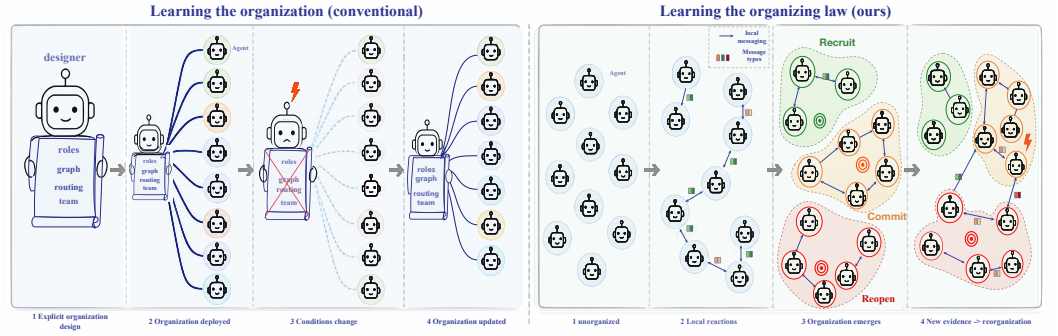


Figure 1: **Learning the organization versus the organizing law.**

maintain, and revise organization online through repeated execution? Figure 1 contrasts learning an explicit organization with learning the local law that generates it.

We introduce Waggle, which instantiates this learning object as one anonymous, parameter-shared policy over local views. Each local decision jointly chooses a task action and a bounded semantic update that can create or revise persistent local coordination state. We train the law with *Swarm-Consistent Distillation* (SCD). Beyond imitating teacher decisions, SCD encourages the policy to remain semantically consistent when opaque identifiers or neighbor orderings change, while learning representations that predict how the local coordination state evolves from one update to the next. Figure 2 summarizes the resulting training-to-deployment pipeline. This consistency carries over to the swarm level: if local views and runtime updates treat relabelings consistently, renaming agents or local references changes only the labels, not the resulting coordination behavior. Across spatial and distributed-language coordination settings, the same learned law remains effective as population size and interaction budget change and adapts more reliably when later evidence contradicts an established commitment.

Our contributions are:

- **Conceptually,** we reformulate the learning target of LLM-agent organization from explicit organizational structures to the local law that generates them: Waggle learns one shared anonymous local reaction law whose repeated execution generates and revises organization online.
- **Methodologically,** we develop *Swarm-Consistent Distillation* (SCD), which trains this law toward consistent decisions across anonymous local views and representations predictive of evolving coordination state, with no additional inference-time machinery.
- **Empirically,** we show that one learned law supports spatial and language-based coordination, remains effective as swarm conditions change, and reorganizes after counterevidence.

2 Related Work

LLM multi-agent organization. LLM multi-agent systems encode organization through roles, identities, protocols, or programmable interaction patterns (Li et al., 2023; Hong et al., 2024; Qian et al., 2024; Wu et al., 2023a). DyLAN and AgentVerse adapt team composition and collaboration (Liu et al., 2024; Chen et al., 2024b). Other methods optimize workflows or communication topology, or jointly generate task-conditioned agent compositions and graphs, as in ARG-Designer (Zhang et al., 2025c; Zhuge et al., 2024; Zhang et al., 2025a;b; Li et al., 2026b). Others route or activate specialists through adaptive connectivity, publish–subscribe structures, or specialized agent classes (Wu et al., 2026; Yang et al., 2025b; Li et al., 2026a; 2025). LLawCo derives natural-language cooperation principles from failures for embodied planners, while multi-agent distillation compresses interaction graphs or dynamics into a student (Zhou et al., 2026; Chen et al., 2024a; Luo et al., 2026). Agent Distillation trains smaller agents on teacher-generated interaction trajectories (Kang et al., 2025). Across these approaches, organization remains represented, adapted, or compressed through system-level structures or agents; Waggle instead learns one shared anonymous local law whose repeated execution generates and revises organization online.

Learned communication and swarm coordination. Learned-communication methods jointly optimize control with message content, broadcast gates, or recipient selection (Sukhbaatar et al.,

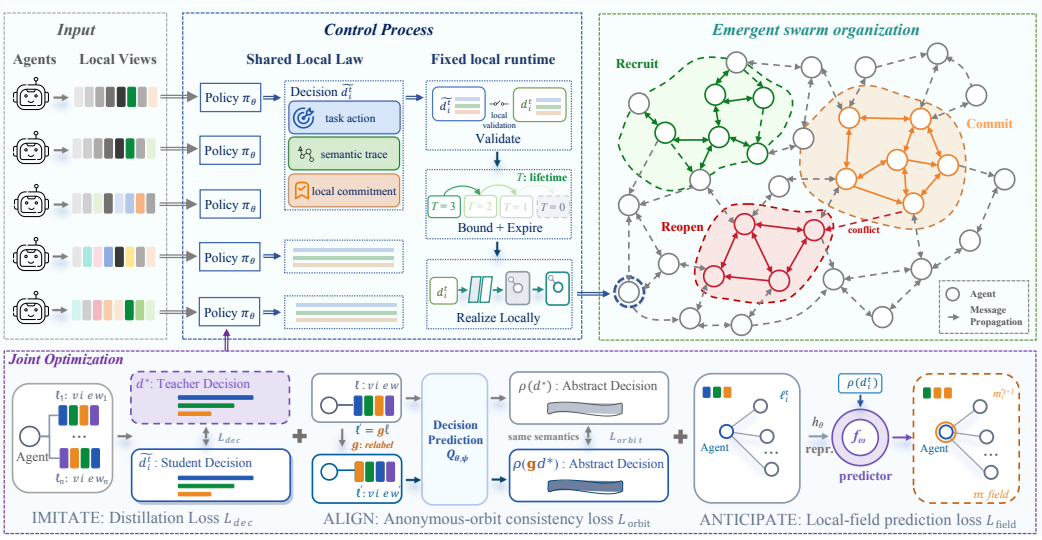


Figure 2: **Overview of Waggle.** SCD trains one shared anonymous local law; each agent deploys it independently through a fixed runtime, and repeated local reactions form and revise swarm organization.

2016; Foerster et al., 2016; Singh et al., 2019; Das et al., 2019), while shared-policy swarm RL uses interchangeable controllers with local, variable-size observations (Hüttenrauch et al., 2019). LLM2Swarm explores LLM-based controller synthesis and per-robot reasoning, whereas GenSwarm generates and deploys task-specific multi-robot code policies from natural-language instructions (Strobel et al., 2024; Ji et al., 2026). Population protocols repeatedly apply a shared local transition rule among anonymous finite-state agents, whereas stigmergic systems coordinate through persistent local state, from ant-colony reinforcement and evaporation to learned deposition and sensing in ForMIC (Angluin et al., 2006; Theraulaz & Bonabeau, 1999; Dorigo & Gambardella, 1997; Shaw et al., 2022). Building on shared control, learned communication, local rules, and persistent state, Waggle learns the organization-generating local law itself.

Symmetry-consistent learning. Anonymous-network theory characterizes when topology and local information permit symmetry-sensitive tasks such as leader election (Yamashita & Kameda, 1996). Permutation structure can be built in through parameter sharing or set architectures, or promoted through transformed examples and consistency objectives; related work also regularizes task-equivalent prompts and models transition regularities in world and local MARL models (Ravanbakhsh et al., 2017; Zaheer et al., 2017; Hounie et al., 2023; Zhou et al., 2022; Park et al., 2022; Wu et al., 2023b). SCD brings these ideas into a deployed anonymous local policy by promoting consistent decisions across equivalent local views and representations that track evolving coordination state.

3 Waggle: The Deployed Anonymous Local Law

Waggle separates a learned semantic reaction law from a fixed local runtime. We first define what each anonymous node observes and decides, then show how repeated decisions generate reversible coordination, and finally state the boundary between learned semantics and substrate realization.

3.1 Anonymous Local View

In the primary synchronous runtime, t indexes local-update rounds on a contact graph $G_t = (V, E_t)$, with all active-node views formed from the same pre-update state. Node i has private task state and evidence p_i^t, a set $\mathcal{H}_i^t$ of currently incident channel handles, and an optional local commitment c_i^t. Handles are opaque, sampled independently for each node from an episode-local pool, and fixed only within that episode; they are routing references rather than designated agent identities. Every active node receives the same bounded interface,

$$\ell_i^t = \mathrm{Canon}\Big(q,\ p_i^t,\ w_i^t,\ \{\phi_{ih}^t\}_{h\in\mathcal{H}_i^t},\ c_i^t,\ b_i^t\Big), \tag{1}$$

where q is the task contract, w_i^t is an adapter-provided candidate native action derived from private evidence p_i^t, and b_i^t is the remaining local budget. The private proposal is an input to the joint local decision, not a separate decision or language-model call. Canon standardizes field structure rather than substrate semantics: adapters place native task contracts, private observations, and proposals in the common q, p_i^t, and w_i^t slots, while incident communication uses a shared bounded trace schema. Native action names accompany the same controller instruction (Appendix D). Serialization exposes neither persistent identities nor a global node ordering.

A policy is *anonymous-local* when all nodes share θ, every social input arrives through an incident channel, and neither its view nor output schema contains a designated identity, role, roster, population-size field, non-neighbor state, or global transcript. The interface enforces anonymity; Section 4.2 trains consistency across equivalent views obtained by permuting incident records and consistently relabeling opaque local references.

3.2 One joint local decision

Each node invokes the same controller once per local update,

$$\tilde{d}_i^t = \mathrm{Decode}_\theta(\ell_i^t), \qquad d_i^t = \mathrm{Valid}_s(\tilde{d}_i^t, \ell_i^t) = (a_i^t, m_i^t, \Delta c_i^t), \tag{2}$$

where s denotes the substrate, a_i^t is a task-level action, m_i^t is an optional bounded semantic update addressed through incident handles, and Δc_i^t is an optional local commitment update. The decoder proposes all three components jointly, and the substrate validator admits only locally legal fields and trace relations. The decision type defines the available local effects, while π_θ learns their state-dependent selection and semantic content. All three components are produced by one decode, so acting and communicating are chosen jointly rather than by separate procedures.

The same decision type is used across substrates. The validator checks local legality, while the fixed runtime transports and expires admitted traces and realizes task-level outputs in native mechanics. The exact JSON serialization, field bounds, and prompt appear in Appendix D.

3.3 From repeated decisions to reversible organization

For claim z, a trace carried by incident handle $h \in \mathcal{H}_i^t$ has bounded state

$$\boldsymbol{\phi}_{ih}^t(z) = [n_{ih}^t(z),\ s_{ih}^t(z),\ x_{ih}^t(z),\ T_{ih}^t(z)], \tag{3}$$

where n, s, x, and T encode novelty, support, conflict, and remaining lifetime. The controller conditions its next local decision on these bounded traces.

Repeated local decisions can produce reversible coordination trajectories. Depending on its current evidence, a node may write or relay a record, form a commitment, or revise one after new conflict. The runtime supplies locally checkable readiness and finite expiry; the policy chooses the underlying semantic write and commitment update. These repeated reactions allow coordination to form and change without persistent roles or a broadcast transcript. Recruitment, commitment formation, and reopening are post hoc event labels rather than explicit controller modes.

3.4 Runtime boundary and substrate realization

The two primary substrates implement different native mechanics behind this boundary. SwarmBench realizes records as egocentric actions and contact-local cue pulses, whereas Silo-Bench uses private-shard operations and local-port evidence frames. Readiness is similarly local: SwarmBench exposes a fixed support-and-coherence predicate and Silo-Bench exposes native constructibility. Fresh writes must match private evidence; relays must reference incident evidence, cannot amplify its trace components, and lose at least one TTL step. Detailed native realizations, duplicate suppression, expiry, and compiler behavior appear in Appendices B, C, and H.

This contract defines what the local law can observe and change; the remaining question is how to learn a law that is stable across anonymous views and sensitive to evolving coordination state.

4 Learning the Law with Swarm-Consistent Distillation

Decision distillation teaches the policy to emit valid local records, but leaves two swarm-specific ambiguities unresolved. First, equivalent anonymous views may induce different semantic decisions

when incident order and opaque names change. Second, imitation alone does not require the controller representation to capture how the local coordination field evolves during collective execution. *Swarm-Consistent Distillation* (SCD) addresses these ambiguities with consistency across anonymous views and rollout-grounded next-field prediction.

4.1 Decision distillation

Let $\mathcal{D}_{\mathrm{dec}} = \{(\ell_a, d_a^\star)\}_{a=1}^{A}$ denote anonymous local views paired with schema-valid teacher decisions. The teacher observes exactly one local view and returns the joint decision defined in Section 3.2. Using assistant-record likelihood only, the decision objective is

$$\mathcal{L}_{\mathrm{dec}} = \mathbb{E}_{(\ell,d^\star)\sim\mathcal{D}_{\mathrm{dec}}}\big[-\log p_\theta(d^\star \mid \ell)\big]. \tag{4}$$

It teaches the controller to map private evidence and incident traces to a legal task action, semantic write, and commitment decision. The token-level expansion and corpus construction appear in Appendix F.1.

4.2 Consistency across anonymous views

For a local view ℓ, let $\mathcal{G}(\ell)$ be the family generated by incident-record permutations and view-local bijections over opaque channel and claim references. For each training pair, we sample $g \sim \mathcal{G}(\ell)$ and construct $(g\ell, gd^\star)$ by applying g consistently to the view and record, while non-identifier task content remains unchanged. Thus the pair encodes one anonymous local situation under two surface representations. Let $\rho(d)$ be an identifier-free summary of the task action, communication act, trace attributes, and commitment; by construction, $\rho(gd) = \rho(d)$.

Orbit supervision has two complementary roles. Transformed token supervision preserves the correct opaque references in the emitted record, while identifier-free alignment requires the two views to represent the same semantic decision. A factorized training head $Q_{\theta,\psi}$ predicts $\rho(d)$ from the controller representation, giving the compact objective

$$\begin{aligned}\mathcal{L}_{\mathrm{orbit}} = \mathbb{E}_{(\ell,d^\star),g}\Big[&-\log p_\theta(gd^\star \mid g\ell) + \mathcal{L}_{\mathrm{sup}}(Q_{\theta,\psi}(\ell), Q_{\theta,\psi}(g\ell); \rho(d^\star)) \\ &+ \beta D_{\mathrm{SKL}}(Q_{\theta,\psi}(\ell), Q_{\theta,\psi}(g\ell))\Big].\end{aligned} \tag{5}$$

Here $\mathcal{L}_{\mathrm{sup}}$ is factorized supervised cross-entropy on both views and D_{SKL} is symmetric KL divergence. Supervision prevents content-free agreement, while transformation-consistent augmentation preserves executable references (Hounie et al., 2023; Zhou et al., 2022); Appendix F.2 gives the complete factorization.

Exact local equivariance lifts to the swarm level under an equivariant runtime. Let a legal Γ have node-permutation component σ and consistently transport every schema-designated opaque reference. It induces $g_{\Gamma,i}^t$ on local views and records, including incident-order permutations, with $(\Gamma D^t)_{\sigma(i)} = g_{\Gamma,i}^t d_i^t$; below we suppress the Γ, t indices on g. Write F for one synchronous update and let $\pi_\theta^{\mathrm{dep}}$ denote the complete local decode-and-admit pipeline, including permitted repair and conservative projection.

Proposition 1 (Conditional local-to-global relabeling equivariance)**.** *Suppose every node uses the same $\pi_\theta^{\mathrm{dep}}$. Suppose also that view construction, $\pi_\theta^{\mathrm{dep}}$, and F commute with every legal relabeling:*

$$\mathrm{View}_{\sigma(i)}(\Gamma S) = g_i\,\mathrm{View}_i(S), \qquad \pi_\theta^{\mathrm{dep}}(g_i\ell) = g_i\pi_\theta^{\mathrm{dep}}(\ell), \qquad F(\Gamma S, \Gamma D; \Gamma\xi) = \Gamma F(S, D; \xi).$$

If $S_\Gamma^0 = \Gamma S^0$ and exogenous randomness is coupled by the same relabeling, then $(S_\Gamma^t, D_\Gamma^t) = (\Gamma S^t, \Gamma D^t)$ for every t. Consequently every trajectory functional J satisfying $J(\Gamma\tau) = J(\tau)$ is unchanged.

Appendices E.1–E.2 give the proof and its event-driven extension under equivariant scheduling.

Corollary 1 (Symmetry preservation)**.** *Under Proposition 1, consider a deterministic execution and a nonidentity legal relabeling Γ satisfying $\Gamma S^0 = S^0$ and $\Gamma\xi^t = \xi^t$. Then $\Gamma S^t = S^t$ and $\Gamma D^t = D^t$ for every t; hence a uniquely selected node must be fixed by Γ.*

Thus anonymity preserves genuine symmetries until task state, topology, interaction history, or episode-local randomness distinguishes the alternatives. SCD promotes the proposition's local premise, while the AgentsNet-RP diagnostic in Appendix I tests the corollary's symmetry-breaking prediction.

4.3 ANTICIPATING THE LOCAL COORDINATION FIELD

Anonymous consistency constrains how a decision is represented but does not directly teach how its local coordination field evolves. After fitting a decision-supervised warm start, we execute complete training-seed episodes through the fixed runtime and collect genuine adjacent transitions $\mathcal{D}_{\mathrm{tr}} = \{(\ell_i^t, d_i^t, \ell_i^{t+1})\}$, where d_i^t is the admitted focal decision and ℓ_i^{t+1} is the next view of the same node after the synchronous update. The bounded, identifier-free target is

$$m(\ell) = [\mathrm{Hist}_n(\ell), \mathrm{Hist}_s(\ell), \mathrm{Hist}_x(\ell), \mathrm{Hist}_T(\ell), r(\ell), \kappa(\ell)], \tag{6}$$

where the four histograms summarize incident novelty, support, conflict, and remaining lifetime, while r and κ denote local readiness and commitment. Writing $h_\theta(\ell_i^t)$ for the controller's final current-view hidden state, a training-only head f_ω receives this representation together with the admitted-decision summary $\rho(d_i^t)$ and predicts $\hat{m}_i^{t+1} = f_\omega(h_\theta(\ell_i^t), \rho(d_i^t))$. We group the prediction loss by target family,

$$\mathcal{L}_{\mathrm{field}} = \mathbb{E}_{\mathcal{D}_{\mathrm{tr}}}\left[\mathcal{L}_{\mathrm{hist}}(\hat{m}_i^{t+1}, m(\ell_i^{t+1})) + \lambda_{\mathrm{state}}\mathcal{L}_{\mathrm{state}}(\hat{m}_i^{t+1}, m(\ell_i^{t+1}))\right]. \tag{7}$$

$\mathcal{L}_{\mathrm{hist}}$ averages cross-entropy over the four trace histograms, whereas $\mathcal{L}_{\mathrm{state}}$ averages over readiness and commitment. We use equal group weight, $\lambda_{\mathrm{state}} = 1$, so the four histogram targets do not dominate the two state targets by count alone; the componentwise expression appears in Appendix F.3.

Because neighboring nodes act simultaneously and the substrate and runtime also determine transport, expiry, and commitment updates, this objective implements rollout-conditioned next-field prediction rather than an isolated causal model of the focal decision. The objective therefore encourages the shared representation to retain local regularities involving support accumulation, readiness, commitment, conflict-triggered reopening, and expiry—local changes relevant to subsequent reorganization—rather than reconstruct a global world state (Park et al., 2022; Wu et al., 2023b).

4.4 JOINT TRAINING AND DEPLOYMENT

The complete objective is

$$\mathcal{L}_{\mathrm{SCD}}(\theta, \psi, \omega) := \mathcal{L}_{\mathrm{dec}}(\theta) + \lambda_{\mathrm{orbit}}\mathcal{L}_{\mathrm{orbit}}(\theta, \psi) + \lambda_{\mathrm{field}}\mathcal{L}_{\mathrm{field}}(\theta, \omega). \tag{8}$$

Here, θ parameterizes the shared local law π_θ retained at deployment, whereas ψ and ω parameterize the training-only orbit and field heads, respectively. Training first obtains a decision-supervised warm start, uses that policy to collect a fixed corpus of rollout transitions, and then jointly updates all three parameter sets with interleaved decision and transition minibatches. Both auxiliary heads are discarded afterward; deployment retains only π_θ, so SCD adds no inference-time components. Appendix F specifies the model, data, complete losses, and optimization schedule.

5 EXPERIMENTS

We organize the evaluation from capability to mechanism. Scaling, specialist retention, and frozen transfer establish what one law can do; structural and late-counterevidence diagnostics connect learning to behavior, while orthogonal controls separate policy and runtime.

5.1 EXPERIMENTAL SETUP AND BASELINES

Our primary matrix evaluates one jointly trained law at $N \in \{16, 32, 64\}$ on two complementary local-interaction substrates, with all evaluation cases and seeds held out from training. SWARMBENCH evaluates spatial coordination with contact-local communication (Ruan et al., 2025); SILO-BENCH integrates private language shards through sparse local ports (Zhang et al., 2026).

We compare three interface-matched controls: NOCOMM suppresses explicit inter-agent messages (Hüttenrauch et al., 2019); NATIVEMSG uses transient task-native messages over matched contacts, without a persistent coordination field or quorum (Sukhbaatar et al., 2016; Foerster et al., 2016; Zhang et al., 2026); and FIXEDSWARM replaces learning with a shared hand-coded local rule over finite-lifetime signals (Angluin et al., 2006; Shaw et al., 2022). Broader comparisons include GPTSWARM, G-DESIGNER, ARG-Designer, AGENTNET, DMoA, SWARMSYS, and Agent Distillation (AGENTDISTILL) (Zhuge et al., 2024; Zhang et al., 2025b; Li et al., 2026b; Yang et al., 2025b; Wu et al., 2026; Li et al., 2025; Kang et al., 2025). All methods share task instances, populations,

Table 1: Cross-substrate scaling over five matched seeds. SILO-BENCH reports success (%) / messages per agent; SWARMBENCH reports raw and task-normalized scores. Lavender/peach mark first/second; paired intervals appear in Appendix H.4.

	SILO-BENCH						SWARMBENCH						Overall ↓
	Population scaling: SR ↑ / msg. ↓			$N = 64$ SR by difficulty ↑			Raw scores ↑: Pursuit / Foraging / Synchronization			Task-normalized ↑			Avg. rank ↓
Method	$N = 16$	$N = 32$	$N = 64$	Level I	Level II	Level III	$N = 16$	$N = 32$	$N = 64$	$N = 16$	$N = 32$	$N = 64$	
Interface-matched controls													
NOCOMM	10.7/0.00	9.8/0.00	11.7/0.00	20.0	10.1	5.0	1.71/1.31/0.30	1.30/0.89/0.10	7.00/2.86/0.00	0.222	0.153	0.627	9.17
NATIVEMSG	74.7/9.00	67.0/9.00	14.3/9.00	25.2	13.0	4.7	1.50/1.03/3.40	0.80/0.53/2.00	0.99/0.82/0.80	0.348	0.194	0.167	8.46
FIXEDSWARM	77.6/9.00	69.7/9.00	15.0/9.00	26.6	13.9	4.5	1.80/0.84/4.20	2.20/1.16/2.80	6.19/2.33/1.50	0.389	0.366	0.615	6.38
Broader baselines													
GPTSWARM	12.0/4.10	11.3/4.98	13.7/4.40	23.2	12.9	5.0	1.61/0.96/2.00	2.40/1.37/1.70	8.21/3.55/1.30	0.275	0.338	0.821	7.96
G-DESIGNER	14.0/3.80	13.6/4.30	16.0/4.00	27.5	15.3	5.2	1.70/1.02/2.40	2.28/1.42/2.00	8.01/3.65/1.50	0.306	0.351	0.829	7.00
ARG-Designer	19.9/3.23	19.1/3.51	20.8/3.41	34.6	22.3	6.0	1.80/1.08/3.20	2.40/1.55/2.60	8.15/3.85/2.00	0.358	0.400	0.880	4.71
AGENTNET	25.7/6.61	16.3/6.66	14.7/6.68	24.8	14.1	5.2	1.90/0.99/2.90	2.60/1.32/2.20	5.70/2.23/1.60	0.340	0.370	0.585	6.88
DMOA	19.0/3.40	18.8/3.70	21.5/3.60	34.8	21.4	8.3	1.82/1.08/3.30	2.33/1.49/2.70	8.11/3.80/2.10	0.364	0.397	0.879	4.79
SWARMSYS	61.7/9.00	44.3/9.00	26.7/9.00	47.9	23.1	9.1	**2.00**/1.00/4.00	1.11/0.75/3.20	7.81/3.59/2.40	0.404	0.292	0.860	5.25
AGENTDISTILL	79.0/8.20	69.0/8.35	43.7/8.50	71.1	48.7	11.3	1.75/1.20/4.60	2.20/1.30/3.70	7.35/3.50/3.10	0.439	0.425	0.864	4.29
Waggle	**86.3**/7.66	**79.0**/7.72	**58.2**/8.03	**87.6**	**69.9**	**17.1**	1.90/**1.55**/**5.80**	**2.64**/**1.64**/**5.00**	**8.23**/**4.33**/**4.20**	**0.540**	**0.546**	**1.041**	**1.12**

evaluation seeds, evaluators, and applicable interaction horizons. Interface-matched controls also share Waggle's topology and private proposals; broader baselines retain their native information-flow and private-computation interfaces subject to the documented budget settings. Appendix H.3 details benchmark integration and selection.

Tables 1–2 report point estimates; Appendix H.4 gives selected paired 95% intervals, resampling matched evaluation units, plus training seeds for independently trained controller comparisons.

5.2 ONE LAW SCALES ACROSS POPULATION AND INTERACTION BUDGETS

Despite observing only bounded local views and no population size, the same Waggle policy remains effective across all tested scales, achieving the strongest overall profile in the primary cross-substrate matrix (Table 1). Its advantage persists as populations grow, including 58.2% SILO-BENCH success at $N = 64$, while the distillation baseline does not close the gap under comparable inference cost (Appendix H.5). Appendices A (lower scales), C.1 (additional tasks), and N (SILO difficulty) extend these results.

We further test robustness to changing deployment conditions along two axes. Without retraining or adaptation, the frozen law scales to $N = 128$, retaining 82.9% of its matched $N = 64$ SILO-BENCH success while keeping per-agent communication bounded (Appendix C.2). The interaction-budget sweep shows a similar pattern: Waggle reaches strong performance with substantially fewer interaction rounds than competing methods (Figure 3). Together, these results indicate that the learned local law is not tied to a fixed population size or interaction horizon; the same local reactions continue to compose as both change.

5.3 ONE SHARED LAW RETAINS SPECIALIST QUALITY AND TRANSFERS

Having established composition across population and interaction budgets, we next test two forms of reuse: preserving specialist quality within the training substrates and transferring the frozen law to a new substrate. Under matched in-domain exposure, the joint law slightly exceeds the two specialists on both substrates. Even when each oracle specialist receives the entire joint data budget in its own domain, the shared law retains 96.8% and 97.6% of oracle quality on SWARMBENCH and SILO-BENCH, respectively, while using one adapter and half the aggregate labeled data (Figure 3(d); Appendix L).

On frozen AgentsNet transfer, the law leads four of five tasks, attains the best strict solved fraction at both sizes and the highest scale retention, and uses fewer messages than every communicating comparator (Table 2). Episode-local priority helps mainly on symmetry-obstructed LeaderElection graphs, consistent with Corollary 1; Appendix I gives the protocol and symmetry diagnostics. Complementary private-worker transfer and paired reuse results appear in Appendices J and K.

5.4 SCD LEARNS ANONYMOUS STABILITY AND LOCAL DYNAMICS

The four-arm factorial reveals a clear division of labor between the two SCD terms while held-out decision NLL remains effectively unchanged. Orbit supervision increases deployed anonymous-view

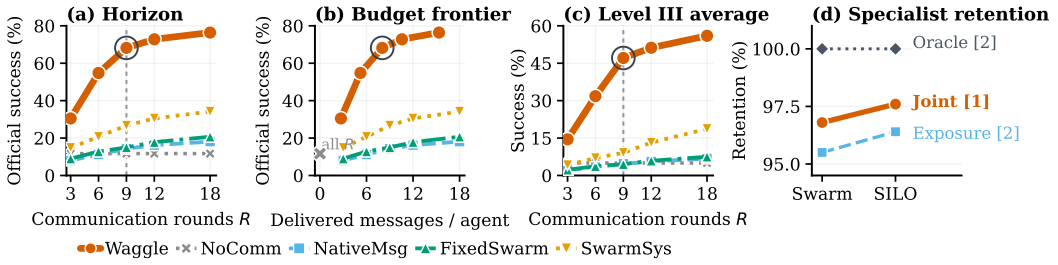


Figure 3: **Communication efficiency and specialist retention at $N = 64$.** (a,c) Success across communication rounds; (b) success versus delivered messages per agent; (d) oracle-relative retention. Outlined markers indicate the primary $R = 9$ setting.

Table 2: Frozen AGENTSNET-RP transfer without parameter updates. Lavender/peach mark first/second; paired 95% intervals for the main comparisons are reported in Appendix H.4.

Method	Task-wise soft score ↑					Strict solved fraction ↑			Scale ret. ↑	Efficiency	Overall ↓
	Coloring	Consensus	Leader-Election	Matching	Vertex-Cover	$N = 8$	$N = 16$	Overall	16/8	Msg./agent ↓	Avg. rank
Interface-matched controls											
NOCOMM	0.10	0.88	0.05	0.04	0.08	0.28	0.18	0.23	0.64	0.0	6.00
NATIVEMSG	0.34	0.90	0.55	0.25	0.18	0.54	0.35	0.44	0.65	28.4	4.93
FIXEDSWARM	0.44	0.91	0.72	0.34	**0.35**	0.64	0.46	0.55	0.72	15.8	2.29
Shared-controller variants											
ZERO-SHOT	0.39	0.90	0.63	0.30	0.21	0.58	0.38	0.48	0.66	14.6	4.07
DECISION ONLY	0.47	0.91	0.68	0.36	0.27	0.64	0.44	0.54	0.69	12.6	2.57
Waggle	**0.56**	**0.94**	**0.80**	**0.44**	0.32	**0.69**	**0.53**	**0.61**	**0.77**	**11.1**	**1.14**

agreement by 22.3 percentage points, whereas field supervision reduces frozen-probe loss by 34.3% on eventful transitions. Complete SCD combines both effects (Table 3). The training-head columns verify that each auxiliary objective is optimized, while the deployment columns measure what remains in the decoder and frozen controller representation after those heads are removed (Appendix G.1).

These diagnostics connect the two SCD objectives to distinct deployed properties. Orbit supervision improves decision consistency across equivalent anonymous views, while field supervision makes evolving local coordination state more accessible in the frozen representation. The former is the local property underlying Proposition 1; the latter captures dynamics relevant to later reorganization. The next experiment asks whether these structural changes translate into better reorganization after counterevidence.

5.5 SCD IMPROVES REORGANIZATION AFTER COUNTEREVIDENCE

The late-counterevidence intervention carries the same four-arm factorial into closed loop. With the runtime fixed, the two single-term arms yield intermediate behavioral gains, while complete SCD has the strongest joint profile. Recovery, requiring conflict-triggered reopening followed by evidence-supported replacement synthesis, rises from 80.0% to 88.5%, while stale-field burden and post-conflict traffic both fall (Table 3; paired effects in Appendix G.1). Stale-field AUC integrates obsolete semantic-trace mass after conflict, so lower values indicate faster clearance. A representative trajectory shows verified conflict dissolving the old commitment, synthesis building a replacement, and both activities subsiding after the new commitment stabilizes (Appendix P.2).

The same structural and behavioral pattern replicates with Llama-3.1-8B-Instruct under the same runtime (Appendix G.2). This motivates the attribution question addressed next: whether the gain comes from learned semantic decisions or from the fixed runtime affordances that realize them.

5.6 LEARNED DECISIONS AND RUNTIME MECHANICS PLAY COMPLEMENTARY ROLES

Figure 4 applies two orthogonal interventions: panel (a) fixes the interface and runtime while varying the decision source, whereas panels (b–d) fix the Waggle controller and remove conflict handling, reversibility, or decay; Appendix P.3 extends the component test. Under this fixed runtime, Waggle achieves the strongest joint profile across task quality and late-conflict reorganization. DECISION ONLY remains close on conventional quality but adapts less effectively after counterevidence, while

Table 3: Four-arm SCD factorial over five paired training seeds. Values are mean ± s.d.; exact equivariance entries are held-out agreement rates. Paired SCD effects are reported in Appendix G.1.

Objective	Imitation	Training objective			Deployed structure					Closed-loop reorganization		
	NLL ↓	Orbit-head acc. (%) ↑	Train-head loss ↓ All	Train-head loss ↓ Event	Summary agree. (%) ↑	Exact record equiv. (%) ↑ Decoded	Exact record equiv. (%) ↑ Admitted	Frozen-probe loss (norm.) ↓ All	Frozen-probe loss (norm.) ↓ Event	Recovery (%) ↑	Stale AUC ↓	SILO msg./agent ↓
DECISION ONLY	0.0191 ±0.0000	24.6 ±8.7	3.643 ±0.258	3.611 ±0.385	72.4 ±3.8	61.2	62.0	1.000 ±0.000	1.000 ±0.000	80.0 ±10.0	22.768 ±1.55	7.157 ±0.18
+ Orbit	0.0192 ±0.0001	96.6 ±0.4	3.512 ±0.301	3.398 ±0.288	94.7 ±0.8	88.6	89.4	0.968 ±0.030	0.944 ±0.040	82.0 ±9.4	22.2 ±1.45	7.05 ±0.17
+ Field	0.0191 ±0.0000	23.5 ±4.8	0.152 ±0.006	1.289 ±0.037	71.6 ±3.1	60.1	60.9	0.724 ±0.035	0.657 ±0.040	86.0 ±8.0	20.0 ±1.25	6.86 ±0.16
Waggle	0.0192 ±0.0001	**97.0** ±0.2	**0.147** ±0.006	**1.248** ±0.019	**95.2** ±0.5	**90.3**	**91.1**	**0.681** ±0.030	**0.608** ±0.035	**88.5** ±7.2	**19.237** ±1.10	**6.737** ±0.14

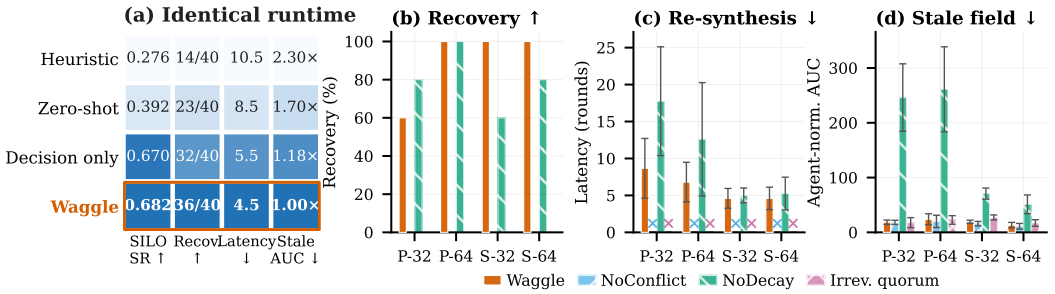


Figure 4: **Decision-source and runtime-component attribution.** (a) Decision laws under a shared runtime. (b–d) Runtime ablations on SWARMBENCH Pursuit (P) and a SILO-BENCH late-conflict task (S). Error bars show ±1 s.d.; crosses denote undefined metrics when no settings recover.

the zero-shot and heuristic laws trail further. These differences are therefore attributable to the decision source; Appendix M.1 reports the full comparison.

Across the instrumented evaluation matrix, nearly every first decode is executable as emitted, and the no-communication fallback is never invoked. Runtime repair can only remove invalid content or reuse the node's private proposal; it cannot originate social content (Appendix C.3; per-mode results in Appendix O).

Removing conflict handling or commitment reversibility collapses recovery; removing decay preserves most recovery but delays re-synthesis and stale-state clearance (Appendix P.1). Nearby runtime settings preserve performance while shifting the communication–responsiveness trade-off (Appendix P.4; execution perturbations: Appendix P.5). Together, these interventions separate learned semantics from the bounded runtime that realizes them.

6 CONCLUSION AND LIMITATIONS

Waggle shows that adaptive LLM-agent organization can be learned as a reusable anonymous local law, without representing organization as an explicit system-level structure. Across the tested substrates and scales, repeated local reactions form, sustain, and revise coordination, while SCD improves anonymous stability and reorganization after counterevidence. More broadly, these results suggest a complementary design principle for multi-agent systems: organization can arise repeatedly from a learned local rule instead of being specified directly.

The present study deliberately isolates this organizing law behind a bounded canonical interface and a fixed runtime. Deterministic substrate adapters expose private state, locally checkable evidence, and legal actions, while validation, bounded transport, expiry, and native realization remain fixed runtime mechanisms and are not learned end to end. The theoretical guarantee requires equivariant local decisions, view construction, and runtime updates; the empirical evidence establishes reuse across the tested local-interaction substrates and transfer settings. Appendices Q and R detail these scope boundaries and extensions.

Reproducibility Statement

Appendices B–D document the substrate adapters, runtime validation, and canonical local interface, while Appendix E gives the assumptions and proof of conditional relabeling equivariance. Training-corpus construction, data splits, optimization settings, and checkpoint selection are detailed in Appendices F and G. Appendix H specifies benchmark protocols, baseline implementations, and budget settings. Transfer configurations appear in Appendices I and J. Training and model selection exclude formal evaluation cases and seeds. The SCD comparison uses five paired training seeds with shared teacher records and optimization budgets, with matched-evaluation and paired-bootstrap procedures described in Appendix G.1.

Ethics Statement

We study coordination in benchmark environments using teacher-generated training records. The experiments contain no human-subject data. Anonymous handles remove identity cues from the controller interface but are not a privacy guarantee. In open-world deployments, unreliable observations or malicious messages could propagate through local interactions and influence collective actions. Bounded transport, finite trace lifetimes, and reversible commitments constrain coordination dynamics but do not establish a complete security boundary. Applications involving consequential external actions should authenticate evidence provenance, sandbox tools, protect sensitive inputs, and retain appropriate human approval. Appendix Q discusses these deployment assumptions and limitations.

AI Use Statement

GPT-4o was used in the experimental teacher and private-worker roles described in Section 4 and Appendix J. Generative AI was used for language polishing, including grammar, clarity, and stylistic editing; to assist with code development and debugging; and to support brainstorming of figure concepts and visual design ideas. All AI-assisted text, code, and analyses were critically reviewed and verified against the underlying sources, data, and implementations. The authors take full responsibility for the final manuscript.

References

Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4): 235–253, 2006. doi: 10.1007/s00446-005-0138-3. URL https://doi.org/10.1007/s00446-005-0138-3.

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999. doi: 10.1093/oso/9780195131581.001.0001. URL https://doi.org/10.1093/oso/9780195131581.001.0001.

Justin Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. MAGDi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7220–7235. PMLR, 2024a. URL https://proceedings.mlr.press/v235/chen24ah.html.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/578e65cdee35d00c708d4c64bce32971-Abstract-Conference.html.

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.

Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997. doi: 10.1109/4235.585892. URL https://doi.org/10.1109/4235.585892.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Florian Grötschla, Luis Müller, Jan Tönshoff, Mikhail Galkin, and Bryan Perozzi. AgentsNet: Coordination and collaborative reasoning in multi-agent LLMs. *arXiv preprint arXiv:2507.08616*, 2025. URL https://arxiv.org/abs/2507.08616.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.00352.

Ignacio Hounie, Luiz F. O. Chamon, and Alejandro Ribeiro. Automatic data augmentation via invariance-constrained learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13410–13433. PMLR, 2023. URL https://proceedings.mlr.press/v202/hounie23a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. Deep reinforcement learning for swarm systems. *Journal of Machine Learning Research*, 20(54):1–31, 2019. URL https://jmlr.org/papers/v20/18-476.html.

Wenkang Ji, Huaben Chen, Mingyang Chen, Guobin Zhu, Lufeng Xu, Roderich Groß, Rui Zhou, Ming Cao, and Shiyu Zhao. GenSwarm: Scalable multi-robot code-policy generation and deployment via language models. *npj Robotics*, 4(1):5, 2026. doi: 10.1038/s44182-025-00065-w. URL https://doi.org/10.1038/s44182-025-00065-w.

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. Distilling LLM agent into small models with retrieval and code tools. In *Advances in Neural Information Processing Systems*, volume 38, pp. 106501–106538. Curran Associates, Inc., 2025. doi: 10.52202/085713-3553. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/99263f9bf46874e2b8b7f104b5063864-Abstract-Conference.html.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17760.

Rui Li, Zeyu Zhang, Xiaohe Bo, Quanyu Dai, Chaozhuo Li, Feng Wen, and Xu Chen. Towards adaptive, scalable, and robust coordination of LLM agents: A dynamic ad-hoc networking perspective. *arXiv preprint arXiv:2602.08009*, 2026a. URL https://arxiv.org/abs/2602.08009.

Ruohao Li, Hongjun Liu, Leyi Zhao, Zisu Li, Jiawei Li, Jiajun Jiang, Linning Xu, Chen Zhao, Mingming Fan, and Chen Liang. SwarmSys: Decentralized swarm-inspired agents for scalable and adaptive reasoning. *arXiv preprint arXiv:2510.10047*, 2025. URL https://arxiv.org/abs/2510.10047.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. Assemble Your Crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23142–23150, 2026b. doi: 10.1609/aaai.v40i28.39481. URL https://doi.org/10.1609/aaai.v40i28.39481.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=XII0Wp1XA9.

Yinyi Luo, Yiqiao Jin, Weichen Yu, Mengqi Zhang, Srijan Kumar, Xiaoxiao Li, Weijie Xu, Xin Chen, and Jindong Wang. AgentArk: Distilling multi-agent intelligence into a single LLM agent. *arXiv preprint arXiv:2602.03955*, 2026. doi: 10.48550/arXiv.2602.03955. URL https://arxiv.org/abs/2602.03955.

OpenAI. Swarm: Educational framework for lightweight multi-agent orchestration. GitHub repository, 2024. URL https://github.com/openai/swarm.

Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan-Willem Van De Meent, and Robin Walters. Learning symmetric embeddings for equivariant world models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17372–17389. PMLR, 2022. URL https://proceedings.mlr.press/v162/park22a.html.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2307.07924.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabás Póczos. Equivariance through parameter-sharing. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2892–2901. PMLR, 2017. URL https://proceedings.mlr.press/v70/ravanbakhsh17a.html.

Kai Ruan, Mowen Huang, Ji-Rong Wen, and Hao Sun. Benchmarking LLMs' swarm intelligence. *arXiv preprint arXiv:2505.04364*, 2025. URL https://arxiv.org/abs/2505.04364.

Thomas D. Seeley, P. Kirk Visscher, Thomas Schlegel, Patrick M. Hogan, Nigel R. Franks, and James A. R. Marshall. Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science*, 335(6064):108–111, 2012. doi: 10.1126/science.1210361. URL https://doi.org/10.1126/science.1210361.

Samuel Shaw, Emerson Wenzel, Alexis Walker, and Guillaume Sartoretti. ForMIC: Foraging via multiagent RL with implicit communication. *IEEE Robotics and Automation Letters*, 7(2):4877–4884, 2022. doi: 10.1109/LRA.2022.3152688. URL https://doi.org/10.1109/LRA.2022.3152688.

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. URL https://openreview.net/forum?id=rye7knCqK7.

Volker Strobel, Marco Dorigo, and Mario Fritz. LLM2Swarm: Robot swarms that responsively reason, plan, and collaborate through LLMs. In *NeurIPS 2024 Workshop on Open-World Agents*, 2024. doi: 10.48550/arXiv.2410.11387. URL https://arxiv.org/abs/2410.11387.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.

Guy Theraulaz and Eric Bonabeau. A brief history of stigmergy. *Artificial Life*, 5(2):97–116, 1999. doi: 10.1162/106454699568700. URL https://doi.org/10.1162/106454699568700.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023a. URL https://arxiv.org/abs/2308.08155.

Xingjian Wu, Junkai Lu, Siyu Yan, Xiangfei Qiu, Jilin Hu, Chenjuan Guo, and Bin Yang. Differentiable mixture-of-agents incentivizes swarm intelligence of large language models. *arXiv preprint arXiv:2605.15706*, 2026. URL https://arxiv.org/abs/2605.15706.

Zifan Wu, Chao Yu, Chen Chen, Jianye Hao, and Hankz Hankui Zhuo. Models as agents: Optimizing multi-step predictions of interactive local models in model-based multi-agent reinforcement learning. *arXiv preprint arXiv:2303.17984*, 2023b. doi: 10.48550/arXiv.2303.17984. URL https://arxiv.org/abs/2303.17984.

Masafumi Yamashita and Tsunehiko Kameda. Computing on anonymous networks: Part I—characterizing the solvable cases. *IEEE Transactions on Parallel and Distributed Systems*, 7(1): 69–89, 1996. doi: 10.1109/71.481599. URL https://doi.org/10.1109/71.481599.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. URL https://arxiv.org/abs/2505.09388.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. AgentNet: Decentralized evolutionary coordination for LLM-based multi-agent systems. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2504.00587.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2410.02506.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-Designer: Architecting multi-agent communication topologies via graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76678–76692. PMLR, 2025b. URL https://proceedings.mlr.press/v267/zhang25cu.html.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=z5uVAKwmjf.

Yuzhe Zhang, Feiran Liu, Yi Shan, Xinyi Huang, Xin Yang, Yueqi Zhu, Xuxin Cheng, Cao Liu, Ke Zeng, Terry Jingchen Zhang, and Wenyuan Jiang. Silo-Bench: A scalable environment for

evaluating distributed coordination in multi-agent LLM systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29379–29398, San Diego, California, USA, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1354. URL https://aclanthology.org/2026.acl-long.1354/.

Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Prompt consistency for zero-shot task generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2613–2626. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.192. URL https://aclanthology.org/2022.findings-emnlp.192/.

Qinhong Zhou, Chuang Gan, and Anoop Cherian. LLawCo: Learning laws of cooperation for modeling embodied multi-agent behavior. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.28182.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.16823.

APPENDIX CONTENTS

A Lower-Scale Anchor Results . . . 16

B Detailed Native Realizations . . . 16

C Population, Controller, and Runtime Evidence . . . 16
- C.1 Population summaries and quality–communication trade-offs . . . 16
- C.2 Population extension beyond the primary matrix . . . 17
- C.3 Held-out local decisions . . . 18
- C.4 Mechanism activity on SWARMBENCH . . . 18

D Canonical Local Interface . . . 18
- D.1 Controller view . . . 18
- D.2 Decision schema . . . 19
- D.3 Anonymous controller prompt . . . 19

E Proof of Conditional Local-to-Global Relabeling Equivariance . . . 19
- E.1 Synchronous execution . . . 19
- E.2 Event-driven extension . . . 20

F Training Details . . . 21
- F.1 Decision loss, corpus, and split construction . . . 21
- F.2 Anonymous-orbit construction . . . 22
- F.3 Sequential transition collection . . . 22
- F.4 Optimization . . . 23

G SCD Training and Replication Protocols . . . 24
- G.1 Primary Factorial and Downstream Protocol . . . 24
- G.2 Replication with a Second Student Backbone . . . 25

H Benchmark Protocol . . . 25
- H.1 SWARMBENCH . . . 25
- H.2 SILO-BENCH . . . 26
- H.3 Baseline implementations . . . 26
- H.4 Paired uncertainty . . . 28
- H.5 Inference accounting . . . 29

I Frozen Transfer to Randomized-Anonymous AGENTSNET . . . 29

J GPT-4o Private-Worker Transfer . . . 31

K Focused Paired Evidence . . . 32
- K.1 Matched private-state reuse . . . 32
- K.2 Eight-seed paired evaluation . . . 32

L Joint Law versus Substrate-Specific Specialists . . . 32

M Same-Runtime Decision-Law Ablation . . . 34
- M.1 Decision-source comparison under a shared runtime . . . 34

N SILO-BENCH Scaling by Difficulty . . . 35

O Held-Out Controller Results . . . 35

P Mechanism Measurements . . . 35
- P.1 Runtime-component intervention . . . 35
- P.2 Representative late-conflict trajectory . . . 36
- P.3 Runtime-component task extension . . . 37
- P.4 Runtime-Parameter Sensitivity under a Frozen Controller . . . 37
- P.5 Delays, message loss, and temporary churn . . . 38

Q Limitations and Deployment Boundaries . . . 38

R Future Directions . . . 39

A LOWER-SCALE ANCHOR RESULTS

These lower-scale evaluations locate the frozen controller before the main population expansion. They use ten held-out seeds and are retained as context; the SWARMBENCH values provide the Pursuit/Foraging $N = 8$ anchors, with Synchronization normalized by its matched 6.40 reference. None of these episodes is reused in a complete-matrix comparison. Compact summaries appear in Tables 4 and 5; the population and task breakdowns below retain the detailed anchor evidence.

Table 4: Lower-scale SWARMBENCH anchor at $N = 8$.

	Pursuit		Foraging	
Method	Score ↑	Msg. ↓	Score ↑	Msg. ↓
NATIVEMSG	3.70 ± 3.33	1742.5 ± 531.9	2.90 ± 2.18	341.2 ± 325.4
FIXEDSWARM	6.10 ± 4.09	509.8 ± 148.5	3.70 ± 1.42	80.5 ± 73.5
Waggle	$\mathbf{6.20 \pm 7.76}$	$\mathbf{430.3 \pm 69.8}$	$\mathbf{3.80 \pm 2.39}$	$\mathbf{57.0 \pm 16.1}$

At the SWARMBENCH anchor, Waggle improves mean score by 0.10 over FIXEDSWARM on both tasks while reducing messages by 15.6% on Pursuit and 29.2% on Foraging.

Table 5: Lower-scale six-task SILO-BENCH anchor at $N \in \{4, 8\}$.

Method	Success (%) ↑	Msg. ↓	kB ↓
NATIVEMSG	**100.00**	54.00	49.21
FIXEDSWARM	**100.00**	54.00	50.04
Waggle	97.92	**41.63**	**34.47**

Table 6: Population breakdown of the lower-scale SILO-BENCH anchor.

N	Success (%)	Msg.	kB	NATIVEMSG/FIXEDSWARM kB
4	100.00	27.27	16.68	21.85 / 22.42
8	95.83	55.98	52.25	76.58 / 77.66

Across this lower-scale evaluation anchor, Waggle reaches 97.92% success with 22.9% fewer messages than either matched baseline and 30.0% fewer bytes than NATIVEMSG. The primary $N \in \{16, 32, 64\}$ cross-method SILO-BENCH evaluation is reported in Table 1; Appendix N breaks Waggle down by population and difficulty.

B DETAILED NATIVE REALIZATIONS

The same learned record is presented to both substrate adapters. Table 7 separates the policy's semantic choices from their fixed native realization. Adapters construct private proposals, expose locally checkable facts, and compile admitted records; they do not originate a claim, communication intent, commitment, or controller mode.

Table 7: Learned semantic decisions and fixed native realization in the two substrates.

Element	Learned local law	SWARMBENCH fixed realization	SILO-BENCH fixed realization
Task	Chooses a legal task action or waits	Egocentric local operation	Private-shard operation or response
Trace	Chooses content, incident target, and whether to relay	Contact-local cue with bounded per-hop relay	Local-port evidence frame with finite-lifetime relay
Commit and reopen	Chooses whether to synthesize, commit, challenge, or revise	Support/coherence readiness; contradictory cue	Native constructibility; incompatible partial evidence
Safeguards	Makes no schema, provenance, duplication, or expiry decision	Validates provenance, suppresses repeats, bounds relay, and expires cues	Validates provenance, suppresses repeats, bounds relay, and expires frames

C POPULATION, CONTROLLER, AND RUNTIME EVIDENCE

C.1 POPULATION SUMMARIES AND QUALITY–COMMUNICATION TRADE-OFFS

The task-normalized SWARMBENCH aggregate averages Pursuit, Foraging, and Synchronization after division by the frozen $N = 8$ Waggle anchors, 6.20, 3.80, and 6.40, respectively. Table 1 reports this

aggregate alongside the raw scores. Flocking and Transport complete the five-task SWARMBENCH evaluation; their full population-scale results appear in Table 8. Figure 5 retains the Pursuit/Foraging telemetry diagnostic, whereas Figure 6 uses all three tasks.

Table 8: Raw Flocking and Transport scores in the five-task SWARMBENCH evaluation. Bold/underline mark first/second within each task–population column, with lavender/peach shading.

	Flocking ↑			Transport ↑		
Method	$N = 16$	$N = 32$	$N = 64$	$N = 16$	$N = 32$	$N = 64$
Interface-matched controls						
NOCOMM	5.40	5.00	4.30	0.00	0.00	0.00
NATIVEMSG	6.00	5.70	5.00	0.25	0.20	0.10
FIXEDSWARM	7.60	7.20	6.50	0.45	0.55	0.45
Organizational baselines						
GPTSWARM	5.80	6.10	5.70	0.20	0.25	0.20
G-DESIGNER	6.20	6.50	6.10	0.30	0.35	0.30
ARG-Designer	6.70	7.00	6.70	0.45	0.55	0.50
AGENTNET	6.60	6.80	6.30	0.40	0.50	0.45
DMoA	6.80	7.10	6.80	0.50	0.60	0.55
SWARMSYS	7.20	7.50	7.00	0.65	0.80	0.75
AGENTDISTILL	7.00	7.10	6.40	0.55	0.65	0.55
Waggle	**7.80**	**8.10**	**7.70**	**1.20**	**1.55**	**1.70**

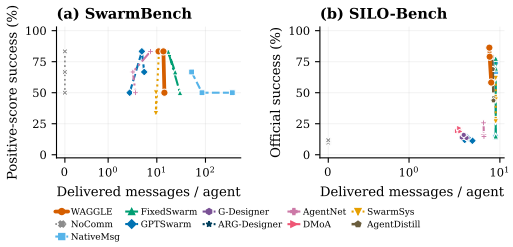


Figure 5: **Quality–communication trade-off over five matched evaluation seeds.** The shared legend lists all eleven methods: the SWARMBENCH Pursuit/Foraging subset shows the seven with both positive-score success and delivered-message telemetry, whereas SILO-BENCH shows all eleven.

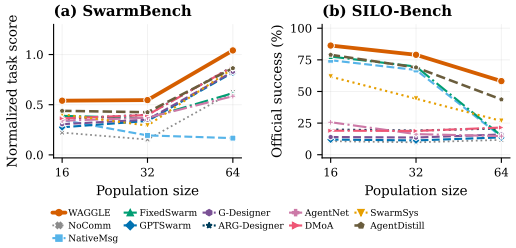


Figure 6: **Cross-method population scaling.** SWARMBENCH scores use task-specific frozen $N = 8$ Waggle references; SILO-BENCH reports official success.

C.2 POPULATION EXTENSION BEYOND THE PRIMARY MATRIX

We test deployment-time population extrapolation without retraining or checkpoint selection. The joint Waggle controller remains frozen, and we compare it with FIXEDSWARM, the interface-matched fixed-rule control, and SWARMSYS, a broader SILO-BENCH organizational baseline. The evaluation uses SILO-BENCH and SWARMBENCH Pursuit and Foraging at $N \in \{64, 128\}$ with the same five held-out seeds, yielding 960 episodes. The $N = 64$ values are newly evaluated five-seed anchors for this extension.

Table 9: Frozen-controller population extrapolation from $N = 64$ to $N = 128$; scale retention is $100\,\mathrm{SR}_{128}/\mathrm{SR}_{64}$.

	SILO-BENCH				SWARMBENCH	
Method	$N = 64$ SR (%) ↑	$N = 128$ SR (%) ↑	Scale retention ↑	$N = 128$ msg./agent ↓	$N = 64$ P/F task-norm. ↑	$N = 128$ P/F task-norm. ↑
Waggle	**68.5**	**56.8**	**82.9%**	**8.25**	**1.230**	**1.380**
FIXEDSWARM	15.4	8.6	55.8%	9.00	0.810	0.860
SWARMSYS	27.1	17.8	65.7%	9.00	1.100	1.150

The frozen law retains 82.9% of its matched $N = 64$ SILO-BENCH success at $N = 128$, compared with 55.8% for FIXEDSWARM and 65.7% for SWARMSYS, while using 8.25 delivered messages per

agent. Its Pursuit/Foraging task-normalized SWARMBENCH score also increases from 1.230 to 1.380. The scaling advantage in the primary matrix therefore persists when the deployment population doubles beyond $N = 64$.

C.3 HELD-OUT LOCAL DECISIONS

We evaluate the frozen controller on 800 held-out anonymous local views. Every generation is valid JSON, satisfies the schema, and is executable. Derived six-mode macro-F1 is 100%, including Challenge and Synthesize, while exact full-record match is 74.75%. Many non-exact records differ only in a bounded deposit value or an equivalent local choice without changing the operational mode. This is a record-format and derived-category fidelity diagnostic; end-to-end coordination is evaluated on full benchmark episodes in Section 5.

Table 10: Frozen-controller validity and derived-mode classification on 800 held-out local views.

	JSON valid	Schema valid	Executable	Mode macro-F1
Qwen3-4B + LoRA	100.00	100.00	100.00	100.00

Decision exact match: 74.75%.

Across the 72 Waggle cells in the core matrix, 85,129 of 85,344 first decodes were executable (99.75%). Of the 215 remaining first decodes, one was envelope-normalized, 207 were repaired by one validator-code-conditioned regeneration, and seven were conservatively projected; the no-communication fallback was never invoked. Wrapper operations are limited to deleting invalid content or reusing the local private proposal; social content remains controller-authored.

C.4 MECHANISM ACTIVITY ON SWARMBENCH

Entries are per-episode mean $\pm$ sample s.d. over the ten $N = 8$ anchor seeds.

Table 11: Controller-selected and runtime-filtered writes on SWARMBENCH at $N = 8$.

Task	Overall writes	Local suppression	Context safeguards
Pursuit	Candidate 420.6 ± 133.5; executed 139.9 ± 17.1	Repeats blocked 147.8 ± 110.8; refreshes 30.2 ± 20.7	Unverified challenges blocked 11.8 ± 7.6; route relays silenced 0.0
Foraging	Candidate 93.8 ± 47.7; executed 24.7 ± 8.3	Repeats blocked 27.6 ± 35.4; refreshes 0.0	Unverified challenges blocked 0.0; route relays silenced 26.6 ± 16.2

Table 11 shows that one local law selects different swarm primitives from event context. Pursuit keeps 1,399 of 4,206 candidate writes (33.3%), blocks 1,478 spatial repeats among 3,259 deposit candidates (45.4%), and executes 302 controlled refreshes as target evidence moves. Every emitted inhibitory challenge is backed by a verified local conflict (52/52), while 118 of 170 unverified challenge candidates are blocked. Foraging keeps 247 of 938 candidate writes (26.3%), blocks 276 spatial repeats among 446 deposit candidates (61.9%), and silences 266 of 492 route-active relay candidates (54.1%). Thus Pursuit activates refreshed evidence and verified conflict, whereas Foraging activates event-once evidence and private-route protection.

D CANONICAL LOCAL INTERFACE

D.1 CONTROLLER VIEW

The serialized controller view contains exactly six components: a task contract, private state, benchmark-native private proposal, incident fields, local commitment, and remaining-budget bin. The historical JSON key `worker_proposal` denotes that native private proposal; the final system does not invoke a second worker LLM. Recursive schema validation rejects identity, roster, population-size, non-neighbor, global-state, and global-transcript keys.

For the reported interfaces, SWARMBENCH uses a 5×5 egocentric window and SILO-BENCH uses degree-three ports; the resulting trace list is bounded by those benchmark contracts.

Each incident trace contains:

```
      {channel, claim, novelty, support, conflict, ttl, direction?}
```

Channel handles are sampled from an episode-local pool, randomized per node, and fixed only within that episode. Incident records are seed-controlled shuffled at each local update. During SCD, incident permutations and opaque handle/claim relabelings are applied as paired, consistency-checked anonymous orbits to both the view and its teacher record. Equation 5 promotes semantic consistency across these paired anonymous views.

D.2 Decision schema

The controller emits, and the validator accepts, the following canonical schema:

```
{
  "task_action": string,
  "response": string | null,
  "deposits": [
    {channel, claim, novelty, support, conflict, ttl, direction?}
  ],
  "commit": {claim, confidence_bin} | null,
  "execution_intent": "PRIVATE" | "FIELD" | "WAIT"  // optional; omission = AUTO
}
```

Numeric trace components use five bins (0–4); TTL uses 1–8; confidence uses 0–4. Canonical serialization includes `execution_intent` only for a non-`AUTO` request; omission denotes the fixed default compiler rule. The validator derives the mode after checking consistency with the local view. In particular, a relay must match incident evidence, a challenge must contain positive conflict, and an abstaining wait cannot write or commit.

For analysis only, the validator assigns the first matching label in the precedence order in Table 12; labels are not controller outputs.

Table 12: Precedence for assigning derived analysis modes.

Mode	First matching condition
Challenge	Valid positive-conflict write
Synthesize	Commitment or response
Deposit	Fresh private-evidence write
Relay	Write matching an incident trace
Explore	Non-wait execution field with no higher label
Abstain	Legal wait with no write or commitment

D.3 Anonymous controller prompt

All nodes use the same instruction. Benchmark-specific action names and the serialized local view are appended mechanically.

> You are one anonymous member of a changing local network. Use only the private proposal, private evidence, current commitment, and semantic traces on incident channels. Return one canonical JSON decision. Continue useful private work unless local evidence justifies a deposit, relay, challenge, or commitment. Treat repeated support as local evidence rather than global consensus. Fresh verified conflict may reopen a commitment. Do not infer an identity, role, roster, population size, or global transcript.

E Proof of Conditional Local-to-Global Relabeling Equivariance

E.1 Synchronous execution

We make the actions in Proposition 1 explicit. The relabeling Γ has node-bijection component σ and transports the graph and all node-indexed state, consistently renaming every schema-designated opaque reference, including reference-bearing action arguments; it does not change topology, population size, reference-free task semantics, directions, or trace values. Its induced $g^t_{\Gamma,i}$ may also permute

incident-record presentation, and we define $(\Gamma D^t)_{\sigma(i)} = g^t_{\Gamma,i} d^t_i$. The map $\pi^{\mathrm{dep}}_\theta$ includes decoding, validation, repair, and conservative projection, so its equivariance concerns the complete admitted record. Node- or edge-indexed random draws are carried to their Γ-images, whereas label-free draws are shared; denote this coupling by $\xi^t_\Gamma = \Gamma\xi^t$. All components of D^t are computed from the same pre-update state S^t before F is applied.

Proof of Proposition 1. The initial states obey $S^0_\Gamma = \Gamma S^0$ by construction. Suppose $S^t_\Gamma = \Gamma S^t$. For every node i, view and local-map equivariance give

$$\begin{aligned}(d^t_\Gamma)_{\sigma(i)} &= \pi^{\mathrm{dep}}_\theta\big(\mathrm{View}_{\sigma(i)}(S^t_\Gamma)\big)\\ &= \pi^{\mathrm{dep}}_\theta\big(g^t_{\Gamma,i}\,\mathrm{View}_i(S^t)\big) = g^t_{\Gamma,i} d^t_i = (\Gamma D^t)_{\sigma(i)}.\end{aligned}$$

Thus $D^t_\Gamma = \Gamma D^t$. Applying the coupled synchronous update yields

$$\begin{aligned}S^{t+1}_\Gamma &= F(S^t_\Gamma, D^t_\Gamma; \xi^t_\Gamma)\\ &= F(\Gamma S^t, \Gamma D^t; \xi^t_\Gamma) = \Gamma F(S^t, D^t; \xi^t) = \Gamma S^{t+1}.\end{aligned}$$

Induction proves both identities for every round. Therefore the two trajectories satisfy $\tau_\Gamma = \Gamma\tau$, and every J with $J(\Gamma\tau) = J(\tau)$ obeys $J(\tau_\Gamma) = J(\Gamma\tau) = J(\tau)$. □

Proof of Corollary 1. The original and relabeled deterministic executions receive the same initial state and exogenous inputs, so they coincide. Proposition 1 also identifies the relabeled trajectory with the Γ-image of the original. Hence every state and joint decision is Γ-invariant. A unique selected node must therefore be fixed by Γ. □

With independently resampled draws, the conclusion holds in distribution when their conditional law is equivariant under the corresponding state and history relabeling. Stochastic decoding analogously requires an equivariant kernel for the complete admitted record.

E.2 Event-driven extension

The local-to-global argument also applies to event-driven execution. We formulate an asynchronous runtime contract here; the primary benchmark executions and training transitions retain their synchronous construction. Appendix P.5 separately tests execution perturbations. Let the complete execution state be

$$X^k = (S^k, M^k, P^k, T^k, R^k), \qquad X^{k+1} = \mathcal{U}(X^k, e_k; \xi_k).$$

Here S^k contains task and local coordination state, M^k contains buffered and in-flight messages, P^k contains pending calls and their saved input snapshots, T^k contains local clocks and expiry timers, and R^k contains active membership, the contact graph, node attributes, and routing state. An event e_k starts or completes a local call, delivers or drops a message, advances a timer or expires a trace, or changes membership or routing; ξ_k contains the associated runtime randomness. The index k orders events for analysis, not through a clock or coordinator available to the agents. A pending call retains its invocation snapshot even when other events change the current local view before it completes.

Relabeling and completion-time admission. A legal Γ consistently renames a universe of possible node incarnations, including future arrivals, and all schema-designated opaque references. It transports message endpoints, pending-call descriptors and snapshots, timer owners, node attributes, and membership/routing records. Reference-free task content and numerical attributes are unchanged in value and move with their associated nodes or records. Thus the active sets obey $V^k_\Gamma = \sigma(V^k)$ in paired executions, although V^k may vary with k. In the asynchronous contract, a completed output is rechecked against the current local state before taking effect: targets must remain valid, required evidence must remain admissible, and the native action must remain legal. This completion-time admission, including cancellation or rejection after departure or expiry, must commute with Γ. Cancelled or rejected calls are represented by a common symbol $\perp$ fixed by relabeling.

Proposition 2 (Conditional event-driven relabeling equivariance)**.** *Suppose all calls use the same equivariant local decode-and-admit map $\pi^{\mathrm{dep}}_\theta$ on their saved views. Suppose view construction and snapshot storage, event legality, completion-time admission, and every non-policy runtime operation commute with each legal relabeling. For any legal event history, couple the two executions by*

$$X^0_\Gamma = \Gamma X^0, \qquad e_{\Gamma,k} = \Gamma e_k, \qquad \xi_{\Gamma,k} = \Gamma\xi_k.$$

Then $X_\Gamma^k = \Gamma X^k$ for every event index k, and corresponding completed calls produce consistently relabeled admitted records, including $\perp$. Consequently, any relabeling-invariant functional of the event trajectory has the same value in the two executions.

Proof. At call initiation, equivariant view construction and storage produce corresponding saved snapshots. At completion, local-map equivariance gives corresponding outputs from those snapshots, and current-state admission preserves this relation for both accepted and rejected records. Delivery, dropping, timer updates, and membership/routing changes commute with relabeling by assumption. Composing the operations for each event therefore gives

$$\mathcal{U}(\Gamma X, \Gamma e; \Gamma \xi) = \Gamma \mathcal{U}(X, e; \xi).$$

The initial states are paired by hypothesis. If $X_\Gamma^k = \Gamma X^k$, equivariant event legality makes the paired event admissible, and

$$\begin{aligned} X_\Gamma^{k+1} &= \mathcal{U}(X_\Gamma^k, e_{\Gamma,k}; \xi_{\Gamma,k}) \\ &= \mathcal{U}(\Gamma X^k, \Gamma e_k; \Gamma \xi_k) \\ &= \Gamma \mathcal{U}(X^k, e_k; \xi_k) = \Gamma X^{k+1}. \end{aligned}$$

Induction proves the state and completed-record identities on every finite event prefix, hence on the event trajectory $\tau_\Gamma = \Gamma \tau$. Relabeling-invariant trajectory functionals are unchanged. □

Random event generation. The proposition pairs particular event histories. To obtain a distributional statement for the same stochastic execution mechanism, let H_k denote the complete history through X^k and let $\mathcal{K}(\cdot \mid H_k)$ be the conditional kernel of the next marked event $\zeta_k = (e_k, \xi_k)$. The joint kernel includes state-dependent timing and runtime randomness, so an equivariant event order alone is insufficient if the remaining random choices depend on arbitrary labels. Require, for every measurable set B of legal marked events,

$$\mathcal{K}(\Gamma B \mid \Gamma H_k) = \mathcal{K}(B \mid H_k).$$

Terminated histories may be padded by an absorbing, label-free event. Couple successive marked events by relabeling. The kernel condition and Proposition 2 inductively give equal relabeling-mapped prefix laws, and therefore

$$\mathcal{L}(\tau \mid \Gamma X^0) = \Gamma_* \mathcal{L}(\tau \mid X^0),$$

where Γ_* denotes pushforward of the trajectory law. For stochastic decoding, require the complete-record kernel to be equivariant as well and include its sampled outcomes in the marks; this supplies the corresponding local-output coupling in the same argument. Node speeds and capabilities need not be identical: their attributes travel with their nodes, and the event kernel respects that transport.

Delays, loss, churn, and symmetry. Variable latency changes completion or delivery events while pending snapshots remain in P^k. Message loss is an explicit drop or the absence of delivery in the paired history. Churn relabels arrivals and their initialization, departures, and the affected pending messages, calls, and routes together. These are two namings of the same event process, not a comparison between different schedules or different membership histories. No delay bound or scheduling fairness is needed for this structural equality at matched event indices. It does not imply schedule independence, task convergence, or successful completion when information fails to arrive. The symmetry-preservation argument of Corollary 1 applies pathwise only when the realized schedule and other exogenous inputs also preserve the symmetry. An equivariant random scheduler can preserve symmetry in distribution while an individual realization breaks it by activating one of two symmetric nodes first.

F Training Details

F.1 Decision loss, corpus, and split construction

For a serialized teacher record $d = (d_1, \ldots, d_{|d|})$, the sequence negative log-likelihood used in Equation 4 expands over assistant tokens as

$$\ell_{\mathrm{tok}}(\theta; \ell, d) = -\sum_{r=1}^{|d|} \log p_\theta(d_r \mid \ell, d_{<r}). \tag{9}$$

The decision corpus contains 6,400 synthetic and 512 train-only native local views. The resulting 6,912-view training split allocates 3,456 views to each substrate. Validation and test each contain 800 synthetic views. The 3,456 SILO-BENCH training views are task-stratified across all three benchmark levels; the six-task lower-scale anchor in Table 5 is evaluation-only and does not define the training task set. Each synthetic curriculum bundle contains one independently instantiated example from each of eight local-situation classes: *private-only progression* (a legal private proposal without usable incident evidence or a commitment event), *unresolved/insufficient coordination evidence* (traces lack sufficient support or coherence for commitment, with no verified conflict or useful new relay), *fresh private evidence* (new, locally verifiable evidence absent from the incident field), *admissible incident relay* (useful, unexpired evidence relayable without increasing trace components), *pre-commit verified conflict* (verified counterevidence to an uncommitted candidate), *commitment contradiction/reopening* (verified counterevidence incompatible with an existing commitment), *readiness-triggered synthesis* (local readiness or constructibility satisfied without active conflict), and *safe abstention/wait* (no warranted social update or commitment, or a required legal wait). These are input-situation classes, distinct from the six validator-derived mode labels. Bundles, rather than individual examples, are assigned to splits before teacher labeling. This grouping controls scenario overlap across splits but does not define a temporal trajectory; adjacent scenarios from these bundles never contribute to $\mathcal{L}_{\mathrm{field}}$.

Teacher calls use GPT-4o at temperature 0.2. Only canonical JSON records that pass batch-level schema validation enter $\mathcal{D}_{\mathrm{dec}}$. Native decision views and all sequential rollouts use collection seeds disjoint from formal evaluation.

F.2 ANONYMOUS-ORBIT CONSTRUCTION

For each decision example and epoch, we sample a view-local bijection over opaque channel and claim identifiers together with an independent incident-record permutation. The mapping is applied consistently to every schema-designated identifier occurrence in the private proposal and evidence, incident traces, local-event claim fields, commitment, and structured teacher record; free-form task content and native action tokens are unchanged. We retain the original example and its transformed partner in the same minibatch. A pair is admitted only when schema validation and all local freshness, relay, challenge, and target-matching relations agree with those of the source pair.

The identifier-free decision summary contains 12 factorized labels: task-action class, execution intent, derived semantic act, communication act, claim source, channel scope, novelty, support, conflict, TTL, commitment action, and commitment confidence. Every inapplicable factor has an explicit null class; let $\mathcal{A}$ index these factors. Orbit heads are linear projections of the final local-view hidden state. They provide training gradients only and are absent from the deployed checkpoint.

Writing $Q^{(k)}_{\theta,\psi}$ for factor k, the compact orbit objective in Equation 5 expands as

$$\mathcal{L}_{\mathrm{orbit}} = \mathbb{E}_{(\ell,d^\star),g}\Bigg[\ell_{\mathrm{tok}}(\theta; g\ell, gd^\star) + \frac{1}{|\mathcal{A}|}\sum_{k\in\mathcal{A}}\Big[\,\mathrm{CE}\Big(Q^{(k)}_{\theta,\psi}(\ell), \rho_k(d^\star)\Big) + \mathrm{CE}\Big(Q^{(k)}_{\theta,\psi}(g\ell), \rho_k(d^\star)\Big) + \beta D_{\mathrm{SKL}}\Big(Q^{(k)}_{\theta,\psi}(\ell), Q^{(k)}_{\theta,\psi}(g\ell)\Big)\,\Big]\Bigg]. \tag{10}$$

The first term reuses the transformed teacher record; it requires no additional teacher call. The two cross-entropies constitute $\mathcal{L}_{\mathrm{sup}}$ in the compact notation, while the last term aligns the corresponding factor distributions, with $D_{\mathrm{SKL}}(p,q) = \frac{1}{2}D_{\mathrm{KL}}(p\|q) + \frac{1}{2}D_{\mathrm{KL}}(q\|p)$.

F.3 SEQUENTIAL TRANSITION COLLECTION

After fitting the common behavior warm start with $\mathcal{L}_{\mathrm{dec}}$ only, we execute complete episodes with the fixed benchmark adapters and runtime on training-only seeds, covering Pursuit, Foraging, and SILO-BENCH. Immediately before an active node decodes, we store ℓ_i^t; after validation we store the admitted d_i^t; and after the synchronous batch is realized we store the next view ℓ_i^{t+1} for the same node. Terminal records without a well-defined next local update are masked. The field target is computed deterministically from ℓ_i^{t+1} using Equation 6. No synthetic curriculum adjacency contributes to this corpus.

Table 13: SCD configuration shared across matched training arms.

Component	Value
Teacher / student	GPT-4o (temperature 0.2) / Qwen3-4B-Instruct-2507
Initial policy	Common semantic-record warm start
	SWARMBENCH Pursuit/Foraging;
Decision-task coverage	SILO-BENCH
Adaptation	LoRA rank 32, alpha 64, dropout 0.05
Target modules	q/k/v/o and gate/up/down projections
Decision train / validation	6,912 / 800 local views
Transition train / validation	14,256 / 4,752 adjacent triplets
Eventful transition fraction	0.50
Stage-two budget	648 optimizer updates (≈ 1.5 source-data passes)
λ_{orbit} / λ_{field}	0.05 / 0.05
λ_{state}	1.0 (equal histogram/state group weight)
Orbit consistency coefficient β	0.10
Auxiliary warm-up / ramp	8 / 32 updates
Optimizer	AdamW; weight decay 0.01
Adapter / auxiliary learning rate	2×10^{-6} / 5×10^{-4}
Effective global batch	16 records (32 source/transformed sequences)
Precision / maximum context	bfloat16 / 16,384 tokens
Checkpoint selection	Minimum validation decision loss
Deployment	Greedy decoding; auxiliary heads removed
Hardware	Two NVIDIA H800 GPUs

Hist_n, Hist_s, and Hist_x are six-way normalized histograms with one null class and bins 0–4; Hist_T has one null class and TTL bins 0–8. A view with no incident traces places all histogram mass on the null class. Readiness has two classes (not-ready/ready), and commitment has six classes (empty or confidence 0–4), so the target width is independent of population size and incident ordering and invariant to opaque aliases.

The predictor concatenates $h_\theta(\ell_i^t)$, the final current-view hidden state, with learned 32-dimensional embeddings of the factors in $\rho(d_i^t)$, applies a 512-dimensional GELU layer with dropout 0.05, and uses one linear output head per target. With $\mathcal{F} = \{n, s, x, T\}$ and $\mathcal{B} = \{r, \kappa\}$, the grouped loss in Equation 7 expands as

$$\mathcal{L}_{\text{field}} = \mathbb{E}_{(\ell_i^t, d_i^t, \ell_i^{t+1})\sim\mathcal{D}_{\text{tr}}}\Bigg[\frac{1}{4}\sum_{k\in\mathcal{F}} \text{CE}\Big(f_\omega^{(k)}(h_\theta(\ell_i^t), \rho(d_i^t)), \text{Hist}_k(\ell_i^{t+1})\Big) + \frac{1}{2}\sum_{k\in\mathcal{B}} \text{CE}\Big(f_\omega^{(k)}(h_\theta(\ell_i^t), \rho(d_i^t)), m_k(\ell_i^{t+1})\Big)\Bigg]. \tag{11}$$

Histogram terms use soft-label cross-entropy and state terms use categorical cross-entropy. The factors $1/4$ and $1/2$ average within the histogram and state families, respectively, giving the two groups equal total weight ($\lambda_{\text{state}} = 1$) rather than allowing the four histogram targets to dominate by count. Targets are extracted deterministically from the next view and require no teacher call.

We stratify the transition sampler into ordinary and eventful updates. The latter includes verified conflict, challenge, commitment, reopening, and expiry transitions. This makes local-field prediction sensitive to reorganization rather than being dominated by unchanged background fields.

F.4 OPTIMIZATION

The student is Qwen3-4B-Instruct-2507 (Yang et al., 2025a) with a rank-32 LoRA adapter (Hu et al., 2022). Stage one optimizes $\mathcal{L}_{\text{dec}}$ and supplies the shared behavior warm start used by every factorial arm. Stage two interleaves decision and transition minibatches and optimizes Equation 8. The two auxiliary heads share the adapted backbone but are removed after training. All factorial arms use minimum validation decision loss for checkpoint selection. Deployed equivariance diagnostics and frozen-probe field loss are computed only after selection, on data excluded from controller training and checkpoint choice; formal benchmark episodes remain untouched.

G SCD TRAINING AND REPLICATION PROTOCOLS

G.1 PRIMARY FACTORIAL AND DOWNSTREAM PROTOCOL

The factorial comparison uses five paired training seeds. Within a seed, all four cells share the same semantic-record warm start, source records, transformed pairs, adjacent transition corpus, deterministic minibatch schedule, token budget, and 648 optimizer updates. Each update uses the same source and transition record indices; orbit transforms follow the same deterministic seed schedule. Disabled losses are computed but multiplied by zero before backpropagation; no arm replaces an auxiliary loss with additional decision supervision. Checkpoints are selected solely by validation decision loss, after which structural diagnostics are evaluated on untouched held-out data.

Deployed equivariance diagnostics. For each held-out local view ℓ, we sample a non-identity anonymous transformation g whenever the orbit is nontrivial. The final greedy decoder (not the training-time orbit head) independently emits $d_\theta(\ell)$ and $d_\theta(g\ell)$. After inverse-mapping the transformed output, summary agreement compares their identifier-free decision summaries. Exact decoded-record equivariance instead requires equality of the complete canonical records; exact admitted-record equivariance repeats the test after the same validator-admission map is applied. The three held-out rates test increasingly strict agreement under incident permutations and fresh opaque-handle bijections.

Frozen-probe field loss. We delete the training-time field predictor and freeze the selected controller backbone for every factorial cell. For each frozen checkpoint, we then fit a fresh linear probe with the same architecture, probe-training transitions, preprocessing, initialization schedule, optimizer, and update budget. No probe is reused across controllers. Evaluation uses an independent held-out transition split, with the eventful subset defined by verified conflict, challenge, commitment, reopening, or expiry. We normalize all-transition and eventful losses separately by the corresponding DECISION ONLY value. This capacity-matched protocol tests field information accessible in the frozen deployed representation without rewarding a controller for having trained its original auxiliary head.

Table 3 consolidates the objective, deployment, and closed-loop diagnostics. Except for the two exact-equivariance rates, entries report mean $\pm$ sample s.d. over five paired training seeds. Train-head losses are raw optimization diagnostics; because those heads are removed at deployment, only deployed agreement rates and fresh frozen probes provide deployment-facing structural evidence.

The conventional-quality matrix compares the two endpoint arms, DECISION ONLY and Waggle, and contains 1,020 episodes: five training seeds and two arms evaluated on SILO-BENCH at $N = 64$ and on Pursuit and Foraging at $N \in \{32, 64\}$, with matched held-out cases in every task–population cell. Table 14 gives the resulting aggregates; its confidence bounds use 10,000 hierarchical paired-bootstrap resamples, first over training seeds and then over matched held-out cases within each task–population stratum. The separate late-counterevidence matrix evaluates all four factorial arms on Pursuit and SILO-BENCH III-26 at $N \in \{32, 64\}$ over ten matched held-out cases. Across five training seeds, this yields 800 episodes, or 200 matched settings per arm. Recovery is the fraction of these settings in which verified counterevidence reopens the old commitment and is followed by synthesis of a replacement supported by the new evidence. These settings provide the final three columns of Table 3. All downstream runs remove the auxiliary heads and use identical validators, transports, and native compilers.

Table 14: Matched conventional task quality under the same runtime; Δ denotes Waggle minus DECISION ONLY.

Benchmark	DECISION ONLY quality	Waggle quality	Δ quality	95% lower bound	Waggle/DECISION ONLY
SILO-BENCH	0.675	**0.711**	**+0.036**	0.007	1.053
SWARMBENCH	4.09	**4.43**	**+0.34**	0.06	1.083

Paired deployment and reorganization effects. Table 15 reports complete SCD relative to DECISION ONLY, separating uncertainty in the paired differences from the per-arm sample s.d. in Table 3. The structural endpoints use their respective held-out diagnostic sets; the behavioral endpoints use the late-counterevidence settings described above. The paired 95% intervals use 10,000 hierarchical bootstrap resamples: first resample the five paired training seeds, then matched held-out views or transitions for structural endpoints, or matched cases within task–population strata for

behavioral endpoints. Each resample preserves method pairing and recomputes the contrast using the original metric definitions and task weights.

Table 15: Paired deployment and reorganization effects of complete SCD. Δ is Waggle minus DECISION ONLY; agreement and recovery differences are in percentage points (pp). Endpoint definitions follow Table 3.

Endpoint	DECISION ONLY	Waggle	Paired Δ	Paired 95% CI
Summary agreement ↑	72.4%	**95.2%**	+22.8 pp	[+18.7, +26.6]
Eventful frozen-probe loss ↓	1.000	**0.608**	−0.392	[−0.435, −0.350]
Recovery ↑	80.0%	**88.5%**	+8.5 pp	[+2.0, +15.0]
Stale-field AUC ↓	22.768	**19.237**	−3.531	[−4.85, −2.15]
Post-conflict msg./agent ↓	7.157	**6.737**	−0.420	[−0.61, −0.23]

G.2 REPLICATION WITH A SECOND STUDENT BACKBONE

We repeat the endpoint comparison with Llama-3.1-8B-Instruct (Grattafiori et al., 2024), reusing the same GPT-4o teacher records, canonical interface, LoRA configuration, SCD coefficients, 648-update stage-two budget, checkpoint rule, and greedy deployment protocol. Only the tokenizer and chat template follow the Llama-native format; memory-preserving microbatching keeps the effective batch unchanged. From a common decision-distillation warm start, we train DECISION ONLY and complete SCD with five paired training seeds.

Structural evaluation uses the same 800 held-out local views and frozen-probe protocol. Behavioral evaluation reuses the late-counterevidence intervention on Pursuit and SILO-BENCH III-26 at $N \in \{32, 64\}$ across the five paired training seeds. The conventional-quality evaluation reports SILO-BENCH success at $N = 64$ over matched held-out cases. Table 16 shows that both the learned structure and its behavioral consequence replicate with the second student family: orbit agreement rises by 17.0 points, eventful probe loss falls by 34.0%, and recovery rises by 7.5 points while stale-field burden and post-conflict traffic both decrease.

Table 16: Replication of SCD with Llama-3.1-8B-Instruct; eventful field loss is normalized to DECISION ONLY.

Objective	Deployed orbit agreement (%) ↑	Eventful frozen-probe field loss ↓	SILO-BENCH $N = 64$ success ↑	Late-conflict recovery ↑	Stale-field AUC ↓	Post-conflict messages/agent ↓
DECISION ONLY	77.0	1.000	0.660	80.0%	23.0	7.25
Waggle	**94.0**	**0.660**	**0.673**	**87.5%**	**20.1**	**6.95**
Change	+17.0 pp	−34.0%	+0.013	+7.5 pp	−12.6%	−4.1%

H BENCHMARK PROTOCOL

H.1 SWARMBENCH

We evaluate all five released SWARMBENCH task families. The main matrix reports Pursuit, Foraging, and Synchronization at $N \in \{16, 32, 64\}$ over five matched seeds; Flocking and Transport are reported under the same protocol in Appendix C.1. The lower-scale anchor uses $N = 8$ and ten held-out seeds. The adapter gives all active agents a local snapshot from the same pre-update environment state in one batch each round and invokes the resident controller. Egocentric observations generate a private movement proposal. Incident edge signals are projected into binned traces with freshly randomized local channel handles. The runtime deterministically resolves the concrete native move from the validated record and local state, and emits at most one compact outbound trace per local decision. Admitted writes are staged with native actions and become incident evidence only after the subsequent environment update.

The fixed candidate-ready gate requires two or more incident channels carrying the same claim, total support at least eight, zero conflict, and support-weighted directional coherence at least 0.6. These are fixed adapter settings held constant across the formal matrix, not learned controller parameters. The gate counts incident channels rather than distinct evidence origins: it is deliberately a two-channel readiness test, not a distinct-origin or independent-source test.

The formal SWARMBENCH system combines:

Qwen3-4B LoRA + Waggle shared local controller
+ local-fingerprint duplicate suppression + per-hop-capped relay
+ fixed native action compilation + verified-conflict inhibition.

Pursuit refreshes moving spatial evidence after twelve rounds; Foraging treats each stationary spatial event as event-once evidence.

H.2 SILO-BENCH

The primary SILO-BENCH matrix evaluates 30 official tasks, with ten tasks in each of Levels I–III. Every method is run at $N \in \{16, 32, 64\}$ with five matched evaluation seeds, yielding 450 task–population–seed episodes per method for Table 1. At each population size, official success and receiver-delivered messages are macro-averaged over the complete task–seed set. Every episode has nine local communication rounds. A directed frame jointly carries one compact trace, proposal intent, and one task snapshot. The official case generator, task solver, partial-correctness criterion, and success metric are retained. Neither controller nor adapter receives a population roster or global transcript. SILO-BENCH does not invoke the spatial transport layer: the shared learned controller law writes through finite-lifetime native incident transport.

The communication-horizon sweep fixes $N = 64$ and evaluates the same task suite on eight held-out seeds at each $R \in \{3, 6, 9, 12, 18\}$. Each horizon therefore contains 240 matched task–seed units per method; task instances and seeds are held fixed across horizons so that the curves vary only the available interaction budget.

The six-task subset I-04/I-09, II-11/II-16, and III-23/III-26 is used only for the independent lower-scale evaluation anchor at $N \in \{4, 8\}$ with ten held-out seeds (Table 5); it defines neither the training corpus nor the primary matrix. Focused intervention experiments state their narrower task scope separately.

H.3 BASELINE IMPLEMENTATIONS

All methods use the same formal task instances, population conditions, evaluation seeds, benchmark-facing legal task-action spaces, evaluators, and, where applicable, interaction horizons. The interface-matched controls additionally share Waggle's local topology and private proposals. NOCOMM suppresses explicit outbound inter-agent messages; NATIVEMSG exchanges transient task-native messages over the same local contacts, without a separate persistent semantic field, quorum, or commitment mechanism; and FIXEDSWARM replaces the learned law with a shared hand-coded local rule using finite-lifetime local signals and fixed support, inhibition, quorum, reopening, decay, and expiry.

Broader baselines instead retain their native information-flow and private-computation interfaces subject to the configurations below. A benchmark-native private proposal is reused only when doing so leaves the baseline unchanged; otherwise the method receives the same underlying private task observation and legal action space. Adapters map these inputs into the native organizational loop and map task-facing decisions back to benchmark-native actions or final answers; they do not replace the baseline's graph construction, agent composition, routing, activation, or role mechanism with Waggle's runtime. Consequently, native differences in model calls remain part of each method, while receiver-delivered messages measure communication rather than total inference cost; model-call and token telemetry is recorded separately. No method receives auxiliary calls or evaluation-answer access outside this recorded execution.

We use a pinned release from the authors when it supports the target substrate; otherwise Table 17 marks a clean-room reimplementation based on the paper and released artifacts; exact commits and integration changes are recorded in the artifact. Published defaults are used when transferable. Other substrate-specific choices use a disjoint development split and one selection metric per benchmark family, never formal evaluation seeds; they are not tuned per seed or population unless the original method explicitly conditions on population size. Model parameters, hyperparameters, checkpoints, and selection rules are fixed before evaluation. Algorithm-specified organizational state may still change within an episode and is reset between episodes.

Coverage of organizational adaptation. The broader comparisons span optimized information-flow graphs (GPTSWARM), task-conditioned topology (G-DESIGNER), joint crew and topology generation (ARG-Designer), within-execution graph adaptation (AGENTNET), step-wise activation

(DMoA), and adaptive profile-based allocation (SWARMSYS). These complement the interface-matched controls and the distillation baseline. Table 17 distinguishes development-time optimization, per-task generation, and within-execution adaptation, and records the mechanisms retained under benchmark integration. In particular, our GPTSWARM topology is selected on development cases and frozen for formal evaluation, whereas AGENTNET retains within-execution graph updates.

Related frameworks outside the quantitative matrix include DyLAN, with contribution-based team selection and dynamic communication (Liu et al., 2024), and AgentVerse, with feedback-driven adjustment of expert composition (Chen et al., 2024b). CAMEL emphasizes role-prompted dialogue (Li et al., 2023). OpenAI Swarm provides an agent-and-handoff orchestration interface (OpenAI, 2024); a quantitative baseline would require specified task-level prompts, tools, and handoff rules.

Agent Distillation (AGENTDISTILL). We use the authors' official Agent Distillation implementation (Kang et al., 2025) in a budget-matched configuration. Qwen3-4B-Instruct-2507 is retrained from the same base checkpoint on GPT-4o reason–action–observation trajectories from the training splits of SWARMBENCH Pursuit/Foraging and SILO-BENCH, with no Flocking or Transport training. We retain the published trajectory SFT loss, masking observation tokens, and train rank-64 LoRA adapters on all linear layers for two epochs with batch size eight and learning rate 2×10^{-4}. One resulting checkpoint is shared by all N agents. No formal-evaluation seeds are used for tuning or selection.

The primary comparison disables first-thought prefix (FTP) and self-consistent action generation (SAG), using one greedy candidate and one student call per active-agent update. External retrieval and code-tool calls are both zero. Agents exchange transient benchmark-native messages without Waggle's persistent field, quorum, commitment, or reopening operations. Communication is counted as receiver-delivered messages per agent; Appendix H.5 reports deployment calls and relative token counts separately, excluding offline teacher generation and training.

Table 17: Implementations of broader baselines in the primary matrix.

Method	Native object	Benchmark integration	Retained mechanism	Implementation / tuning
GPTSWARM	Computational DAG of operation nodes and information-flow edges.	Map proposal-producing agents or worker operations to DAG nodes; route over legal contacts; translate benchmark actions or final answers at the boundary.	Edge-probability optimization and topological execution.	Clean-room; select a population-matched topology on disjoint development cases, then freeze it for formal evaluation.
G-DESIGNER	Task-conditioned communication graph over agents and a virtual task node.	Map task information to node features; decode a topology over legal contacts; translate exchanged content and benchmark actions or final answers.	Graph encoder/decoder, virtual task node, and learned topology generation.	Pinned released code; published defaults or disjoint-development selection; frozen parameters with per-task topology decoding.
ARG-Designer	Autoregressively generated agent composition and communication topology.	Map task information and the available agent pool to the graph generator; execute the generated collaboration graph within the benchmark-legal interaction space.	Task-conditioned autoregressive generation of crew composition, roles, and communication links.	Pinned released code; published defaults; frozen generator with per-task graph generation.
AGENTNET	Decentralized, dynamically evolving agent DAG.	Feed benchmark observations and outcomes into the native execution loop; routing evolves within the legal interaction space.	Forward/Split/Execute routing, local memory, and outcome-based edge updates.	Clean-room; fixed defaults across populations; within-execution graph updates retained and reset per episode.
DMoA	Sparse step-wise activation over an expert-agent pool.	Give the router benchmark-valid context and history; selected agents return benchmark actions or final answers through the benchmark boundary.	Context/history-conditioned recurrent routing and sparse activation.	Clean-room; published defaults or disjoint-development selection; frozen router with per-step activation retained.
SWARMSYS	Explorer–Worker–Validator population with adaptive profiles.	Map benchmark events into the native role loop; profile matching allocates work; translate benchmark actions or final answers.	Specialized roles, profile–event matching, and validation-driven reinforcement.	Clean-room; fixed defaults across populations; online matching and validation feedback retained and reset per episode.
AGENTDISTILL	Distilled reason–action–observation policy.	Shared checkpoint across N agents; local observations, native task actions, and transient inter-agent messages.	Trajectory SFT with observation masking; one greedy call per active-agent update.	Authors' official code; train-only retraining; budget-matched configuration with FTP/SAG and external tools off; no evaluation tuning.

H.4 PAIRED UNCERTAINTY

Unless a caption states otherwise, reported paired confidence intervals use 10,000 bootstrap resamples while keeping the compared methods coupled within a matched unit. Fixed-controller benchmark contrasts resample matched task–seed units within benchmark strata. Comparisons spanning independently trained controllers instead use the hierarchical procedure specified in Appendix G.1: training seeds are resampled first and matched environment seeds second within task–population

strata. Key paired effects are $+0.07$ (95% CI $[+0.02, +0.12]$) for Waggle versus DECISION ONLY on AgentsNet strict solved fraction, $+0.06$ ($[+0.01, +0.11]$) versus FIXEDSWARM on the same endpoint, and -0.017 ($[-0.029, -0.005]$) for Joint minus the oracle specialist on SILO-BENCH success at $N = 64$.

Table 18: Paired uncertainty for the primary scaling contrasts. The comparator has the strongest non-Waggle point estimate for each endpoint and population in Table 1. Δ is Waggle minus comparator; SILO-BENCH differences and intervals are in percentage points (pp).

Endpoint	N	Comparator	Score	Waggle	Δ	95% CI
SILO-BENCH SR (%)	16	AGENTDISTILL	79.0	86.3	$+7.3$	$[+3.1, +11.5]$
	32	FIXEDSWARM	69.7	79.0	$+9.3$	$[+4.7, +13.8]$
	64	AGENTDISTILL	43.7	58.2	$+14.5$	$[+8.2, +20.5]$
SWARMBENCH norm.	16	AGENTDISTILL	0.439	0.540	$+0.101$	$[-0.01, +0.21]$
	32	AGENTDISTILL	0.425	0.546	$+0.121$	$[+0.01, +0.23]$
	64	ARG-Designer	0.880	1.041	$+0.161$	$[+0.05, +0.28]$

Table 18 retains the primary task weights and frozen normalization anchors. At each population, SILO-BENCH contributes $30 \times 5 = 150$ matched task–seed episodes and SWARMBENCH contributes $3 \times 5 = 15$ task–seed results, with five seeds within each task. The 10,000 paired-bootstrap resamples preserve the method pairing and task weights. These fixed-checkpoint contrasts concern evaluation uncertainty rather than variation across training runs. All three SILO-BENCH intervals lie above zero; the SWARMBENCH interval at $N = 16$ includes zero.

In Table 1, Avg. rank averages ranks over 12 equally weighted endpoints: three population-specific SILO-BENCH success rates and nine raw SWARMBENCH task scores, with ties averaged. SR/msg. pairs are ranked by SR; bold/underline duplicate the first/second-place colors. Rankings indicate point-estimate order, not statistical significance.

H.5 INFERENCE ACCOUNTING

Table 19 separates deployment inference from receiver-delivered communication on SILO-BENCH at $N = 64$. Calls are reported as absolute episode averages; input and output token totals are each normalized to Agent Distillation. These deployment figures exclude offline teacher generation and training and are not measurements of FLOPs or wall-clock time.

Table 19: Deployment inference and communication on SILO-BENCH, $N = 64$. Token ratios use Agent Distillation as the reference ($1.00\times$); LLM calls are absolute counts per episode.

Method	SR (%) ↑	LLM calls / episode ↓	Input tokens (relative) ↓	Output tokens (relative) ↓	Msg./agent ↓
Agent Distillation	43.7	576	$1.00\times$	$1.00\times$	8.50
DMoA	21.5	806	$1.47\times$	$1.34\times$	3.60
SWARMSYS	26.7	864	$1.62\times$	$1.48\times$	9.00
Waggle	58.2	577	$1.13\times$	$1.09\times$	8.03

Waggle uses nearly the same number of calls as Agent Distillation, with 13% more input tokens and 9% more output tokens, while improving success by 14.5 percentage points. DMoA delivers fewer messages but requires more calls and tokens. Communication savings therefore do not imply proportional inference savings. For Waggle, A active-agent updates require $C = A + C_{\text{regen}}$ controller calls, where $0 \le C_{\text{regen}} \le A$ under the one-regeneration limit (Appendix C.3). A no-message decision still requires inference, and bounded local input does not make total population compute independent of N.

I FROZEN TRANSFER TO RANDOMIZED-ANONYMOUS AGENTSNET

This experiment asks whether a controller learned only from spatial and distributed-language substrates transfers, without parameter updates, to an unseen graph-structured coordination substrate. We construct AGENTSNET-RP from the released AgentsNet benchmark (Grötschla et al., 2025). The

controlled adaptation retains the graph instances, five task definitions, paper-specified round budgets, and official task evaluators, while replacing persistent node names with opaque incident handles and supplying private episode-local randomness for symmetry breaking. This setting tests transfer of the learned law through a new deterministic adapter. AgentsNet is distinct from the singular AGENTNET system baseline used in the primary comparison matrix.

Protocol. Formal evaluation covers COLORING, CONSENSUS, LEADERELECTION, MATCHING, and VERTEXCOVER on the official SMALLWORLD, SCALEFREE, and DELAUNAY graph families. We use the released $N \in \{8, 16\}$ instances, with three graphs for each size–topology pair, yielding

$$5 \text{ tasks} \times 3 \text{ families} \times 2 \text{ sizes} \times 3 \text{ graphs} = 90 \tag{12}$$

matched settings per method. Released $N = 4$ graphs are used only to validate schema legality, task-action mapping, and propagation of the priority encoding; they are excluded from LoRA training, SCD transition collection, loss-weight selection, checkpoint selection, and performance-driven prompt or threshold tuning.

We follow the communication horizons specified in AgentsNet. Coloring, Matching, and VertexCover use five synchronous rounds at $N = 8$ and six at $N = 16$; LeaderElection and Consensus use $2D + 1$ rounds for graph diameter D. Each node accumulates non-private information only through incident messages, as in the benchmark's randomized-LOCAL-inspired protocol.

Episode-local randomized anonymity. At the start of each episode, node i receives

$$u_i \sim \text{Uniform}\big(\{0, \ldots, 2^{64} - 1\}\big), \tag{13}$$

with collisions resampled so that priorities are unique within that episode. Each node initially sees only its own u_i. Priorities are fixed within an episode, resampled between episodes, never exposed as a roster, and available to another node only after ordinary incident-local propagation. All methods share the same graph and priority realization in each matched setting. The task contract states only that smaller priorities may serve as episode-local tie breakers. The runtime validates and executes the controller's legal task action; it neither propagates priorities automatically nor selects a color, partner, coordinator, leader, or consensus value. Priority therefore provides private randomness rather than persistent identity or an assigned role.

The same contract supports stable retreat from adjacent coloring conflicts, partner contention in matching, and redundant coordinators in vertex cover; LeaderElection and Consensus may propagate the minimum observed priority as a common anchor. These uses are controller decisions rather than runtime rules.

Frozen transfer and comparisons. All model parameters, validator behavior, and decoding parameters are frozen before any AgentsNet instance is accessed. No AgentsNet teacher record, rollout transition, or evaluation answer contributes to training or checkpoint selection, and the training-only SCD heads are absent at deployment. A deterministic adapter exposes the task contract, legal task-action vocabulary, opaque incident handles, received traces, a remaining-round bin, and the focal node's private priority through the canonical local interface. It maps validated decisions to the official answer format without choosing the solution.

We compare frozen Waggle with two shared-controller variants: Qwen3-4B zero-shot applies the unadapted base checkpoint to the same local prompt and decision schema without distillation, while DECISION ONLY removes orbit and field supervision. The interface-matched controls are NATIVEMSG, which sends transient task-native messages without a persistent field; FIXEDSWARM, which applies fixed support, inhibition, quorum, and decay; and NOCOMM. Every method shares graphs, priorities, contracts, round budgets, legal actions, and final evaluators. The primary endpoint is official strict solved fraction, macro-averaged across tasks. We also report task-wise scores using the official soft evaluators and receiver-delivered messages per agent. Paired uncertainty resamples matched task–graph settings while stratifying by graph size and topology.

In Table 2, scale retention is the $N{=}16$ to $N{=}8$ strict-solved ratio. Avg. rank averages ranks over five task scores and two size-specific strict-solved endpoints, with equal weights and averaged ties. Bold/underline repeat the color ranking; traffic ranks exclude NOCOMM.

Results. Table 2 shows that frozen Waggle leads on four of five task-specific soft scores and on the strict Overall endpoint; FIXEDSWARM is strongest only on VertexCover. Within the shared-controller family, overall solved fraction rises from 0.48 zero-shot to 0.54 after decision distillation and 0.61 with

Table 20: LeaderElection with and without episode-local priority; all else fixed.

Variant	$N = 8$	$N = 16$	Overall
Waggle without priority	0.61	0.39	0.50
Waggle with priority	**0.89**	**0.71**	**0.80**

complete SCD; FIXEDSWARM reaches 0.55. The corresponding matched effects are approximately $+0.07$ (95% CI $[+0.02, +0.12]$) against Decision only and $+0.06$ (95% CI $[+0.01, +0.11]$) against FIXEDSWARM. The Waggle–Decision gap is positive at both scales and widens from 0.05 at $N = 8$ to 0.09 at $N = 16$. Relative to NATIVEMSG, Waggle reduces delivered communication by $1 - 11.1/28.4 = 60.9\%$; NOCOMM confirms that private priority alone does not solve these graph-level tasks.

Table 20 isolates the randomized-anonymity contract on the task that explicitly requires symmetry breaking. Episode-local priority raises LeaderElection from 0.50 to 0.80 overall, without changing the learned controller or runtime. Consistent with Corollary 1, the gain concentrates on symmetry-obstructed settings ($+0.49$) and is much smaller when the initial attributed graph already distinguishes a candidate ($+0.06$; Figure 7). Taken together, the experiment shows that one frozen local law transfers through a deterministic adapter to a new graph distribution and action vocabulary, with the advantage of complete SCD persisting at the larger graph size.

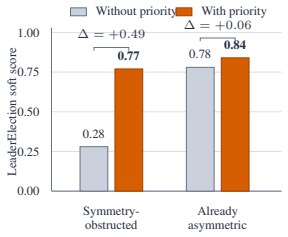


Figure 7: **LeaderElection with and without episode-local priority, stratified by initial symmetry.** Symmetry-obstructed settings have no singleton candidate orbit after priority removal and opaque-name quotienting (10/18); the remaining eight are already asymmetric. Bars show official soft score.

J GPT-4O PRIVATE-WORKER TRANSFER

Table 21 compares the quality–compute operating points of the comparison methods under a common full-refresh schedule. The comparison keeps the private worker and refresh schedule fixed across methods, isolating the operating point of coordination rather than a schedule change. A separate within-Waggle comparison in Table 22 removes 41.1–43.4% of logical GPT-4o input through private-state reuse while preserving or improving mean quality.

Table 21: GPT-4o full-refresh transfer at $N = 32$: mean native quality ↑ / logical input kTokens ↓ / delivered messages per agent ↓.

Method	Pursuit	SILO-BENCH II-16	SILO-BENCH III-26
Waggle (full refresh)	4.33 / 1867.0 / 22.77	1.000 / 406.5 / 7.62	0.794 / 742.6 / 7.12
NATIVEMSG	1.40 / 1856.3 / 167.88	1.000 / 210.3 / 9.00	1.000 / 411.8 / 9.00
FIXEDSWARM	4.60 / 1860.0 / 29.36	1.000 / 210.3 / 9.00	1.000 / 411.8 / 9.00
G-DESIGNER	3.40 / 1720.0 / 10.50	1.000 / 285.0 / 6.10	0.900 / 505.0 / 5.90
DMoA	3.65 / 1480.0 / 8.90	1.000 / 235.0 / 5.40	0.950 / 425.0 / 5.20
AGENTNET	3.00 / 1859.9 / 8.38	—	—
SWARMSYS	—	0.388 / 136.1 / 9.00	0.475 / 268.5 / 9.00

Table 22: Private-state reuse within Waggle at $N = 32$: mean native quality / logical input kTokens / delivered messages per agent.

Task	Full refresh	Matched reuse	Input reduction
SILO-BENCH II-16	1.000 / 406.5 / 7.62	1.000 / **239.3** / 7.58	41.1%
SILO-BENCH III-26	0.794 / 742.6 / 7.12	0.957 / **420.6** / 7.27	43.4%

Figure 8 shows the full-refresh reference trade-off. Marker area encodes receiver-delivered messages per agent, so a point farther up and left with a smaller marker is preferable. Logical input tokens count every executed private-worker request independently of cache use.

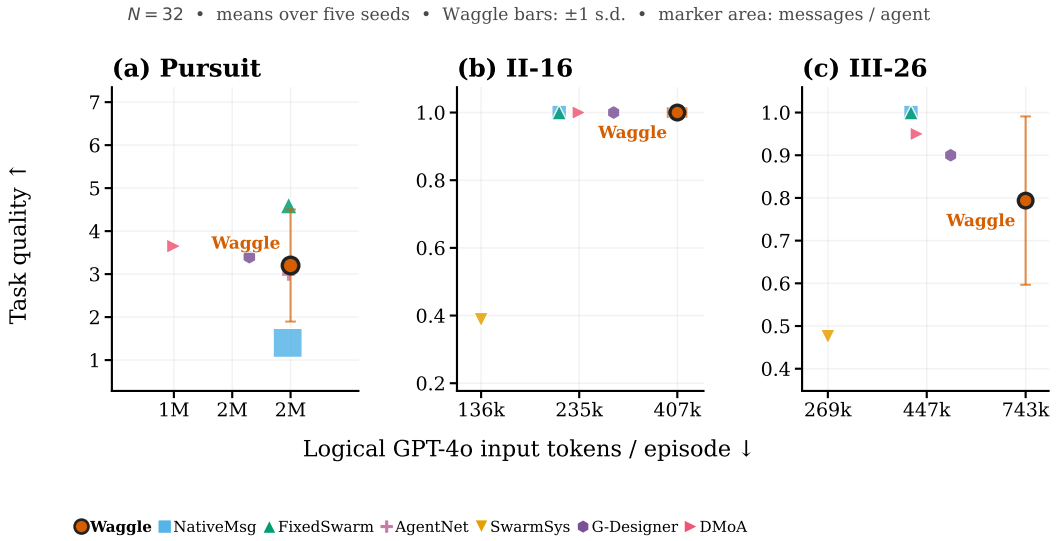


Figure 8: **Full-refresh GPT-4o quality–logical-input trade-off at $N = 32$.** Means use five matched seeds; marker area encodes delivered messages per agent, and Waggle error bars show ± 1 sample s.d.

K Focused Paired Evidence

K.1 Matched private-state reuse

The matched schedule retains the initial private solve and every communication-conditioned packet update, but reuses each agent's latest private state for redundant controller refreshes. The same-seed comparison is reported in Table 22; it separates this within-Waggle schedule change from the common full-refresh baseline comparison in Table 21.

K.2 Eight-seed paired evaluation

The focused paired analysis uses eight held-out seeds for the principal contrasts. The complete eleven-method five-seed breadth matrix remains in Table 1; Figure 9 concentrates the added evidence on population behavior and private-compute efficiency.

L Joint Law versus Substrate-Specific Specialists

All controllers in this comparison start from Qwen3-4B-Instruct-2507 and use the same rank-32 LoRA architecture, target modules, dropout, GPT-4o teacher, anonymous local prompt, canonical decision schema, complete SCD objective, optimizer, update count, checkpoint-selection rule, greedy decoding, validator, and fixed benchmark runtime. Thus the comparison changes the allocation of decision-labeled local views across substrates, rather than the controller interface or the runtime that realizes records.

Formal evaluation fixes $N = 64$ and uses eight held-out seeds, yielding 16 matched SwarmBench Pursuit/Foraging task–seed units (two tasks $\times$ eight seeds) and 240 matched Silo-Bench task–seed units per deployment. For the deliberately favorable oracle comparison, we pre-specify a practical tolerance of -0.030 for the Joint-minus-oracle Silo-Bench success difference and a minimum retention ratio of 0.950 for the Joint-to-oracle SwarmBench score. The paired Silo-Bench interval uses 10,000 bootstrap resamples over the matched task–seed units; the SwarmBench contrast is reported descriptively because it contains only 16 units.

Table 23 reports the scores, retention ratios, and deployment counts summarized in Figure 3(d).

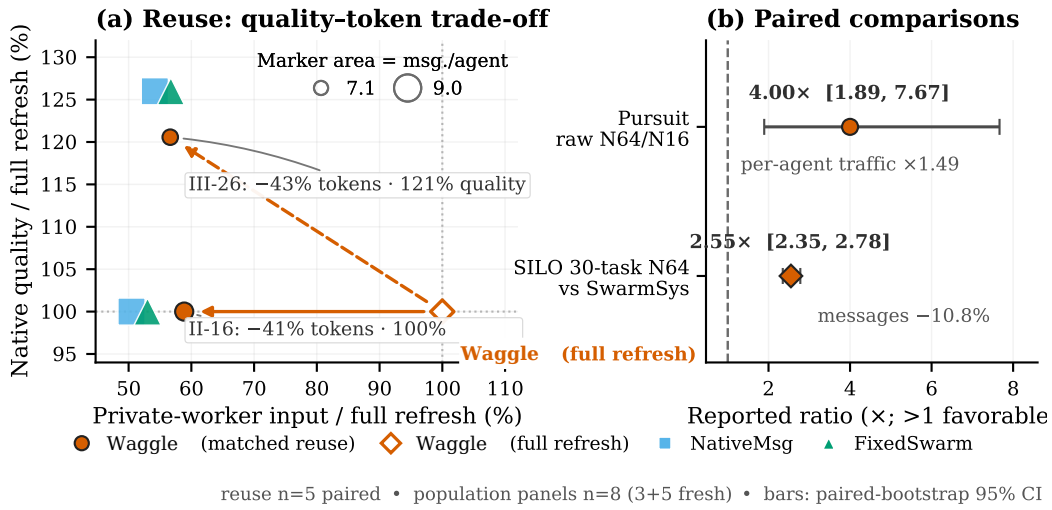


Figure 9: **Focused paired evidence.** (a) Matched private-state reuse versus full refresh; marker area encodes delivered messages per agent. (b) Eight-seed paired ratios with paired-bootstrap 95% intervals.

Table 23: Joint and substrate-specific SCD deployments at $N = 64$; retention is relative to the oracle pair.

Deployment	SWARMBENCH P/F score / oracle retention ↑	SILO-BENCH SR / oracle retention ↑	Deployed LoRA adapters ↓
Joint Waggle	**1.133/96.8%**	**0.682/97.6%**	**1**
Exposure-matched specialist pair	1.118/95.5%	0.674/96.4%	2
Oracle specialist pair (6,912 per specialist)	1.171/100.0%	0.699/100.0%	2

Exposure-matched specialists receive the joint controller's in-domain views; oracle specialists each receive the full 6,912-view joint budget.

Table 24: Training-data allocation for joint and specialist controllers.

Training regime	Swarm unique views	Silo unique views	Aggregate unique views	Optimizer updates	Deployed LoRA adapters
Joint Waggle	3,456	3,456	6,912	Matched	1
Exposure-matched specialist pair	3,456	3,456	6,912	Matched†	2
Oracle specialist pair (6,912 per specialist)	6,912	6,912	13,824	Matched	2

Note. † Each exposure-matched corpus is sampled with replacement to match the joint controller's optimizer updates; no additional unique examples are introduced. Every SILO-BENCH in-domain corpus uses task-stratified sampling.

Table 25: Joint-minus-specialist effects at $N = 64$; the paired 95% interval applies only to the SILO-BENCH contrast.

Comparator	Δ SWARMBENCH P/F normalized score	Δ SILO-BENCH SR	SILO-BENCH paired 95% CI	Reading
Exposure-matched specialist pair	+0.015	+0.008	—	Joint is higher on both
Oracle specialist pair	−0.038	−0.017	[−0.029, −0.005]	96.8% / 97.6% retained

Table 26: Late-conflict reorganization for joint and specialist controllers (40 matched settings).

Deployment	Recovery ↑	Re-synthesis latency ↓	Relative stale AUC ↓	Post-conflict messages/agent
Joint Waggle	36/40	4.5	1.00×	8.2
Exposure-matched specialist pair	35/40	4.8	1.05×	8.3
Oracle specialist pair	37/40	4.3	0.96×	8.1

Note. Relative stale-field AUC is normalized by Joint Waggle.

M SAME-RUNTIME DECISION-LAW ABLATION

M.1 DECISION-SOURCE COMPARISON UNDER A SHARED RUNTIME

For the controller ablation, the complete fixed runtime is

$$\mathcal{R} = \{\textsc{View}, \textsc{Validator}, \textsc{Readiness}, \textsc{Conflict}, \textsc{Transport}, \textsc{Expiry}, \textsc{Compiler}\}.$$

Every decision source receives the same anonymous local view and native private proposal, emits at most one candidate semantic record per update, and is subject to the same admissibility checks, duplicate suppression, expiry, and native compilation. Learned variants use deterministic decoding with at most one validator-code-conditioned regeneration. The runtime may reject or realize a record, but never originates a claim, commitment, challenge, or communication intent.

The fixed heuristic has access to every runtime-supported operation, but uses no learned context-dependent rule: it challenges and revokes an incompatible commitment after verified conflict; otherwise it commits to a ready highest-supported candidate, relays the admissible trace with highest support, deposits fresh private evidence, or takes the native private proposal in that order. Zero-shot Qwen3-4B receives the identical local prompt and decision schema without distillation. Private-only suppresses every social record; the random control samples only from locally admissible records. GPT-4o sees the same one-view interface and serves solely as a local teacher reference.

Table 27: Decision-source ablation under an identical interface and runtime; only the semantic-record source changes.

Decision source	SILO-BENCH success ↑	SILO-BENCH messages/agent ↓	SWARMBENCH P/F $N = 64$ normalized score ↑	Late-conflict recovery ↑	Re-synthesis latency ↓
GPT-4o local teacher	0.736 [0.70, 0.77]	8.45	1.18 [1.08, 1.28]	38/40 (95.0%)	4.0
Waggle	0.682 [0.65, 0.72]	8.03	1.13 [1.05, 1.22]	36/40 (90.0%)	4.5
Qwen3-4B zero-shot	0.392 [0.33, 0.45]	8.56	0.97 [0.87, 1.06]	23/40 (57.5%)	8.5
Fixed local heuristic	0.276 [0.22, 0.34]	8.82	0.90 [0.80, 1.00]	14/40 (35.0%)	10.5
Private-only	0.112 [0.08, 0.15]	0.00	0.94 [0.84, 1.03]	1/40 (2.5%)	—
Random legal record	0.093 [0.06, 0.12]	6.00	0.68 [0.55, 0.80]	1/40 (2.5%)	—

Note. Brackets are paired-bootstrap 95% confidence intervals; late-conflict metrics use 40 matched settings, and latency is computed on recovered settings.

Table 28: Late-counterevidence outcomes for alternative decision sources under the same runtime; process metrics are computed on recovered settings.

Decision source	Recovery ↑	Re-synthesis latency ↓	Relative stale field AUC ↓	Post-conflict messages/agent
GPT-4o local teacher	38/40	4.0	0.90×	8.5
Waggle	36/40	4.5	1.00×	8.2
Qwen3-4B zero-shot	23/40	8.5	1.70×	8.8
Fixed local heuristic	14/40	10.5	2.30×	8.9
Private-only	1/40	—	—	0.0

Note. Relative stale-field AUC is normalized by Waggle. The random legal-record control is omitted because it reaches the endpoint in only 1/40 settings, leaving the trajectory summaries undefined.

The endpoint differences in Tables 27 and 28 coincide with a distinct post-conflict trajectory: Waggle clears stale fields and re-synthesizes a replacement commitment in roughly half the latency of zero-

shot or heuristic control. This source-level comparison complements the component interventions in Figure 4 and Appendix P.3: the former holds all runtime machinery fixed, whereas the latter removes individual parts of the reorganization loop.

N SILO-BENCH SCALING BY DIFFICULTY

Table 29: Waggle success (%) by population and SILO-BENCH difficulty level.

Population	Level I	Level II	Level III	Overall
$N = 16$	94.5	87.8	76.6	86.3
$N = 32$	91.8	80.7	64.5	79.0
$N = 64$	87.6	69.9	17.1	58.2

Table 1 gives the cross-method comparison at $N = 64$; this table tracks how Waggle's difficulty profile changes with population size.

O HELD-OUT CONTROLLER RESULTS

Table 30: Per-mode classification on 800 held-out local views.

Derived mode	Support	Predicted	F1 (%)
Explore	200	200	100.00
Deposit	100	100	100.00
Relay	100	100	100.00
Challenge	200	200	100.00
Synthesize	100	100	100.00
Abstain	100	100	100.00

All 800 predictions are valid JSON, canonical under the contract, schema-valid, and executable. There are no invalid predictions and no safe-fallback decisions. Full-record exact match is 598/800 (74.75%).

P MECHANISM MEASUREMENTS

P.1 RUNTIME-COMPONENT INTERVENTION

The formal trace stores both controller intent and executed transport. We report the following quantities without assigning persistent roles:

- **Recruitment events**: executed Deposit or Relay writes carrying novelty/support, plus spatial actions guided by a model-selected relay.
- **Readiness-to-commit events**: transitions from no commitment to a controller-selected commitment after candidate readiness.
- **Commitment reopenings**: Challenge decisions that revoke an incompatible local commitment after verified counterevidence.
- **Decay**: reduction from initial to final trace mass and the number of expiry advances.
- **Explore actions**: validated decisions assigned the Explore mode in Table 12, which execute a benchmark-native private action without a social write or commitment.

On SWARMBENCH, the spatial transport layer additionally records candidate, executed, and suppressed writes by mode; duplicate-fingerprint, missing local trace, exhausted-lifetime, and zero-mass suppressions are separated. These counters distinguish a learned communication decision from a transport invariant and verify that communication savings do not come from invalid generations.

Table 31: Runtime-component interventions over 40 matched late-conflict settings. The complete runtime recovers 36/40; ΔRec. is Waggle minus each ablation.

Variant	Recovery	ΔRec. (pp)	Pooled Waggle signature
NoConflict	0/40	+90.0	No explicit recovery in any group
NoDecay	33/40	+7.5	3.89 rounds faster; 86.0% less stale AUC
Irrev. Quorum	0/40	+90.0	No explicit recovery in any group

P.2 Representative late-conflict trajectory

Figure 10 makes the population-level $A \to \varnothing \to B$ loop visible. Before the intervention, A dominates. At relative round zero, a controlled target relocation injects counterevidence through native egocentric views and the committed share falls, exposing an opening phase. Challenge peaks at $+1$; B first exceeds A and Synthesize reactivates at $+2$. Field mass peaks at $+3$ and combined Challenge/Synthesize activity at $+4$–$+5$, after which both actions subside while B remains dominant and total commitment rebuilds.

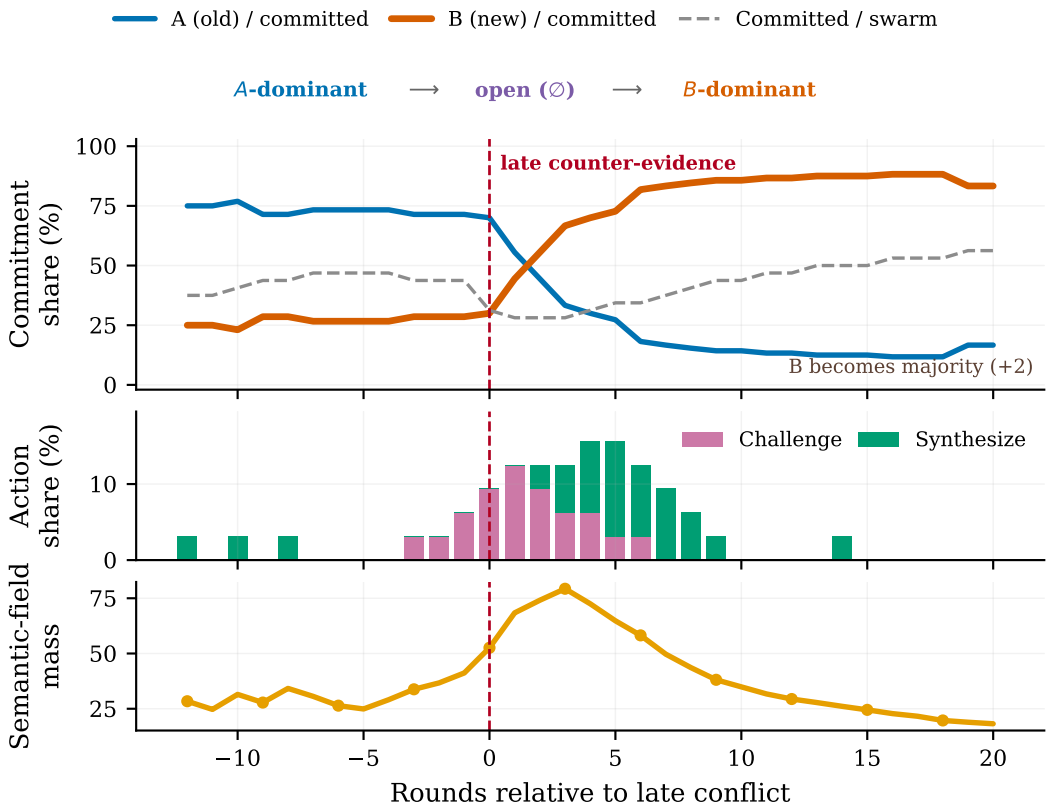


Figure 10: **Representative late-conflict reorganization in SwarmBench Pursuit** ($N = 32$)**.** A target relocation at round 0 enters only through native local views; panels show commitment shares, Challenge/Synthesize activity, and semantic-field mass.

P.3 RUNTIME-COMPONENT TASK EXTENSION

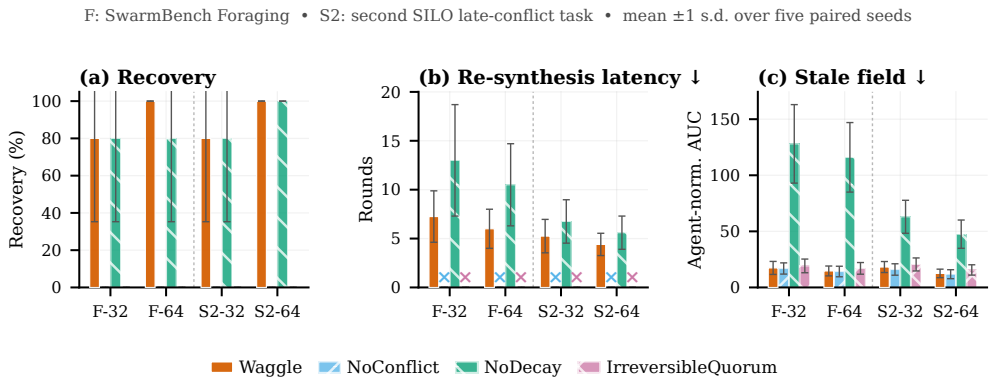


Figure 11: **Runtime-component extension.** SWARMBENCH Foraging (F) and a second SILO-BENCH late-conflict task (S2) are evaluated at $N \in \{32, 64\}$ over 20 matched settings; latency and stale-field AUC show mean ± 1 s.d.

P.4 RUNTIME-PARAMETER SENSITIVITY UNDER A FROZEN CONTROLLER

We characterize the spatial runtime around its default operating point while keeping the Waggle checkpoint, decoder, validator, transport, compiler, benchmark instances, and evaluation seeds fixed. All primary results use the default configuration; the sensitivity analysis varies one factor at a time. We vary the support threshold $\tau_s \in \{6, \mathbf{8}, 10\}$, the coherence threshold $\tau_c \in \{0.5, \mathbf{0.6}, 0.7\}$, and the Pursuit refresh interval $R_{\text{ref}} \in \{8, \mathbf{12}, 16\}$, holding the two-channel and zero-conflict requirements fixed. The sweep is specific to the numerical SWARMBENCH readiness gates; SILO-BENCH readiness follows native constructibility.

Each of the seven configurations is evaluated on Pursuit and Foraging at $N \in \{32, 64\}$ with five paired held-out seeds, and on the Pursuit late-counterevidence protocol at both populations with the same five seeds. This gives $7 \times (20 + 10) = 210$ frozen-controller runs. Conventional task quality and receiver-delivered messages are normalized within each task–population cell by the default configuration before macro-averaging.

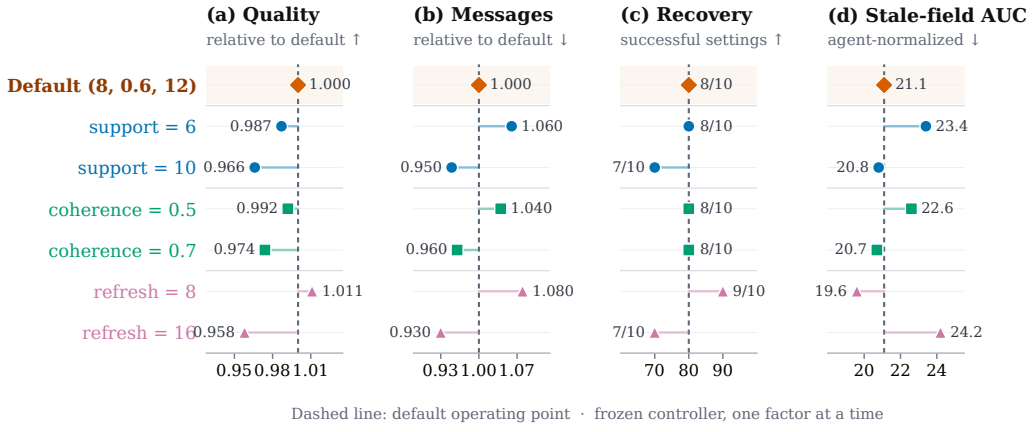


Figure 12: **Runtime-parameter sensitivity with a frozen controller.** Seven one-factor-at-a-time configurations vary the support threshold, coherence threshold, or refresh interval. Quality and messages are normalized to the default within each task–population cell; dashed lines mark the default. Recovery and stale-field AUC use ten matched Pursuit settings.

Across the sweep, normalized task quality remains between 0.958 and 1.011, and recovery counts range from 7/10 to 9/10. Readiness-gate perturbations change traffic by at most 6%. The faster-refresh setting records 9/10 recovery and lower stale-field AUC at higher traffic; the slower-refresh

setting records 7/10 recovery and higher stale-field AUC at lower traffic. Nearby settings preserve performance while shifting the communication–responsiveness trade-off.

P.5 Delays, message loss, and temporary churn

We reuse the late-counterevidence matrix of SwarmBench Pursuit and Silo-Bench III-26 at $N \in \{32, 64\}$, with ten matched held-out cases per task–population cell (40 settings). The controller and task instances remain fixed across regimes. For each delayed message, the additional delay is sampled from $\mathrm{DiscreteUniform}\{0, 1\}$ or $\mathrm{DiscreteUniform}\{0, 1, 2\}$. One delay unit is the interval between consecutive nominal local updates in the synchronous reference execution; zero means delivery at the normally available next update. Loss regimes drop messages at the specified rate. For temporary churn, 10% of active nodes are selected uniformly at counterevidence injection and suspended for two nominal update intervals. Their task actions, controller calls, and outbound communication pause; incoming messages are dropped, not buffered. Nodes then rejoin with their private state and local commitment preserved.

Recovery follows Appendix G.1. Re-synthesis latency counts completed agent-local updates from counterevidence injection to the first valid synthesis of a replacement commitment, divided by the reference population N. One round-equivalent thus corresponds to N completed local updates; this measures update count, not API wall-clock latency. As in the runtime-component analysis, latency is summarized over recovered settings. Table 32 reports task quality and stale-field AUC relative to the synchronous reference, and distinguishes attempted from receiver-delivered messages per agent.

Table 32: Execution perturbations over 40 matched late-counterevidence settings. Quality and stale AUC are relative to the synchronous reference; latency is in round-equivalents. Message columns report attempted/delivered counts per agent.

Execution regime	Quality / sync ↑	Recovery ↑	Re-synthesis latency (round-equiv.) ↓	Relative stale AUC ↓	Msg./agent attempted/delivered
Synchronous	1.000	36/40 (90.0%)	4.5	1.00×	8.2 / 8.2
Delay $\{0, 1\}$	0.986	35/40 (87.5%)	5.0	1.07×	8.3 / 8.3
Delay $\{0, 1, 2\}$	0.969	34/40 (85.0%)	5.5	1.15×	8.4 / 8.4
10% message drop	0.951	33/40 (82.5%)	5.8	1.20×	8.4 / 7.6
20% message drop	0.905	30/40 (75.0%)	6.7	1.34×	8.5 / 6.8
10% temporary churn	0.925	32/40 (80.0%)	6.1	1.27×	8.6 / 8.1
Delay $\{0, 1, 2\}$ + 10% drop	0.895	31/40 (77.5%)	6.5	1.33×	8.6 / 7.8

Delays and moderate loss preserve most synchronous task quality, while recovery slows and stale-field burden rises. Stronger loss and combined perturbations degrade performance further. These tests characterize finite delivery delays and temporary unavailability; they do not establish robustness to arbitrary schedules, permanent information loss, or adversarial agents.

Q Limitations and Deployment Boundaries

Interface and substrate scope. Waggle assumes a deterministic adapter that exposes bounded private task state, locally checkable incident evidence, and legal task actions through the shared anonymous interface (Section 3; Appendix D). We evaluate this contract on spatial and distributed-language substrates, through frozen transfer to graph-structured tasks with a new action vocabulary, and under population extrapolation beyond the primary matrix (Tables 1 and 2; Appendix C.2). These results support reuse across the tested local-interaction settings; environments in which the observation interface or transport mechanism must itself be learned remain outside the present scope.

Anonymity and identity-dependent coordination. The present interface excludes persistent identity-indexed reputation and assigned global roles, so it does not maintain reliability histories for named agents across episodes. Task-relevant capability differences could instead enter through bounded local descriptors while arbitrary identities remain hidden (Appendix R).

Learned semantics and fixed runtime. Waggle learns state-dependent task, trace, and commitment decisions, while the runtime supplies validation, bounded transport, expiry, and native realization (Section 3). This leaves low-level coordination affordances in the deployment contract rather than

learning them end to end. Identical-runtime decision-source comparisons, controller-fixed component interventions, and nearby spatial-runtime sweeps isolate the two sides of this boundary (Section 5.6; Appendices M.1, P.1, and P.4). Across the instrumented evaluation matrix, 99.75% of first decodes are executable as emitted and the no-communication fallback is never invoked. Remaining cases require envelope normalization, at most one validator-code-conditioned controller regeneration, or conservative projection; wrapper operations cannot originate social content (Appendix C.3). Thus the runtime constrains and realizes controller decisions rather than supplying their social semantics.

Conditional symmetry guarantee. Proposition 1 states the synchronous case; Appendix E.2 extends the structural guarantee to paired event-driven executions, with a distributional result under an equivariant marked-event kernel. Both require equivariant view construction, local decisions, and runtime operations, including completion-time admission in the asynchronous contract. SCD promotes the local premise but does not enforce it architecturally. We measure the learned approximation through held-out summary-, decoded-record-, and admitted-record agreement and the symmetry-breaking diagnostic (Table 3; Figure 7). Our primary benchmark executions and next-field training targets use synchronous updates. Appendix P.5 measures degradation under finite delivery delays, message loss, and temporary churn. More general scheduling and membership changes can reduce evidence freshness, alter temporal reachability, or remove task-relevant information, affecting both progress and feasibility beyond these tested regimes. Structural relabeling consistency is distinct from equal task performance across execution regimes, schedule independence, or task convergence.

Supervision and transfer. The law is trained from teacher-labeled single local views and rollout transitions on the training substrates, so its coverage is bounded by those local situations (Section 4; Appendix F). Training collection and formal evaluation use disjoint cases and seeds; the factorial arms share teacher records and optimization budgets, and the same structural and behavioral pattern replicates with a second student backbone (Appendices G.1 and G.2). In AGENTSNET-RP, the controller is frozen before any benchmark instance is accessed, and no AgentsNet teacher record, rollout transition, or evaluation answer enters training or model selection (Appendix I). Continual adaptation in open-ended domains is not evaluated here.

Open-world deployment. The evaluation assumes faithful benchmark observations and non-adversarial agents. Local admission checks record legality and consistency with available evidence, not the truthfulness of arbitrary external observations. In particular, the spatial readiness gate counts incident channels rather than independent evidence origins (Appendix H). Duplicate suppression and finite trace lifetime limit repetition and persistence; they do not authenticate sources or establish Byzantine fault tolerance. In open tool environments, untrusted observations could imitate a verified-conflict marker or evade canonicalization. Deployments with consequential external actions should authenticate evidence provenance, sandbox tools, and retain appropriate human approval. The experiments contain no human-subject data.

R FUTURE DIRECTIONS

Adaptive interfaces and broader substrates. The local interface in Waggle separates substrate mechanics from the law that generates organization (Section 3; Appendix D). Cross-substrate evaluation and frozen AGENTSNET-RP transfer provide a starting point for extending this separation to multimodal observations, tool-using agents, embodied interaction, partially specified action spaces, and environments whose local interface changes over time (Tables 1 and 2). Parts of the adapter could also be learned or continually refined while retaining provenance, locality, boundedness, and legal actuation. The central design question is how much interface plasticity can be introduced without relocating organizational semantics from the shared law into the substrate layer.

Asynchronous and changing populations. Appendix E.2 establishes conditional relabeling consistency for event-driven execution, and Appendix P.5 tests bounded delays, message loss, and temporary churn. Further evaluation could examine irregular clocks, longer delays, ongoing membership turnover, and asynchronous model-call completion. It should distinguish matched call budgets from matched time deadlines and account for whether departing nodes remove information needed to solve the task.

Capability-conditioned local laws. Parameter sharing constrains the coordination rule rather than requiring identical task-relevant capabilities. A heterogeneous extension could expose bounded local

descriptors of available tools, action constraints, and context or compute budgets while retaining one shared controller and the joint decision schema. The adapter would validate actions against the focal node's capabilities. Budget-dependent truncation and summarization would remain local and satisfy the same equivariance requirements, including insensitivity to arbitrary handle ordering. Capability attributes would travel with their nodes under relabeling, while information about other nodes would arrive through incident communication. Under equivariant view construction, the complete deployed map, and runtime operations, the local-to-global argument (Proposition 1; Appendix E.2) applies to this augmented state. Anonymity removes dependence on arbitrary names while preserving genuine capability differences as a source of task-relevant asymmetry, without assigning organizational roles. This extension retains a shared coordination policy; mixtures of independently parameterized controllers require separate analysis. Generalization of the current checkpoint to unseen capability mixtures remains to be evaluated.

More autonomous and continual learning. SCD currently combines teacher-labeled local decisions with rollout-grounded transition supervision (Section 4; Appendix F). Disjoint evaluation and frozen target-substrate transfer establish a clean basis for reducing external supervision further (Appendices F.1 and I). Candidate signals include self-supervised local prediction, preference or outcome feedback, self-play, offline or online reinforcement learning, and locally generated counterfactual trajectories. Continual updates introduce a sharper problem: new local experience should improve the law without erasing anonymous-view consistency or destabilizing previously useful coordination reactions. Local replay, constrained policy updates, and decentralized parameter adaptation offer possible routes to combining plasticity with the invariances promoted by SCD.

Learned transport and multi-timescale coordination state. The current runtime fixes validity, transport, expiry, and native realization so that semantic decisions can be attributed to the policy. Controller-fixed component interventions and runtime sensitivity sweeps provide an empirical base for constrained co-design of these mechanics (Section 5.6; Appendices P.1 and P.4). Future systems could learn when to allocate communication bandwidth, how long locally useful information should persist, which traces to compress or forget, and how readiness thresholds should respond to local conditions. Longer tasks may also benefit from several timescales of bounded coordination state. Such adaptation should preserve hard local constraints on provenance, bounded communication, legal actuation, and reversibility while learning the operating policy around them.

Theory beyond exact relabeling equivariance. Propositions 1 and 2 provide a structural basis for a quantitative theory of learned local laws. Approximate extensions could bound finite-horizon trajectory deviation under uniform local error controls over reachable histories. Beyond symmetry, open questions include conditions for convergence or recurrent organization under asynchronous updates, communication complexity, recovery time after counterevidence, bounds on stale coordination mass, and the relation between local memory horizon and population-scale stability. Progress and convergence results would additionally need conditions on event scheduling, information reachability, and the local reaction dynamics. These questions connect learned LLM swarms to population protocols, distributed algorithms, and stochastic interacting systems.

Robustness, trust, and partially aligned participants. The bounded interface makes trust assumptions explicit and creates a natural place to attach stronger safeguards. Open systems may contain corrupted observations, misleading messages, prompt injection, compromised tools, or participants whose objectives are only partially aligned. Research directions include authenticated provenance, robust local evidence aggregation, privacy-preserving coordination, and trust mechanisms that do not depend on stable global identities. Formal fault models could measure graceful degradation under Byzantine, unavailable, or colluding nodes, while mixed human–agent and mixed-incentive populations would test whether local reversibility remains effective when disagreement is strategic rather than incidental. These extensions build on the deployment boundary stated in Appendix Q.

Longer-horizon evaluation and measurement of emergent organization. The current evaluation separates population and interaction scaling, frozen transfer, anonymous stability, and reorganization after counterevidence (Sections 5.2–5.6). Future benchmarks could combine these stresses through long-lived objectives, multiple competing commitments, changing goals, node churn, heterogeneous capabilities, and repeated environmental shocks. Richer diagnostics could measure formation time, turnover of coordination state, complete recovery trajectories, communication concentration, and the diversity and persistence of emergent organizational patterns. Causal interventions on local evidence

or topology could further distinguish communication patterns that merely correlate with success from organization that mediates collective behavior. Broader backbone families, larger deployments, and system-level measurements of latency, memory, energy, and model-call cost would complement the algorithmic evidence reported here.

Across these directions, the broader question is how far adaptive organization can emerge from repeated execution of a reusable local law, and when, if ever, explicit system-level structure provides additional value.