

Controlling Collectives of AI Agents in Reasoning Space with Spatial Transformers

Frederic Vatnsdal, Roshan Gopal, Romina Garcia Camargo, Vijay Kumar, and Alejandro Ribeiro

Abstract—Large Language Models (LLMs) introduce an exciting new paradigm for planning and navigation in robotics, but fail on even simple multi-robot tasks as team sizes grow. We propose COMPASS, a scalable, decentralized multi-robot architecture for controlling large collectives of agentic robots with reasoning space feedback control. Feedback is generated locally on each robot by a spatial transformer which aggregates multi-hop messages across the fleet into a learned feedback token. Our experiments find that collectives of language models demonstrate performance gains from structured diversity of the input command, which can cancel biases; an advantage that is held across scale. Compared against a centralized frontier LLM policy and a language-only communication ablation, we find that the coupled design of COMPASS decisively produces cohesive flocking formations that accurately fly the commanded intent. We show that reasoning feedback works best when composed with a compact learned token. Our ablations show that hand engineered feedback with raw state appearing in the language channel obliterates cohesion. COMPASS generalizes zero-shot to unseen instructions of ambiguous meaning while commanding flocks up to $16\times$ its training scale, flying up to 1024 robots under natural language commands.

I. INTRODUCTION

Embedding language models (LMs) into robotic planning and control frameworks harnesses general-purpose reasoning for embodied tasks, granting unprecedented zero-shot capabilities [1]–[5]. The collaborative feedback control of large-scale collectives of embodied agents could enable robust reasoning over challenging tasks, enabling sophisticated multi-robot reasoning. We present COMmanding Multi-robot Policies with Attention and Semantic Steering (COMPASS); [Figure 1](#). COMPASS is a decentralized, feedback control architecture operating in reasoning space that integrates spatial transformers in the loop to infuse collective fleet embeddings into the feedback tokens injected by each robot’s LM. COMPASS out-scales and outperforms approaches that rely only on the language-channel for coordination.

The methods for controlling embodied language models at scale remain underdeveloped in the literature [6]. Providing a language model with overwhelming knowledge about teammates can easily corrupt reasoning over the original task, hinting that new methods must be developed to leverage

F. Vatnsdal, R. Garcia Camargo, and A. Ribeiro are with the Department of Electrical and Systems Engineering, University of Pennsylvania, USA. R. Gopal is with the School of Arts and Sciences, University of Pennsylvania, USA. V. Kumar is with the GRASP Laboratory, University of Pennsylvania, USA. Emails: {vatnsdal, rominag, kumar, aribeiro}@seas.upenn.edu, rgopal@sas.upenn.edu. This work was supported by grant ARL DCIST CRA W911NF-17-2-0181.

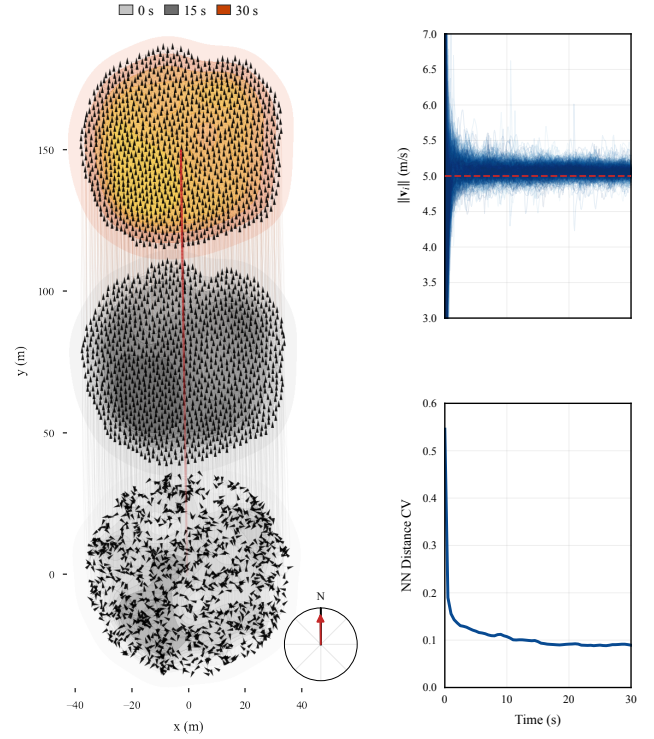


Fig. 1. **Left:** “Fly toward twelve o’clock” – the $N = 1024$ decentralized swarm individually decodes the language instruction and mobilizes north-bound at 5 m s^{-1} , shown at 0, 15, and 30 s (agents black, communication graph at $r_c = 4 \text{ m}$). The center of mass is drawn as a red throughline. **Top Right:** the speed of every agent converges to the commanded 5 m s^{-1} over the rollout. **Bottom Right:** the coefficient of variation of the robots’ nearest-neighbor distances converges as the flock settles into an even lattice.

reasoning at scale [7], [8]. Our empirical study of frontier models reveals that language models struggle with spatial awareness and reasoning over the states of teammates while simultaneously completing a navigation task; a limitation that COMPASS addresses directly.

Language commanded flocking is our chosen test-bed for examining how collaborative reasoning feedback loops to both steer a fleet toward a common objective while maintaining cohesive formations. COMPASS builds on the perception–action–communication literatures, with learned communication backbones Graph Neural Networks (GNNs) [9], [10] and spatial transformers [11] that provide the scalability that language models lack on their own. We make the following contributions:

- **Scale:** zero-shot generalization to $16\times$ the training team size on kilo-scale flocks.
- **Structured Diversity:** COMPASS has the unique ability to spread multiple phrasings of the same task across the fleet, so that a minority running an easier phrasing can raise the accuracy of the collective.
- **Generalization:** COMPASS taps into the latent dimension of language models with a learned query-only interface that is able to ground unseen directions to executable intents.

A. Related Work

Large Language Models (LLM) for Mission Planning

Integrated into control frameworks, LLMs enable zero-shot execution across robotic navigation and manipulation tasks [1]–[3]. Kwon et al. [1] demonstrate that an LLM can generate low-level end-effector trajectories using only off-the-shelf vision modules. SayCan [2] demonstrated that LLMs are effective high level mission planners due to their rich semantic reasoning. Nonetheless, the models require grounding to work in robotics tasks that have more stringent physical constraints and metrics of success, often achieved through vision or human interaction [12], [13]. Approaches for online planning with LLMs incorporate feedback from the environment as well as previously generated plans to influence subsequent generations [14]–[16]. PRISM [16] distilled online semantic planning into SLMs that run on robot hardware. Building on this approach, Ro-SLM [17] optimizes on-device SLMs by employing the LLM as a reward function to guide fine-tuning on synthesized task-planning data.

Coordination of LLMs for Multi-Robot Tasks Recent works study the capabilities of LLMs for coordinating multi-robot teams with centralized [6], [18], [19] or decentralized approaches [20]–[23]. Decentralized consensus mechanisms generate robust team plans that avoid instability and inconsistent behaviors. LLM-Flock [20] addresses formation control of robots, each endowed with an independent LLM generating plans locally. RoCo [21] uses a dialectic style of communication between LLM agents that govern individual robots, with focus on multi-robot manipulation that supports human interaction. ZeroCAP [22] demonstrated that a VLM-LLM combined architecture enables reasoning about spatial tasks with multi-robot teams such as infilling and encircling objects.

II. REASONING-SPACE FEEDBACK CONTROL OF COLLECTIVE AGENTS

COMPASS pilots a team of N networked robots with physical state at control step τ $\mathbf{X}(\tau) = [\mathbf{x}_1(\tau), \dots, \mathbf{x}_N(\tau)]^\top \in \mathbb{R}^{N \times d_x}$ and actions $\mathbf{U}(\tau) = [\mathbf{u}_1(\tau), \dots, \mathbf{u}_N(\tau)]^\top \in \mathbb{R}^{N \times d_u}$. Alongside this fast physical system, each robot i carries an LLM and a persistent *program* of embedded tokens,

$$\text{PROG}_i(t) = [\mathbf{P}_i \parallel \mathbf{T}_i \parallel \mathbf{f}_i(t) \parallel \mathbf{g}], \quad (1)$$

that is comprised of the operator prompt $\mathbf{P}$, a thinking trace $\mathbf{T}$, a feedback token $\mathbf{f}$, and a learned goal token $\mathbf{g}$, all in the model’s embedding space $\mathbb{R}^{d_{\text{LM}}}$. COMPASS is therefore formed by the coupling of a physical dynamical system with latent reasoning dynamics, optionally evolving on separate timescales; Figure 2. The fleet’s state $\mathbf{X}(\tau)$ is advanced by a spatial transformer (Section II-B), while a latent *reasoning state* $\mathbf{z}_i(t)$ updates at the slower reasoning rate. Reasoning is re-derived from the program each time the feedback token is refreshed with a projection of the fleet state (Section II-A). The prompt $\mathbf{P}$, e.g., “vector two o’clock”, is updated when the operator redirects the fleet, triggering a regeneration of $\mathbf{T}$.

A. Reasoning in the Loop

COMPASS reasons through both *thinking* in language space at generation time and *latent* reasoning at control time. The generic update of the reasoning state is

$$\mathbf{z}_i(t+1) = \phi_{1:\ell}(\mathbf{P}_i \parallel \mathbf{T}_i \parallel \psi(\pi_i(t)) \parallel \mathbf{g}), \quad (2)$$

where $\phi_{1:\ell}$ denotes the first ℓ layers of the LM, ψ is a learned adapter that embeds the output of the downstream controller $\pi_i(t)$ (Section II-B) as the feedback token $\mathbf{f}_i(t)$. Equation (2) defines a family of collective reasoning space controllers.

1) *Thinking*: Each robot generates its own trace conditioned on the prompt, one token at a time using all layers and the language head,

$$\mathbf{T}_i(\tau+1) = \mathbf{T}_i(\tau) \parallel \phi_{\text{gen}}(\mathbf{P}_i \parallel \mathbf{T}_i(\tau)), \quad (3)$$

until a fixed thinking length is reached. Traces are sampled by each individual robot and N robots produce N distinct readings of the same instruction; diversity that the fleet must reconcile. Once complete, a trace is frozen until the next prompt change and KV-cached. Every subsequent latent update in (2) reuses it, saving computation cost.

2) *Latent*: The reasoning state $\mathbf{z}_i(t)$ is the hidden state at the goal token’s position after a forward pass of only ℓ of the LM’s L layers over the program. Because of causal masking and their terminal position, the feedback and goal tokens query the entire program while no language token attends to them. This simplifies learning as the loop tokens $\mathbf{f}$ and $\mathbf{g}$ need not lie on the language manifold. The extraction depth ℓ is a design parameter; empirically, vague commands require deeper extraction but performance does not always improve monotonically. Stacking the latent states of the fleet gives $\mathbf{Z}(t) = [\mathbf{z}_1(t), \dots, \mathbf{z}_N(t)]^\top$, the collective reasoning state of the system.

B. Spatial Transformer

A multi-agent spatial transformer (MAST) [11] turns the fleet’s reasoning states into actions, diffusing them hop-by-hop over the communication graph using neighbor communication only. The input to MAST is constructed by concatenating the reasoning state with the observed state, $\mathbf{h}_i^{(0)} = \text{MLP}([\mathbf{z}_i \parallel \mathbf{x}_i])$, where $\mathbf{h}_i^{(0)} \in \mathbb{R}^{d_{\text{MAST}}}$. Absolute position never enters MAST; geometry enters attention only

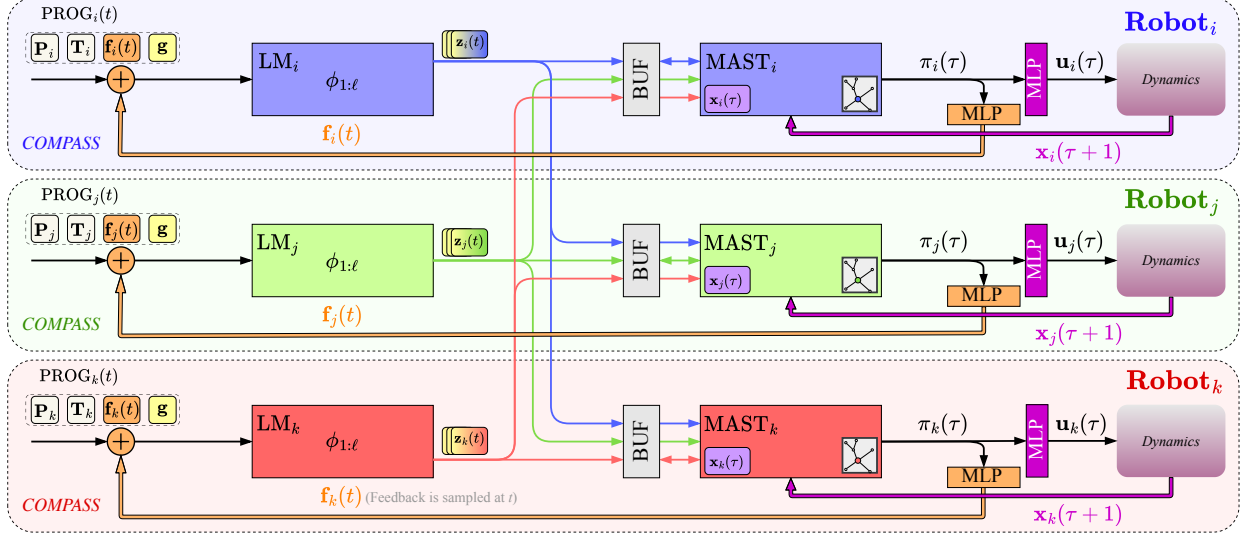


Fig. 2. COMPASS uses local LMs to update the reasoning state $\mathbf{z}(\mathbf{t})$ from both the command intent and the learned feedback token (orange) produced by MAST from neighborhood information. COMPASS is deployed identically on each robot despite being trained centralized with spatial masking. Neighboring robots exchange reasoning over the communication network and store received communications in buffers. The same MAST output used to produce feedback is also projected into the action space for execution of low-level control (purple). MAST incorporates low-level state feedback into its input features.

through relative encodings, making the policy translation invariant without breaking permutation equivariance.

2D rotary positional encoding [24] injects geometry into the attention logits by splitting the query and key into 2ζ channel pairs indexed by axis $a \in \{x, y\}$. Choosing the encoding period λ , we sweep frequencies $\omega_f = 2\pi f/\lambda$ for $f = 1, \dots, \zeta$ with $\zeta = d_{\text{MAST}}/4$, and each channel pair is rotated by the matching coordinate of the robot's position $\mathbf{p}_i$,

$$\tilde{\mathbf{q}}_i^{(a,f)} = R(\omega_f p_i^a) \mathbf{q}_i^{(a,f)}, \quad \tilde{\mathbf{k}}_j^{(a,f)} = R(\omega_f p_j^a) \mathbf{k}_j^{(a,f)}, \quad (4)$$

generating rotated queries and keys, $\tilde{\mathbf{q}}_i, \tilde{\mathbf{k}}_j \in \mathbb{R}^2$ (dropping the (a, f) notation for clarity). Then, $R(\theta)$ is the 2×2 rotation matrix, and $\mathbf{q}_i, \mathbf{k}_i \in \mathbb{R}^{d_{\text{MAST}}}$ are the original agent query and key, respectively. Since $R(\alpha)^\top R(\beta) = R(\beta - \alpha)$, the resulting logit

$$\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j = \sum_{a,f} \mathbf{q}_i^{(a,f)\top} R(\omega_f(p_j^a - p_i^a)) \mathbf{k}_j^{(a,f)} \quad (5)$$

depends on positions only through the displacement $\mathbf{p}_j - \mathbf{p}_i$ [24]. The encoding period λ upper-bounds the resolvable geometry: too short and distant neighbors alias, too long and the encoding is inert. Rotary encoding is applied at the first MAST layer only.

Attention is restricted to the communication graph. With $d_{ij} = \|\mathbf{p}_j - \mathbf{p}_i\|_2$ and mask $\mathcal{M}_{ij} = 1$ iff $d_{ij} \leq r_c$, per head of dimension d_h ,

$$A_{ij} = \text{sm}_{j: \mathcal{M}_{ij}=1} \left(\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j / \sqrt{d_h} \right), \quad (6)$$

so a robot attends only to teammates within communication range. Messages are additionally conditioned on edge geometry through the normalized displacement $\mathbf{e}_{ij} = \lfloor \mathbf{p}_j - \mathbf{p}_i \rfloor / r_c$, for $\mathbf{h}_i^{(\ell)} = \mathbf{h}_i^{(\ell-1)} + \mathbf{W}_o \sum_j A_{ij} (\mathbf{W}_v \mathbf{h}_j^{(\ell-1)} + \mathbf{W}_e \mathbf{e}_{ij})$, followed by a pre-norm residual MLP. Here, the learned parameters per layer and per head are, $\mathbf{W}_o, \mathbf{W}_v \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_e \in \mathbb{R}^{d_h \times 3}$. Each masked attention layer is exactly one round of message exchange with 1-hop neighbors, so L_{MAST} layers give every robot an L_{MAST} -hop receptive field. Therefore, due to overlapping neighborhoods, information is able to diffuse across the swarm at each step. The communication graph's regularity is learned by MAST allowing the same parameters at $N = 64$ to generalize to $N = 1024$. MAST is trained centralized but can be executed decentralized as each robot only needs to evaluate its own attention row from tokens received over the network.

The final token $\boldsymbol{\pi}_i = \mathbf{h}_i^{(L_{\text{MAST}})}$ is decoded twice to obtain feedback and action. An MLP head produces the action $\mathbf{u}_i = \text{MLP}(\boldsymbol{\pi}_i)$ with neighbor actions predicted as byproducts but only its own is executed. A second MLP head acts as an adapter to the LM's latent space and learns $\mathbf{f}_i = \text{MLP}(\boldsymbol{\pi}_i)$, closing the reasoning loop of (2).

III. TRAINING FOR CONSENSUS

Flocking under language specified directions – “head East”, “steer toward dusk” – is our demonstrable problem, selected for its key properties: (i) the velocities of each robot are scored against the given semantic instruction, enabling reasoning quality to be measured as heading error in degrees, (ii) the policy must balance the semantic objective with safety by avoiding collisions learned from the expert rather

TABLE I: Training and Evaluation Corpus

Class	Example	Prompts	Surfaces	Templates	Dirs.	Tr/Va/Te [%]
Plain	“steer North”	1920	64	30	8	45/10/45
Abbreviation	“go N”	1920	64	30	8	45/10/45
Opposite	“the opposite of South”	1920	64	30	8	45/10/45
Map	“toward the top of the map”	1920	64	30	8	45/10/45
Clock	“toward twelve o’clock with twelve at the top”	1920	64	30	8	45/10/45
Numeric	“travel 0.0”	5760	720	8	720	50/10/40
Hold	“stop.”	21	21	–	–	60/10/30
Vague/Conceptual	“all units where Santa Claus lives”	384	48	8	8	0/0/100
Secondary interc.	“navigate North-northeast”	192	24	8	8	0/0/100

than scripted, (iii) global heading coherence emerges from neighbor interactions only over long horizons. COMPASS is an incarnation of reasoning-as-policy trained for flocking where the task enters only through the expert that supervises training. Training fits the learned parameters of the loop in (2): the goal and feedback tokens as well as MAST; the language model is frozen. The training data for behavior cloning is comprised of pre-generated thinking traces and centralized expert rollouts. The seven training classes of operator prompt shown in Table I (above the rule) generate traces with the SLM Qwen3-1.7B [25], sampled at temperature = 0.6, top-p = 0.95 and top-K = 20 under a fixed multiple-choice wrapper. The expert rollouts contain state-action pairs sampled at $dt = 0.1$ s for teams of size $N = [32, 64]$, initialized uniformly on a disk of radius proportional to $\sqrt{N}$ at a target density of 0.25 robots per m^2 with random velocities and flown for $T = 10$ s.

Our chosen oracle is a centralized potential-field controller comprised of four terms summed per robot: (i) alignment with the fleet’s mean velocity, (ii) tracking of the target velocity, (iii) cohesion toward the group centroid, (iv) collision avoidance through a repulsion term. We train the decentralized policy to imitate this behavior from neighbor information alone.

To construct the language-action dataset, each prompt is human decoded to obtain the ground-truth intent. Each sample in the language component is constructed through the combination of prompt *surface*, i.e., the style and direction, and *template*, e.g., “go {surface}”. This composite is then ingested by Qwen3-1.7B to generate 64 thinking traces to give each robot a unique trace during training; Table I. The ground truth direction associated with the prompt informs the target velocity direction for the expert; for directional commands, the target speed is 5 m s^{-1} . The directions in the training set map to the 8-point rose and STOP; the exception being the angular prompt class, which sees 720 numerically given bearings (0.5° increments). We follow the aeronautical convention, North at 0° moving clockwise to East at 90° ; this matches the native assumption of the language model, measured through probing. Therefore, this mapping is never included in the prompt. COMPASS is trained with behavior cloning against the centralized expert, augmented by DAgger relabelling and truncated backpropagation through time for the recurrent feedback. We remark that we leave the weights of the LM frozen, relying only on learned tokens $\mathbf{f}$ and

$\mathbf{g}$. This reduces training requirements while simultaneously leaving the optimization of the LM itself as a possible axis to better fit COMPASS to new tasks in the future. MAST uses $L_{\text{MAST}} = 4$ layers of hidden size 256 with 8 heads of size $d_h = 32$, a rotary encoding period of $\lambda = 96$ m, and a communication radius of $r_c = 4$ m. Each robot builds the input token for MAST from its reasoning state $\mathbf{z}_i$, its own velocity $\mathbf{v}_i$, and $\mathbf{s}_i = [\mu_{N_i}, v_e]$, the mean neighbor distance and velocity error against the neighborhood mean.

We use the AdamW optimizer with the learning rate set to 10^{-4} , weight decay 10^{-4} and a cosine annealing learning rate scheduler. COMPASS is trained end-to-end for 200 epochs on an NVIDIA RTX A5000 with 24 GB of video memory. Evaluation is executed on a pair of NVIDIA RTX A5000 to support larger fleets of language models.

IV. RESULTS

We evaluate COMPASS on its ability to fly cohesive flocks up to kilo-scale team sizes (Section IV-A), with structured prompt diversity (Section IV-B), and from new directives (Section IV-C). The operator directives fall on the in-distribution directions: the 8-point compass rose, numerically given bearings and Stop. An additional 8 out-of-distribution directions along the subintercardinals, e.g., North-northeast, test whether $\mathbf{g}$ lies on a roughly circular manifold. Each trial is a rollout of a policy over 60 s and policies are always compared with the same initial conditions and seed. Double integrator robots are initialized uniformly on the disk of radius $\sqrt{4N/\pi}$ with random initial speeds in $\pm 5 \text{ m s}^{-1}$, stepped at $dt = 0.1$ s with zero acceleration.

Policies use Qwen3-1.7B as the default reasoning engine with sampling temperature 0.6. Each robot receives a unique thinking trace, $\mathbf{T}$ of length 160, generated by the same operator prompt $\mathbf{P}$ unless otherwise specified. This emulates the operator voicing a single goal to the team, but each robot reasoning about it individually. We found that the length of the thinking trace can be thought of as another hyperparameter and 160 is an empirically determined sweet-spot (we validated 120 and 320 as well). The feedback token is refreshed using MAST output at a rate of 1 Hz, while the inner MAST control loop runs at 10 Hz. All initial positions and thinking traces are held-out from training and are generated for evaluation.

We compare COMPASS to the oracle expert it was trained on, a frontier model LLM serving as a centralized controller,

and two levels of ablation: (i) COMPASS without communications, i.e., $r_c = 0$. (ii) A frozen SLM with learned register reasoning state readout of comparable size to MAST, which encodes neighboring state in language, e.g., “...My state: pos (x, y), velocity (vx, vy), Neighbor1: relative pos (x, y), velocity (vx, vy), Neighbor 2 ..., Average neighbor vel (vx, vy)”.

The heading error is computed by measuring the heading of the center of mass, θ_{COM} and comparing it to the ground truth heading, $\hat{\theta}$ of the given directive, $e_h = |(\theta_{\text{COM}} - \hat{\theta} + 180) \bmod 360 - 180|$. The fragmentation number describes whether the swarm is connected (frags = 1) or not (frags > 1). Velocity variance is the variance in the agents’s velocities about the fleet mean over the settled window; it measures the consensus of the swarm. The coefficient of variation (CV) of the robots’ nearest-neighbor distances summarizes the spatial uniformity of the system. Overly clumped flocks inflate the CV while an even lattice drives it to zero.

A. Scalability

Although we train COMPASS on $N = [32, 64]$, we evaluate the system and baselines on scales of up to $N = 1024$ to determine how the system performs under increasingly challenging settings. The results in Figure 3 illustrate that COMPASS maintains consistent performance across scale and heading error *decreases* with scale. This is a boon of collective reasoning – the consensus of many diverse reasoning outputs, integrated through the multi-hop communication of MAST, provides the swarm with a more accurate heading. The error goes from 2.79° at $N = 8$ to 0.34° at $N = 1024$; scale brings tangible advantages.

Under language-only communication, neighboring states are front-loaded into the semantic portion of a robot’s prompt and the $N = 64$ flock fragments into 15 groups and slows down, traveling at an average velocity of 2.5 ms^{-1} (half of the target velocity). This degenerate result is worse than ablating communication altogether and appears to stem from requiring the thinking trace to reason over substantial spatial information from neighboring agents. This is a task that even frontier models struggle with when not aided by tooling. Diagnosing the thinking trace reveals that reasoning over the neighbor state overwhelms the model and little compute is spent determining the intended direction of heading.

A centralized LLM-based controller is able to decode the given semantic directive perfectly, flying the core of the flock in the correct direction. As N increases, the LLM fails to provide coherent accelerations to all robots and the flock fragments as more of these robots execute ill-coordinated accelerations. Sweeping across scales (Figure 4), we find that the centralized LLM demonstrates two failure modes: (i) generating accelerations that satisfy both consensus and cohesion and (ii) the generation of the incorrect *number* of actions for the team that becomes worse with increasing N and requires a retry to obtain the correct number of actions; Figure 4. This highlights the scalability failure of a centralized LLM controller, which COMPASS resolves natively. We remark that, given enough compute on each

robot, the frontier model itself could serve as the LM in COMPASS.

B. Structured Diversity

Does a fleet of LMs reason about a vague command better than one? We find that the agentic crowd is unwise when given same prompt or when each robot receives a different vague phrasing for the same directional intent, i.e., robot i receives phrasing $i \bmod 4$; Figure 5. COMPASS is able to leverage an advantage inherent to a collective of LMs: it can provide diverse re-phrasing of the task; crucial when some phrasings are easier to reason over than others. Phrase diversity with an included easier rephrasing of the task is a boon that enables recovery of the intent through feedback via MAST; both in terms of heading error and speed of the flock. Notably, the gaps in capabilities remain consistent across scales, evaluated over 10 trials at each scale.

C. Command Generalization

COMPASS embeds an off-the-shelf LM to serve as a reasoning engine. A key advantage of COMPASS is the interchangeability of the LM, providing deployment flexibility for platform and task constraints. We measure the heading error and latency of three sizes of the Qwen3 LM on our cardinal navigation task, trained with an MLP instead of MAST and evaluated as individual agents ($N = 1$); Table II. The linear head regresses $\mathbf{z}_i(t)$ onto the ground-truth unit bearing, $(\cos \theta, \sin \theta)$. Reasoning state $\mathbf{z}_i(t)$ is read from the last layer of each model for compatibility of comparison; empirically we find that extracting $\mathbf{z}_i(t)$ from a deeper layer improves performance on vague inputs but the relationship between model layer and accuracy is non-linear and this comparison is designed to be relative. A layer-sweep over the selected model enables the optimal layer selection for the given task.

TABLE II: Reasoning-engine tradeoff on zero-shot vague decoding.

Engine	ℓ	d_{LM}	Err. (v. ref)	Latency / VRAM
Qwen3-0.6B (ref)	28	1024	1.00	3.7 s / 1.2 GB
Qwen3-1.7B	28	2048	0.80	4.0 s / 3.3 GB
Qwen3-4B	36	2560	0.65	5.0 s / 7.7 GB

Larger models gain improved heading on the held-out vague prompts at the cost of higher latency and compute required, measured on an RTX A5000; Table II. The simplified probe is trained until each LM obtains a fit of approximately 5° on the training set and is therefore only for comparison and not representative of the final performance in COMPASS. We select Qwen3-1.7B with reasoning readout at $\ell = 27$ as the default configuration for evaluation as it is edge-deployable while generalizing well to unseen vague operator inputs.

1) *Vague Commands*: Piloting swarms of $N = 64$ robots, we sample both in-distribution and held-out prompt templates, e.g., “new heading {surface}” as well as different prompt classes; Table I. These episodes show that COMPASS is able to generalize zero-shot to held-out prompts, Figure 6,

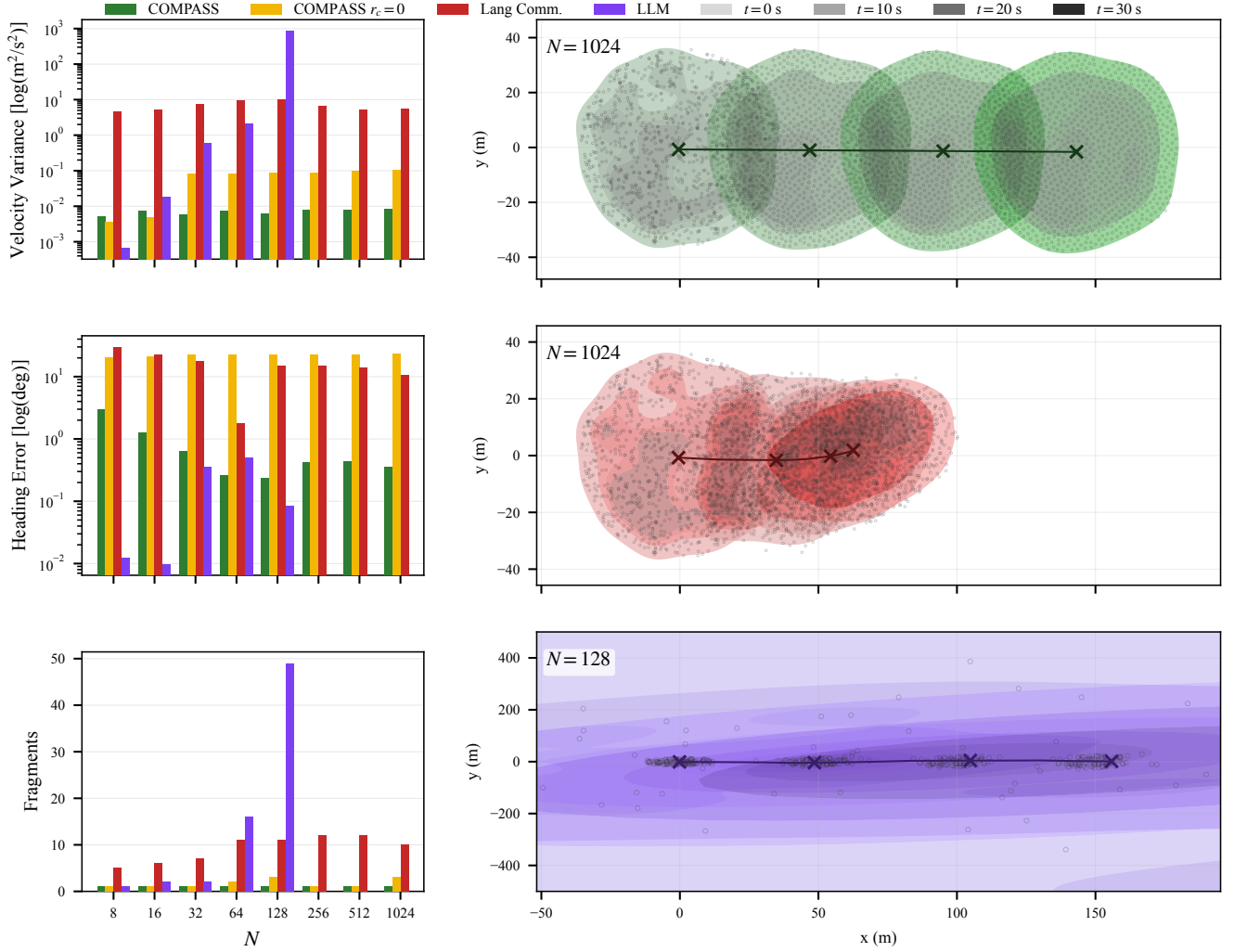


Fig. 3. Scalability evaluation for the prompt "go east" for $N = [8, 16, 32, 64, 128, 256, 512, 1024]$ and 30 s rollouts. Our model, COMPASS (green) is compared against a centralized controller with hand engineered feedback using API calls to Claude Opus 4.8 (purple) [26] and the ablated language-only communication method (red). **Left:** Velocity variance (consensus) and heading error are shown in log-scale and fragmentation measured during the last 15 s of the rollout at each N . **Right:** Rollouts for each policy shown at the largest evaluated scale. The trajectory of the center of mass (COM) overlays the migrating flock, marked with an $\times$ at each of the captured snapshots in time.

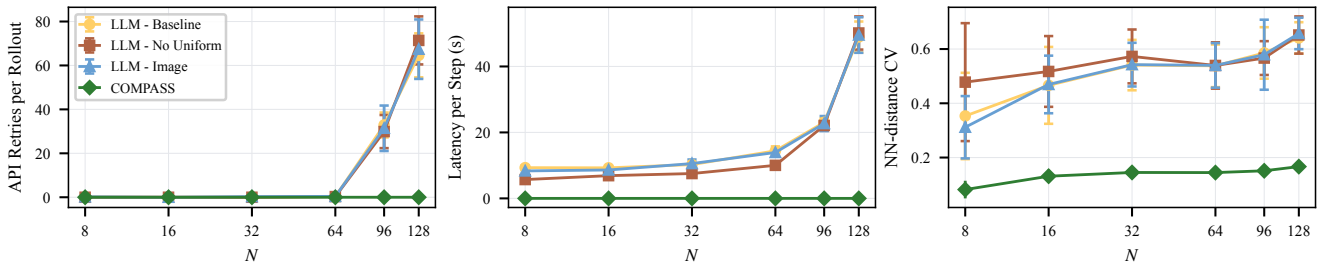


Fig. 4. Performance of Claude Opus 4.8 tasked to flock a team of robots over growing scales in a centralized scenario. The baseline prompt (yellow) requested flocking with uniform distance between agents. The no-uniform prompt (brown) limited the task to . The image prompt (blue) incorporates vision for flocking with uniform distance. We evaluate COMPASS on the same initial conditions and demonstrate that it maintains near-uniform nearest-neighbor spacing and latency as the fleet grows. We use the coefficient of variation of each robot's nearest-neighbor distance, averaged over the rollout, to evaluate the spatial distribution of the fleet across scales.

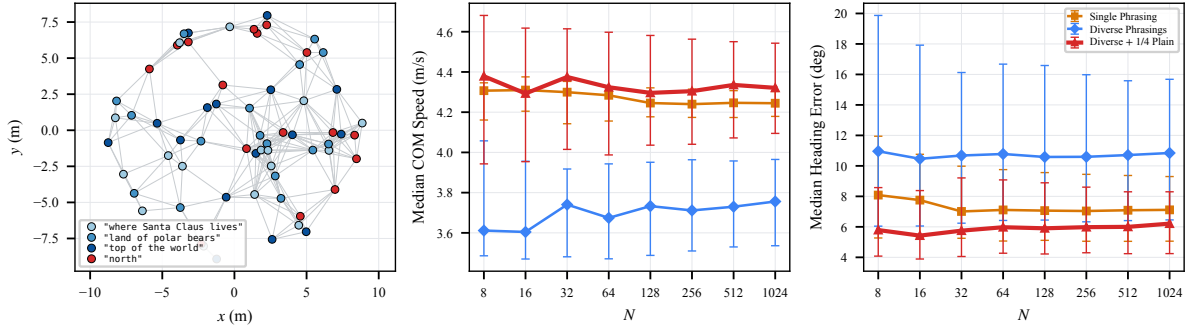


Fig. 5. **Left:** Example initialization where four different phrasings are distributed over the team. A 1/4 of the team receives a simpler command for the LM to reason over. **Middle:** Median COM speed averaged over the trajectory against team size N on a selection of cardinal vague prompts. **Right:** Median settled heading error over N .

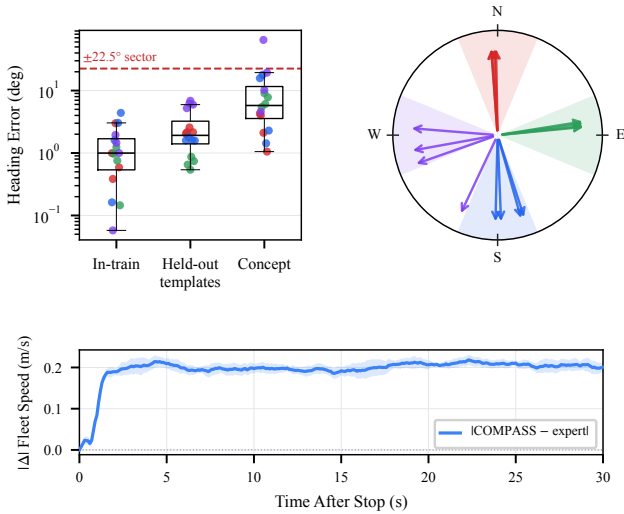


Fig. 6. Zero-shot generalization to new prompt classes ($N = 64$, cardinal directions). **Left:** The box plot shows the settled heading error per evaluation category; strip plot points are colored by the commanded direction. Each class has 16 flights: 4 per cardinal direction. **Right:** the settled headings for the held-out concept commands; arrows are colored according to the commanded direction. **Bottom:** The COMPASS flies the command “go east” for 15 s and then the command switches to “stop” with thinking regenerated at the switch. The trace shows the absolute difference $|\text{COMPASS} - \text{expert}|$ in mean fleet speed over time and over 10 trials with $N = 64$.

while simultaneously maintaining a cohesive flock. Generalizability is a property driven by the reasoning engine. The thinking traces often reveal that smaller models sometimes struggle to select a direction for a vague prompt. For 16 trials, the median heading error per class is: in-training 1.0° , held-out templates 1.9° , concept 5.7° . When the fleet is commanded to stop mid-mission, it retains a small drift velocity of 0.2 ms^{-1} compared to the virtually stationary expert.

2) *Unseen Headings:* For natural language commands, COMPASS is trained on the 8-point compass rose and Stop; does the reasoning state $\mathbf{z}_i(t)$ generalize to new directions zero-shot? We evaluate this by adjusting the wrapper to request reasoning for the 16-point compass rose, which

TABLE III: Generalization to the 16-point rose and numeric bearings.

Direction family	Trials	In sector	Med. err. [$^\circ$]
Cardinals (trained)	20	100%	1.1
Intercardinals (trained)	20	100%	1.2
Sec. intercardinals (zero-shot)	24	75%	7.7
Numeric bearings (zero-shot)	16	N/A	2.0

includes the secondary intercardinal directions such as North-northeast, which bisect the 45° sectors at 22.5° angles; Figure 7 and Table III. Despite being trained on the 8-point rose, the policy generalizes to a 16-point rose without any retraining. The 16-point wrapper itself is not included in the training examples either. COMPASS achieves a mean error of $7.96^\circ \pm 4.41^\circ$ on secondary intercardinals for over 24 trials. While the zero-shot performance degrades, it also hints that $\mathbf{z}_i(t)$ is able to generate outputs for new angles that are better than random directions or the nearest direction on the 8-point rose. The policy does not generalize equally well to every held-out secondary intercardinal direction and has the lowest mean heading error when “east” is given as the first direction, $1.7 \pm 1.0^\circ$ and $3.8 \pm 1.8^\circ$ for ESE and ENE, respectively. Direction NNE is the worst offender, falling outside the sector in 2/3 trials with an average error of $13.2 \pm 2.7^\circ$; overall 18/24 trials have in-sector headings.

COMPASS is trained on 720 numerically expressed bearings wrapped in natural language, with North at $0^\circ/360^\circ$ (aeronautical convention) and clockwise increments, COMPASS is able to ground bearings in $\mathbf{z}_i(t)$ such that out-of-distribution bearings in prompts are flown with mean error $2.2 \pm 1.6^\circ$; Table III. Empirically, we find that training on bearings did not improve the performance on the secondary intercardinals, even if the equivalent bearings are included in the training set. Therefore, grounding of new directions in language is independent of numeric bearings.

V. CONCLUSION

We introduced COMPASS, a decentralized, scalable architecture designed to elicit and harness the collective dynamics of an agentic fleet of robots. Our experiments elucidate the architectural roles: the language channel decides *where to go*

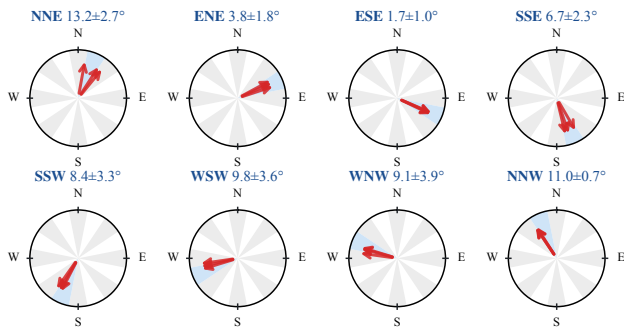


Fig. 7. Settled COM headings (red) for the eight held-out secondary intercardinals, $N = 64$, 60 s rollouts. The shaded sector is the commanded 16-wind sector; each panel reports its mean $\pm$ std error.

and is limited by the biases of the chosen LM. Engineered methods such as structured diversity can be employed to limit that bias, demonstrating performance gains that hold across scales. MAST decides *getting there cohesively*, integrating agentic decisions into motor commands through multi-hop communication at variable N . The reasoning state acts as the downstream interface between the components; the language model never needs to reason spatially. A learned feedback token enables collective information diffused through MAST to appear in the input of the LM for the next reasoning step.

With language steered flocking as the backdrop, we studied how scale and feedback shape the system’s performance. We found that scale synergizes well with language under *structured* diversity and that the wisdom of the crowd is not guaranteed. A diverse mixture of phrasings of similar decoding difficulty does not improve performance, while mixing in an easy phrasing across a quarter of the fleet can promote bias cancellation. Feedback is most effective when it combines a learned representation that the language model may reason over. When introducing feedback into the language channel, small language models that are edge-deployable on SWaP-limited hardware can easily be overwhelmed by too much feedback, i.e., thinking becomes re-tasked. In principle, the COMPASS loop is not task specific and tasks that are verifiable and have a language-expressible objective can operate the same loop.

REFERENCES

- [1] T. Kwon, N. D. Palo, and E. Johns, “Language models as zero-shot trajectory generators,” *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6728–6735, 2024.
- [2] M. Ahn *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” 2022. [Online]. Available: <https://arxiv.org/abs/2204.01691>
- [3] J. Varley *et al.*, “Embodied ai with two arms: Zero-shot learning, safety and modularity,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3651–3657.
- [4] V. S. Dorbala, J. F. Mullen, and D. Manocha, “Can an embodied agent find your “cat-shaped mug”? llm-based zero-shot object navigation,” *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4083–4090, 2024.
- [5] H. Singh, R. J. Das, M. Han, P. Nakov, and I. Laptev, “Malm: Multi-agent large language models for zero-shot robotic manipulation,” in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 20 386–20 393.

- [6] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, “Scalable multi-robot collaboration with large language models: Centralized or decentralized systems?” *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4311–4317, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:263143275>
- [7] F. Shi *et al.*, “Large language models can be easily distracted by irrelevant context,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 31 210–31 227. [Online]. Available: <https://proceedings.mlr.press/v202/shi23a.html>
- [8] X. Guo *et al.*, “Embodied llm agents learn to cooperate in organized teams,” *IEEE Transactions on Computational Social Systems*, 2026.
- [9] S. Agarwal, R. Muthukrishnan, W. Gosrich, V. Kumar, and A. Ribeiro, “LPAC: Learnable Perception-Action-Communication Loops with Applications to Coverage Control,” Feb. 2024.
- [10] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, “Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks,” in *Proceedings of the Conference on Robot Learning*. PMLR, May 2020, pp. 671–682.
- [11] D. Owerko, F. Vatnsdal, S. Agarwal, V. Kumar, and A. Ribeiro, “MAST: Multi-agent spatial transformer for learning to collaborate,” 2025. [Online]. Available: <https://arxiv.org/abs/2509.17195>
- [12] J. Gao *et al.*, “Physically grounded vision-language models for robotic manipulation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 462–12 469.
- [13] Z. Wang *et al.*, “I can tell what i am doing: Toward real-world natural language grounding of robot experiences,” in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: <https://openreview.net/forum?id=iZF0FRPgqf>
- [14] Z. Ravichandran, V. Murali, M. Tzes, G. J. Pappas, and V. Kumar, “SPINE: Online Semantic Planning for Missions with Incomplete Natural Language Specifications in Unstructured Environments,” Mar. 2025.
- [15] W. Huang *et al.*, “Inner monologue: Embodied reasoning through planning with language models,” in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 1769–1782. [Online]. Available: <https://proceedings.mlr.press/v205/huang23c.html>
- [16] Z. Ravichandran, I. Hounie, F. Cladera, A. Ribeiro, G. J. Pappas, and V. Kumar, “Distilling On-device Language Models for Robot Planning with Minimal Human Intervention,” Oct. 2025.
- [17] W. Wang, Y. Li, L. Jiao, and J. Yuan, “Ro-SLM: Onboard small language models for robot task planning and operation code generation,” in *Findings of the Association for Computational Linguistics: ACL 2026*, M. Liakata, V. P. Moreira, J. Zhang, and D. Jurgens, Eds. San Diego, California, United States: Association for Computational Linguistics, Jul. 2026, pp. 34 436–34 460. [Online]. Available: <https://aclanthology.org/2026.findings-acl.1720/>
- [18] S. S. Kannan, V. L. N. Venkatesh, and B.-C. Min, “SMART-LLM: Smart Multi-Agent Robot Task Planning using Large Language Models,” Mar. 2024.
- [19] K. Liu, Z. Tang, D. Wang, Z. Wang, X. Li, and B. Zhao, “Coherent: Collaboration of heterogeneous multi-robot system with large language models,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 10 208–10 214.
- [20] P. Li and L. Zhou, “LLM-Flock: Decentralized Multi-Robot Flocking via Large Language Models and Influence-Based Consensus,” May 2025.
- [21] Z. Mandi, S. Jain, and S. Song, “RoCo: Dialectic Multi-Robot Collaboration with Large Language Models,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. Yokohama, Japan: IEEE, May 2024, pp. 286–299.
- [22] V. L. N. Venkatesh and B.-C. Min, “ZeroCAP: Zero-Shot Multi-Robot Context Aware Pattern Formation via Large Language Models,” Mar. 2025.
- [23] H. Yang, W. Xu, D. Pham, L. Qi, and M. Ba, “Decentralised task planning and motion coordination for scalable multi-robot collaborative manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 100, p. 103255, 02 2026.
- [24] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, “RoFormer: Enhanced transformer with rotary position embedding,”

Neurocomput., vol. 568, no. C, Feb. 2024. [Online]. Available:
<https://doi.org/10.1016/j.neucom.2023.127063>

- [25] A. Yang *et al.*, “Qwen3 technical report,” 2025. [Online]. Available:
<https://arxiv.org/abs/2505.09388>
- [26] Anthropic, “Claude opus 4.8,” 2026, large language model. Accessed:
2026-09-15. [Online]. Available: <https://claude.ai>