ABSTRACT

Title of Thesis: Distributed Swarm Formation Recognition via Local Features and Gossip-Averaged Graph Neural Networks

Samuel Badalov
Master of Science, 2026

Thesis Directed by: Dr. Michael W. Otte
Department of Aerospace Engineering

Distributed swarm formation recognition requires each robot to independently classify the global formation using only local peer-to-peer communication and without access to global position information or a centralized coordinator. We present a fully distributed pipeline in which each robot computes a local feature vector from its one-hop communication neighborhood, passes it through a shared-weight graph neural network, and accumulates a final embedding through fixed-round gossip averaging. We evaluate three feature vector representations across three communication radii. Results demonstrate that structural features inspired by graphlets outperform distance features at sparse connectivity, while distance features dominate at high connectivity. Their concatenated feature vector performs well at all stages, demonstrating their successes as complementary. To our knowledge this is the first fully distributed formation classification problem of such manner.

Distributed Swarm Formation Recognition via Local Features and Gossip-Averaged Graph Neural Networks

by

Samuel Badalov

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2026

Advisory Committee:
Dr. Michael W. Otte, Chair/Advisor
Dr. Aravind Srinivasan
Dr. David Mount

Acknowledgments

I would like to express my gratitude to my advisor, Dr. Michael Otte, for his guidance and help throughout this entire process. I have learned incredible lessons both in my professional and personal life in the last year from the experience the lab has provided me. I would additionally like to thank Dr. David Mount and Dr. Aravind Srinivasan for taking the time to serve on my committee, as well as for their insightful questions and commentary.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1: Introduction

1.1 Background and Motivation

Many multi-robot tasks require robots to understand their collective spatial configuration. This is challenging in swarm scenarios in which robots typically must communicate locally and lack access to global position data. An additional challenge is presented in swarm systems without a global controller, relying entirely on distributed inference.

We address the problem of formation shape recognition under a strict distributed constraint: each robot may only communicate with neighbors within a communication radius r, knowing only the scalar distance between them. No base station is permitted at any stage of inference, and the final shape classification process is computed entirely on each robot. The challenge is that a robot in such a setting sees only a small local window of the global formation and must infer global shape from this incomplete view.

Our key observation is that swarm formations have compositional structure. Each shape may be characterized by the roles of robots within its local structure: shapes such as Star consist of a high-degree hub surrounded by degree-one leaves, Pentagon and Hexagon formations follow ring structure, and BinaryTree or Kite follow complex branching structures as cycles of their local motifs. Motivated by the graphlet literature [29], which showed that local subgraph counts provide topological fingerprints of a node's neighborhood, we design a set of structural features computable entirely from one-hop neighbor communication. We additionally pair these structural features with scalar inter-robot distance statistics, forming a joint 10-dimensional embedding that captures both the topology and geometry of a local neighborhood. Our approach adapts prior graphlet work to local computation as opposed to global knowledge, deploys such computation to online testing, and combines existing topological ideas with geometric distance information, analyzing the successes and failures of both feature types.

We embed these features as node inputs to a shared-weight graph neural network inspired by the distributed GNN architectures of Tolstaya et al. [18] and Gosrich et al. [19], but repurposed for recognition rather than control. A gossip averaging stage [10] replaces centralized global pooling, enabling each robot to accumulate a swarm-level embedding through repeated local message exchange. Each robot then independently classifies the formation without any coordinator.

We evaluate this pipeline across ten formation classes and three communication radii, demonstrating that formation self-awareness can emerge entirely from local peer-to-peer communication among robots with no global knowledge.

1.2 Related Work

Our work combines ideas from several research areas in swarm intelligence and machine learning for multi-robot systems. We now briefly survey relevant prior work motivating distributed inference and graph neural networks in swarm systems, as well as the grammar and topology based inspirations for node embeddings.

1.2.1 Kilobot Swarm Systems

The communication and GPS constrained environment in simulation is motivated by deployment on Kilobot systems, generally deployed lacking access to global coordinate systems and relying on local message passing. The Kilobot was introduced by Rubenstein et al. [1] as a low-cost robot designed for testing collective algorithms in large multi-robot systems. Rubenstein et al. [2] further demonstrated scalability specifically through Kilobots that form user-defined formations using seed robots placed manually to define helpful auxiliary information. The use of seed robots motivated initial implementations of our algorithm but was later altered [3]. Several works have deployed distributed or communication constrained algorithm approaches on Kilbot hardware [4–6]. Otte [7] demonstrated that a neural network can be trained and executed fully distributed across a Kilobot swarm without a central coordinator, establishing that learned inference is achievable under the same local communication constraints as our work assumes. Pluhacek et al. [8] explored global coordinate construction on Kilobots through local message passing and distance measurement, requiring robot arrangement in lattice form, motivating a similar global reconstruction approach across varying structures.

1.2.2 Distributed Inference in Swarm Systems

A central challenge in swarm robotics is enabling robots to reason about global properties using only local observations. Swarm systems may exist in an environment without GPS or access to a global topology, and therefore rely on distributed inference. Desai [9] firstly established graph-theoretic foundations for robot formations, motivating the representation of a swarm as a communication graph. Olfati-Saber et al. [10] proved that repeated local averaging over a connected graph converges to a global consensus, with similar conclusions being presented in Xiao et al. [11] and Zhang et al. [12] in regards to sensor fusion and time-varying communication respectively, collectively motivating the theoretical basis for our use of fixed-round gossip averaging as an alternative to global pooling. Cao et al. [13] demonstrated this principle in practice, achieving multi-robot SLAM through distributed updates over local communication alone. Chinchali et al. [14] further highlighted communication overhead as a fundamental design constraint, motivating architectures that minimize broadcast rounds. The most directly relevant prior work is spectral swarm robotics [15], in which Kilobot swarms estimated the second Laplacian eigenvalue through decentralized diffusion to classify arena shape without a base station. In all of these works, robots either reason about the external environment, require mobility to sample a space, or produce a single aggregate statistic. We seek to extend these findings to the classification of stationary formation formed by the robots themselves.

1.2.3 Graph Neural Networks for Multi-Robot Systems

Graph neural networks have increased in popularity as newfound tools for relational reasoning in robotics; the communication topology directly defines a graph suitable for message passing. Kim et al. [16] demonstrated that representing the structure of a robot through a graph allows generalization across unique topologies through GNN message passing. Miao et al. [17] validated further that graph features can be leveraged through GNNs for relational reasoning. Most directly relevant to our setting, Tolstaya et al. [18] applied shared-weight aggregation GNNs to learn decentralized flocking controllers strengthened by multi-hop aggregation. Gosrich et al. [19] extended this to coverage tasks, demonstrating that such policies transfer across team sizes and sensing radii. We additionally survey a range of works in graph neural network development to better understand GNN structure and the particular importance of several message passing schemes, as well as specific methodology for node embedding systems in permuted graphs [20–24]. These works motivate the message passing structure and validate choices of hyperparameters such as dropout.

1.2.4 Grammar-Based Structural Reasoning in Robotics

Our use of feature embeddings distinguishing distance and topological features is inspired by formal grammar architecture applied to compositional structure in robots. Zhao et al. [25] demonstrated in RoboGrammar that graph grammars can automate robot morphology design by encoding valid assembly rules as production rules over a part vocabulary. Steels [26] investigated how grammar-like communication protocols emerge through local interaction in autonomous robot populations. Though our work does not implement a grammar in the formal sense, we are motivated by the observation that robot formations and communication have compositional structure [27].

1.2.5 Structural Node Features

The use of local subgraph structure as node features has multiple roots in graphical analysis. Pržulj [28] introduced the use of graphlets as small connected non-isomorphic induced subgraphs of a larger network, later extending to graphlet degree vectors counting a certain nodes participation in subgraphs, providing topological fingerprints across its network [29]. Shervashidze et al. [30] used a similar process for graph comparison through efficient graphlet kernels. Bouritsas et al. [31] developed Graph Substructure Networks (GSN), augmenting node features with induced subgraph pattern counts and provably increasing expressiveness. These works all share the assumption of global graph access at feature computation time.

1.3 Contributions

This work provides several contributions to the field of distributed analysis in multi-robot systems. Similar prior work on distributed pattern recognition on hardware in Kilobot-like robotic systems or graph topology requires either seed robots, a base station, or centralized control. We propose a fully distributed formation classification protocol through graph neural network message passing and gossip averaging rather than global pooling. To our knowledge, this is the first entirely distributed formation classification problem under the communication and GPS constraints set. Furthermore, our work relies solely on a one-hop message passing system that does not require robots to be part of a fully connected communication graph.

The work additionally provides an analysis on different feature embeddings, both scalar and topological, as well as a comparison of the two separately and concatenated. We provide analysis on when structural information derived from communication graph patterns may be useful in formation classification, motivating further work on graphlet-based embeddings for multi-agent systems represented as a graph neural network.

1.4 Thesis Outline

Chapter 2 outlines the problem formation, detailing the representation of the swarm system as a communication graph G as well as describing the shape objective classification problem and the varying constraints following a simulation environment that reflects Kilobot deployment.

Chapter 3 formally defines the locally computable features described in Section 1.1, outlining three types of features that will be compared in later sections. We compare 5D distance, 5D structural, and their 10D combined vectors, with choices for each specific feature being described further.

Chapter 4 describes the graph neural network architecture to be used at both training and test time. This chapter covers the full three-stage network mapping each robots initial feature vector to a predicted shape label using strictly local communication. We describe the message passing layers and their respective hyperparameters, the gossip averaging approach as replacement to global pooling, and the final per-robot classification stage. We additionally provide a step-by-step algorithm for the entire pipeline.

Chapter 5 describes the experimental setup through both the dataset used and training procedure. Firstly, we detail the design of the dataset and provide visual representation of all ten shapes across medium communication radius. We additionally compare four specific shapes across varying communication radii to provide insight as to how graph topology changes as communication edges densify. Finally, we describe the training procedure in both the dataset used and any important hyperparameters.

Chapter 6 presents the results of simulation experiments across nine graph neural network models, with a comparison of the three feature vectors described in Chapter 3 and three communication radii described in Chapter 5. The chapter additionally presents an analysis of the ring shapes earlier described leading to a final discussion regarding the use of scalar distance and structural topology in node embeddings.

The paper is then concluded in Chapter 7, describing the implications of results presented in Chapter 6 as well as limitations of the approach and future work.

Chapter 2: Problem Formulation

We consider a swarm of n robots, with positions $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n \in \mathbb{R}^2$, forming a geometric configuration in correspondence to one of multiple predefined shape classes. Individual robots do not have access to their global location or position information, relying solely on local sensing and short-range communication with nearby neighbors. In the following, we formalize the swarm as a communication graph, define local neighborhoods under range constraints, and specify the shape classification objective for each robot.

2.1 Communication Graph

The swarm can be modeled as an undirected communication graph $G = (V, E)$, with each vertex $v \in V$ representing a robot and each edge $(u, v) \in E$ indicating that robots u and v can communicate directly. This communication is limited by range:

$$(u, v) \in E \iff \|\mathbf{r}_u - \mathbf{r}_v\|_2 \leq r,$$

where r is the maximum communication radius and $\mathbf{r}_u, \mathbf{r}_v \in \mathbb{R}^2$ are the positions of robots u and v, respectively.

2.2 Local Neighborhoods

We define the local neighborhood of robot v as

$$N(v) = \{u \in V \mid (u, v) \in E\}, \qquad d(v) = |N(v)|$$

which contains all robots that v can communicate with directly.

While a robot v is then able to exchange information freely with any member of $N(v)$, it can access information from more distant robots only through multi-hop communication along the edges of G.

2.3 Shape Classification Objective

In this work, we use $c = 10$ shape classes $y \in \mathcal{S} = \{s_1, s_2, \ldots, s_c\}$. Each robot v independently produces a local prediction $\hat{y}_v \in \mathcal{S}$ using only information gathered through direct communication within its neighborhood. The swarm is considered to have correctly classified its formation if and only if every robot independently arrives at the correct label:

$$\hat{y}_{\text{swarm}} = y \iff \forall v \in V,\ \hat{y}_v = y.$$

Our objective imposes two hard constraints: (1) no robot has access to global position information or robot information outside of its neighborhood, and robots do not have knowledge of their full graph G unless their communication radius allows; (2) no base station or centralized controller is permitted to aggregate robot observations or intermediate embeddings, and the final swarm-level output emerges entirely from local computation and peer-to-peer communication.

Chapter 3: Local Feature Computation

For each robot v to independently classify the swarm formation it must derive some feature vector of meaningful information representing its local position within the graph G. However, v lacks access to its global coordinates, and therefore every feature must be derived from connectivity structure and distances of robots within $N(v)$. We evaluate three feature representations: a 5D vector capturing scalar distance relationships within the neighborhood, a 5D structural vector motivated by graphlet analysis of local topology, and their concatenated 10D vector.

3.1 5D Distance Features

The most direct information available to robot v is the set of distances to its neighbors. Robot v therefore is able to construct a 5D distance feature vector $\mathbf{x}^{\text{dist}}(v) = (\bar{d},\, d_{\min},\, d_{\max},\, \sigma_d,\, \delta)$, where $\bar{d}$, $d_{\min}$, $d_{\max}$, and σ_d are the mean, minimum, maximum, and standard deviation of pairwise distances to all $u \in N(v)$, and $\delta = |N(v)|$ is the degree of v. These statistics summarize the geometric spread of the local neighborhood, and are sensitive to changes in scale and formation density.

3.2 5D Structural Features

Drawing from the topology of a communication graph described in Sections 1.2.3 and 1.2.5, we observe that for each vertex v within our graph its local connectivity reveals its structural role within the formation. Each of our ten shapes is assembled from a small vocabulary of recurring local structures: a Triangle can be represented by 3 mutually connected robots, a Star can be represented as a single high-degree robot surrounded by robots with $N(v) = 1$, a Pentagon and Hexagon can both be represented as a Ring with $N(v) = 2$ for every robot. We can therefore capture this topology through the computation, for each robot v, of a set of graphlet counts over its included one-hop subgraph. These counts will describe the local motifs v participates in and therefore characterize its role within the formation structurally.

Let A denote the adjacency matrix of G. The related 5D structural feature vector $\mathbf{x}^{\text{struct}}(v) = (\delta,\ \lambda,\ \gamma,\ \rho,\ \eta)$ is defined in Table 3.1.

Table 3.1: 5D structural feature vector components for robot v.

Feature	Definition	Description
$\delta(v)$	$\lvert N(v) \rvert$	Number of robots within communication range.
$\lambda(v)$	$\frac{1}{2}\lvert\{(j,k) : j,k \in N(v),\ j \neq k,\ A_{jk} = 0\}\rvert$	Pairs of neighbors not directly connected.
$\gamma(v)$	$\frac{1}{2}\lvert\{(j,k) : j,k \in N(v),\ j \neq k,\ A_{jk} = 1\}\rvert$	Pairs of neighbors directly connected.
$\rho(v)$	$\lvert\{w : \exists j \in N(v),\ w \in N(j),\ w \notin N(v),\ w \neq v\}\rvert$	Distinct robots reachable in two hops from v but outside $N(v)$.
$\eta(v)$	$\mathbf{1}[\delta(v) \geq 3 \ \wedge\ \lambda(v) > \gamma(v)]$	Binary hub indicator: high degree with $\lambda(v) > \gamma(v)$.

3.3 10D Combined Features

We form the 10D combined feature vector as being the concatenated distance and structural vectors:

$$\mathbf{x}^{\text{comb}}(v) = \left[\mathbf{x}^{\text{dist}}(v) \parallel \mathbf{x}^{\text{struct}}(v)\right] \in \mathbb{R}^{10}.$$

This allows the GNN to draw on both the geometric content of the neighborhood and its topological structure, and serves as a direct test of whether the two sources of information are complementary.

Chapter 4: Distributed GNN Architecture

Each robot v runs an identical copy of a three-stage neural network that maps its local feature vector to a predicted shape label using strictly communication with its neighbors. The three stages are as follows: (1) GNN message passing, in which v updates its embedding through aggregation with its neighborhood; (2) gossip averaging, in which the embedding is propagated across the swarm through 5 rounds of averaging; (3) per-robot classification, in which each robot maps its gossip-averaged embedding to a predicted shape label. This process at test time requires no central controller or global position information, as well as no inter-robot coordination beyond local message exchange. Figure 4.1 illustrates the formation-to-graph construction and local feature computation that feed into this pipeline. Following the introduction of each stage in Sections 4.1 – 4.3, Figure 4.2 presents the full pipeline for stages two through four.

4.1 Message Passing Layers

Message passing consists of three stacked graph convolution layers. In each layer, a robot v applies a shared linear projection to the mean of its own and its neighbors embeddings. This is followed by a ReLU nonlinearity and dropout ($p = 0.3$). The hidden dimension is 16 throughout every layer, and every weight matrix is shared across all robots, allowing each robot to perform computation over its own local neighborhood. This returns a 16-dimensional embedding that summarizes the local structure of $N(v)$.

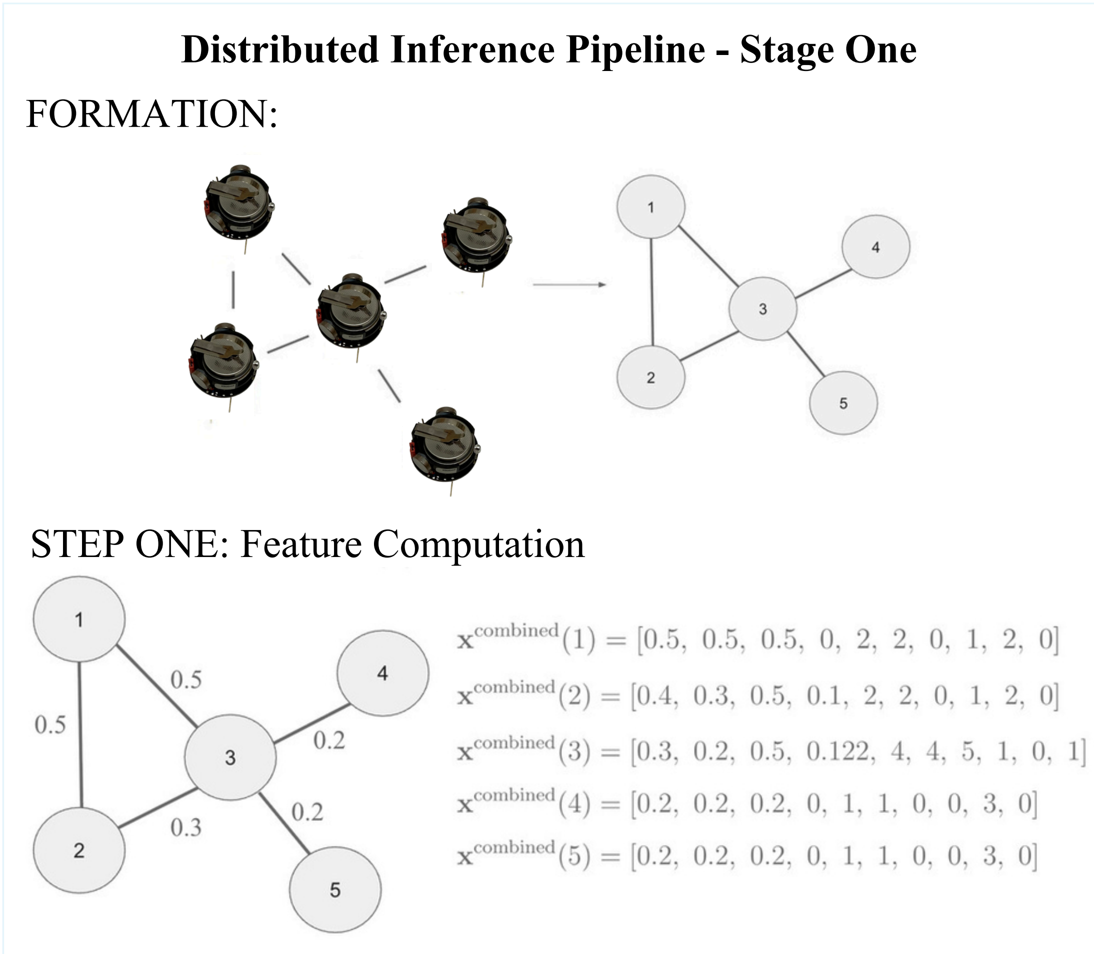


Figure 4.1: Distributed inference pipeline, stage one: a possible Plus formation of five Kilobots is modeled as a communication graph $G = (V, E)$, and each robot v independently computes its 10D combined feature vector $\mathbf{x}^{\text{combined}}(v)$ from local distances and graphlet counts within $\mathcal{N}(v)$.

4.2 Gossip Averaging

For each robot v to make an informed and accurate prediction of its shape class, it benefits v to have a final vector embedding using global pooling. However, due to communication restriction, global pooling is not possible.

To propagate information beyond the message passing horizon without a central controller, we apply five rounds of gossip averaging after the message passing stage. In each round, every robot broadcasts its current embedding to its neighbors, and updates that embedding using the mean of its received embeddings and its own. Unlike the message passing layers, gossip averaging uses no learning weights. Through five rounds, information is able to travel up to five hops from v, which exceeds the diameter of all shape formations in our dataset and provides substantial coverage. Therefore, after five rounds, each robot holds an embedding that approximates the global mean.

4.3 Per Robot Classification

Each robot independently passes its gossip-averaged embedding through a two-layer fully-connected classifier with a hidden dimension of 64, ReLU activation, and dropout ($p = 0.5$) during training. The output is a distribution over the ten shape classes, from which each robot independently takes the argmax as its local prediction $\hat{y}_v$. The swarm-level prediction is then obtained by majority vote over all per-robot predictions as defined in Section 2. All classifier weights are shared across robots, consistent with the fully distributed design of the preceding stages.

4.4 Distributed Inference Pipeline & Algorithm

In this section we present the distributed inference pipeline executed identically by each robot v at test time. Figure 4.2 illustrates stages two through four of the pipeline (GNN message passing, gossip averaging, and per-robot classification) following the local feature computation described in Figure 4.1, run on one highlighted robot within the system. Algorithm 1 formalizes all four stages as a sequence of local operations, confirming that no stage requires global position information or central controller.

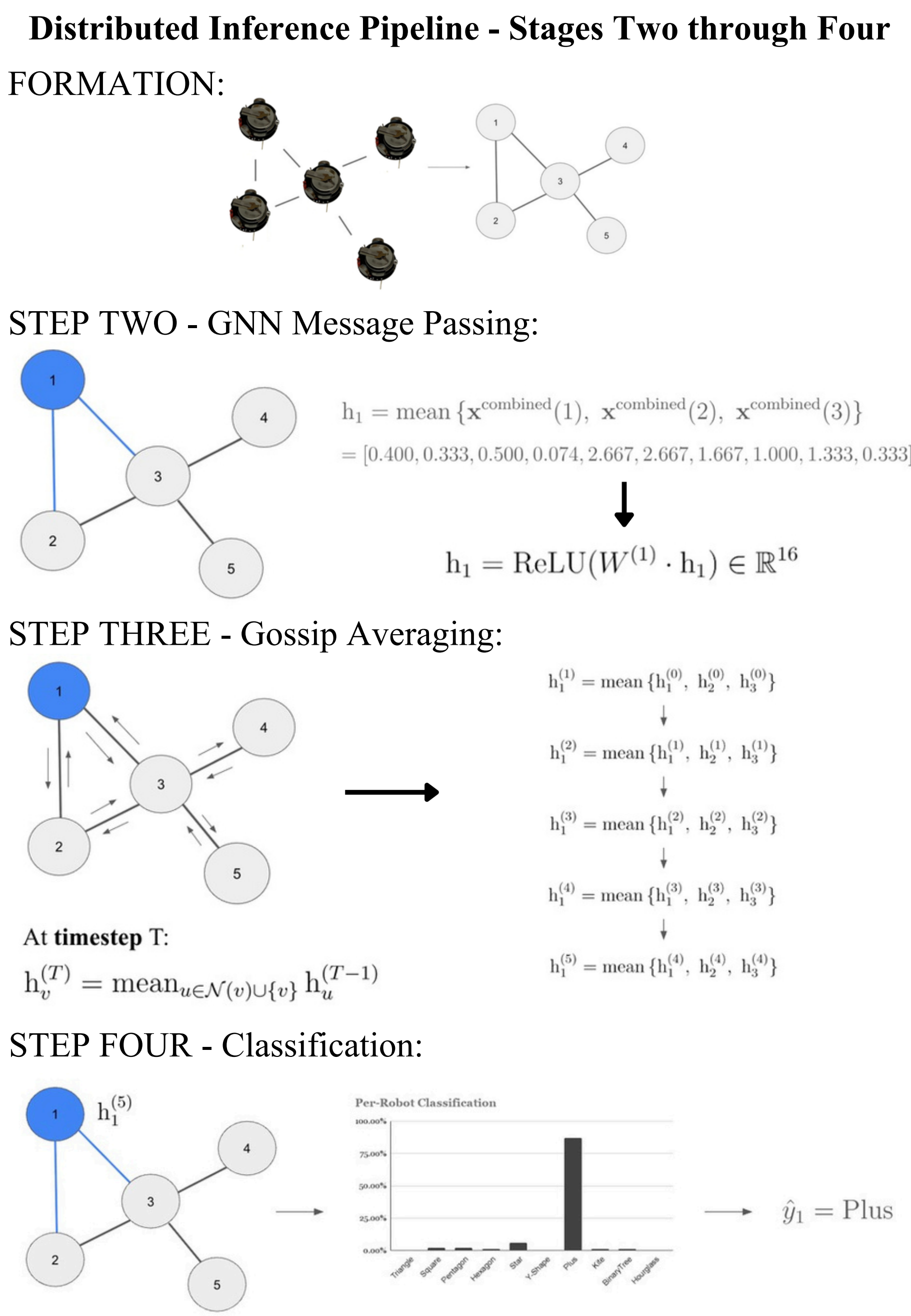


Figure 4.2: Distributed inference pipeline, stages two through four: GNN message passing maps local feature vectors to 16-dimensional embeddings, gossip averaging propagates embeddings across the swarm over five rounds, and each robot independently classifies its averaged embedding to produce a shape prediction $\hat{y}_v$.

Algorithm 1 Distributed Shape Classification (per robot v)

STEP ONE: Feature Computation

Discover neighbors $\mathcal{N}(v)$ via local broadcast

Compute feature vector $\mathbf{x}_v \in \mathbb{R}^5$ (or $\mathbb{R}^{10}$) from distances and graphlet counts over $\mathcal{N}(v)$

STEP TWO: GNN Message Passing

$\mathbf{h}_v \leftarrow \mathbf{x}_v$

for each of 3 GNN layers **do**

  Broadcast $\mathbf{h}_v$ to neighbors; receive $\mathbf{h}_u$ from all $u \in \mathcal{N}(v)$

  $\mathbf{h}_v \leftarrow \mathrm{ReLU}\big(W^{(k)} \cdot \mathrm{mean}\{\mathbf{h}_u : u \in \mathcal{N}(v) \cup \{v\}\}\big)$ $W^{(k)} \in \mathbb{R}^{16\times d}$

end for $\mathbf{h}_v \in \mathbb{R}^{16}$

STEP THREE: Gossip Averaging

for each of 5 gossip rounds **do**

  Broadcast $\mathbf{h}_v$ to neighbors; receive $\mathbf{h}_u$ from all $u \in \mathcal{N}(v)$

  $\mathbf{h}_v \leftarrow \mathrm{mean}\{\mathbf{h}_u : u \in \mathcal{N}(v) \cup \{v\}\}$

end for $\mathbf{h}_v \in \mathbb{R}^{16}$

STEP FOUR: Classification

$\mathbf{z} \leftarrow \mathrm{ReLU}(C_1 \cdot \mathbf{h}_v)$ $C_1 \in \mathbb{R}^{64\times 16}$

$\hat{y}_v \leftarrow \mathrm{argmax}\,\mathrm{softmax}(C_2 \cdot \mathbf{z})$ $C_2 \in \mathbb{R}^{10\times 64}$

return $\hat{y}_v$

Chapter 5: Experimental Setup

In this section we will detail the dataset creation process for the problem formulated in Section 2 as well as the general training procedure given such a dataset.

5.1 Dataset

We design a dataset of ten synthetic swarm formations to span a range of structural complexity and represent different important features: Triangle, Square, Pentagon, Hexagon, Star, Y-Shape, Plus, Kite, Binary-Tree, and Hourglass. These formations are shown in Figure 5.1 at a communication radius r = 1.6, with edges representing $N(v)$. These formations range from n = 3 to n = 7 robots and are designed to maintain topological structure variability. However, certain exceptions remain. Four pure-ring shapes (Triangle, Square, Pentagon, Hexagon) are locally indistinguishable at smaller communication radii and therefore serve as representations of the importance of scalar distance information combined with topological structure.

Each formation is evaluated at three communication radii: SHORT ($r = 1.2$), MEDIUM ($r = 1.6$), and LARGE ($r = 2.5$). The radius determines which robots can communicate directly and therefore defines the structure of the graph G on which the GNN operates.

For each shape and radius we generate 120 samples by placing robots at their canonical positions and applying additive Gaussian position noise $\mathcal{N}(0, 0.06^2)$ so that shapes are not perfectly aligned in every instance and uniform scale variation $\mathcal{U}[0.75, 1.25]$ so that message passing does not generalize on exact scalar distance values. Samples are filtered to retain only connected graphs. The dataset is split 80/20 by stratified sampling, yielding 96 training and 24 test samples per shape class per radius.

To illustrate how radius affects graph connectivity, Figure 5.2 shows four representative formations

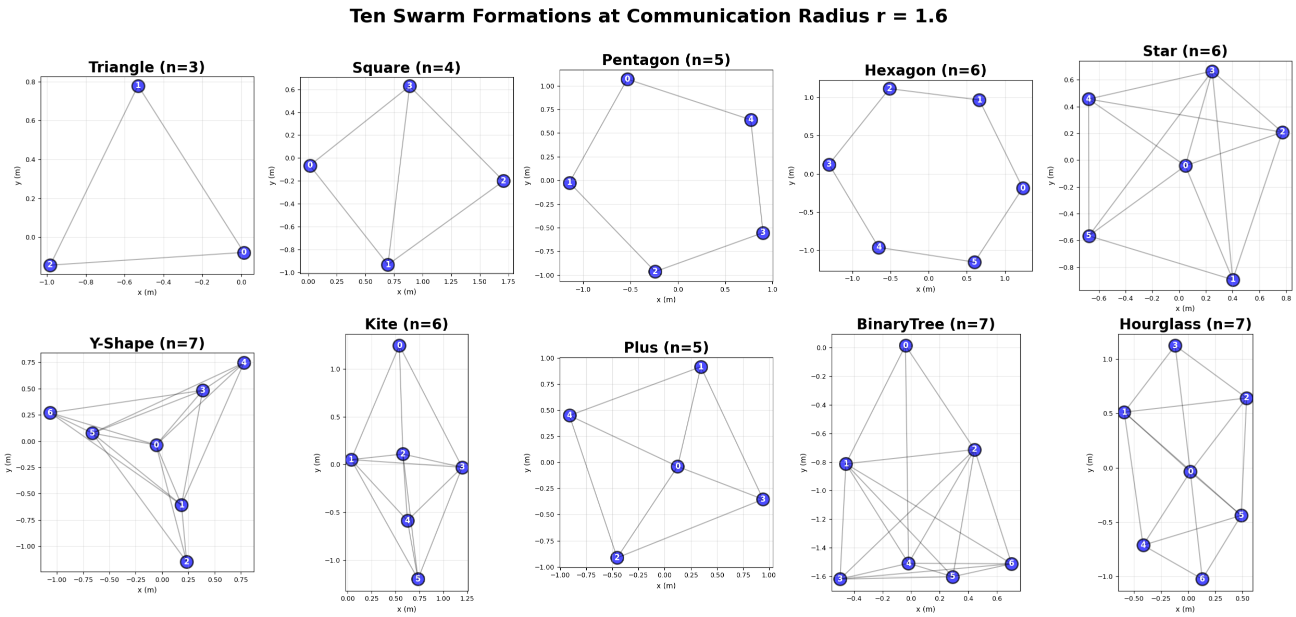


Figure 5.1: Ten swarm formations used in the dataset, shown at communication radius $r = 1.6$. Nodes are robots; edges indicate direct communication links.

across all three communication radii. At SHORT ($r = 1.2$), graphs are sparse and robots see only their immediate neighbors. At MEDIUM ($r = 1.6$), connectivity increases and intermediate robots begin to share neighbors. At LARGE ($r = 2.5$), every robot shares a communication edge, collapsing distinct topological roles into a near-complete graph. These differences motivate our analysis of all three initial feature vectors across all three radii.

5.2 Training Procedure

We train nine independent models, one for each combination of communication radius (SHORT, MEDIUM, LARGE) and feature type (5D distance, 5D structural, 10D combined). Each model is trained from a random initialization with no weight sharing across configurations, so that results reflect the capacity of each feature–radius pairing in isolation.

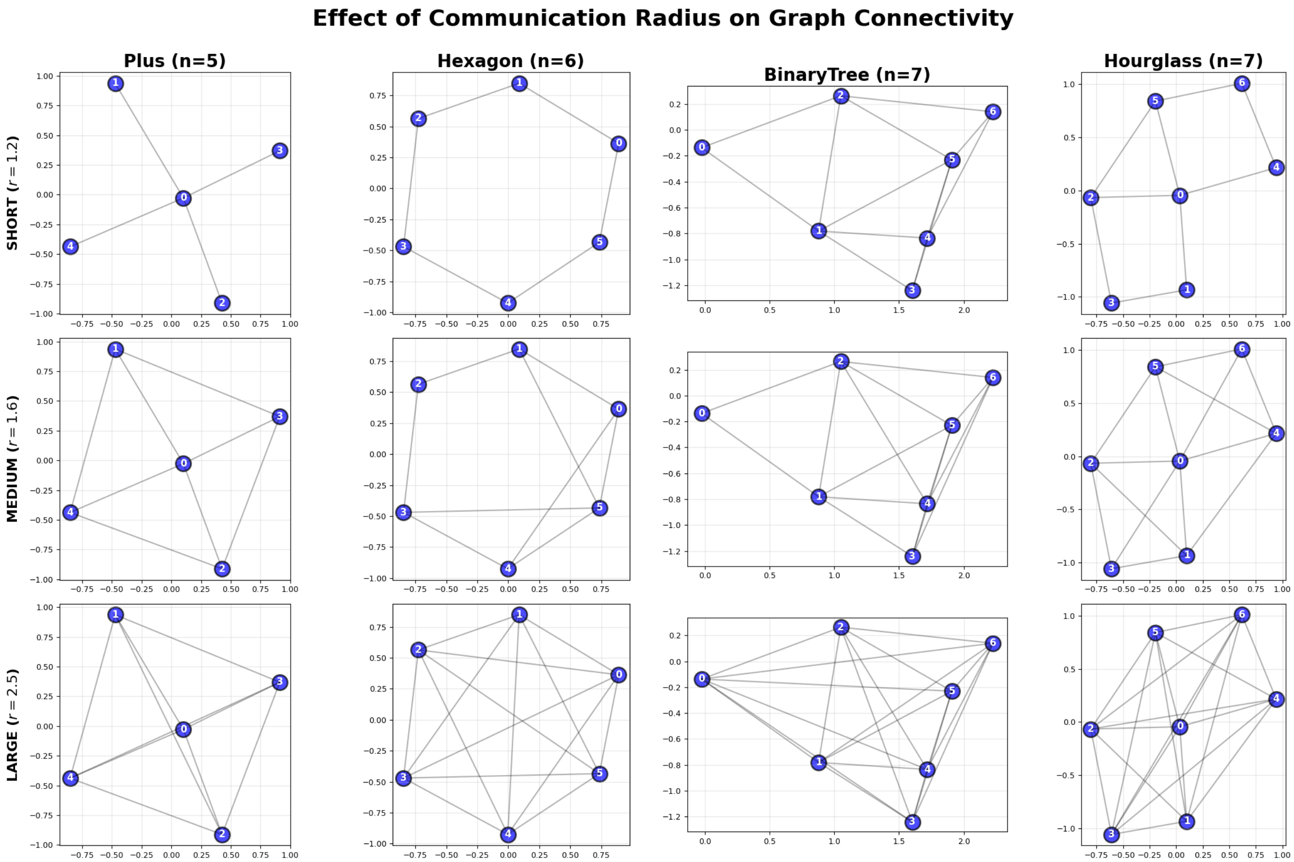


Figure 5.2: Four formations shown at all three communication radii (SHORT $r = 1.2$, MEDIUM $r = 1.6$, LARGE $r = 2.5$). Increasing radius densifies the communication graph, reducing topological variability across robot roles.

All models share the same architecture and hyperparameters. We minimize the negative log-likelihood (NLL) loss using the Adam optimizer with learning rate 10^{-3} and weight decay 5×10^{-4}. Training runs for 300 epochs with a batch size of 32 graphs. Dropout is applied after each GNN layer at rate $p = 0.3$ and after the first fully-connected classifier layer at rate $p = 0.5$. Each model is trained and evaluated on the split described in Section 5.1: 96 training and 24 test samples per class per radius.

Chapter 6: Results

We evaluate all nine model configurations across three communication radii and three feature types, reporting swarm-level classification accuracy, per-robot training dynamics, and per-class confusion structure. All reported accuracies are averaged over three independent random seeds and presented as mean $\pm$ standard deviation. We then examine the ring-shaped subclasses in detail before closing with a broader discussion of the results.

6.1 Training Dynamics

Figure 6.1 shows the NLL training loss and training accuracy across 300 epochs for all nine configurations. Shaded bands indicate ± 1 standard deviations the three seeds.

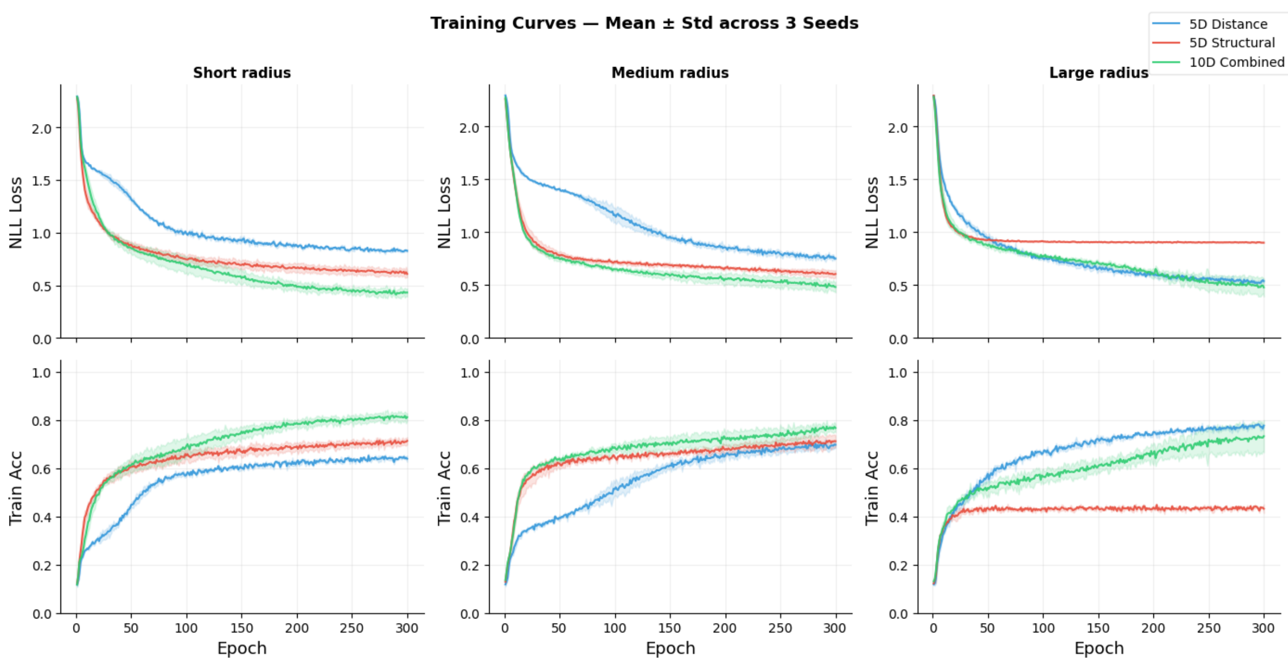


Figure 6.1: Training loss (top) and per-swarm training accuracy (bottom) for all nine model configurations across three communication radii. Lines show the mean over three seeds; shaded bands show ± 1 standard deviation.

We observe that 5D structural and 10D combined features converge to lower loss and higher accuracy than 5D distance at SHORT and MEDIUM radii. Furthermore, we observe these features showing a sharp initial drop in loss followed by relatively smooth convergence to a plateau, while distance features fluctuate throughout training early prior to convergence. This suggests the discrete integer-value structural features induce a more well conditioned landscape, in which the gradient descent process consistently points towards a reliable minimum.

5D distance features show greatest variance in both NLL loss and training accuracy at MEDIUM radius, indicated by a larger shaded region around epoch 100 to 150. This feature is most sensitive to perturbations in weight initialization at intermediate connectivity, most likely due to distance features at medium connectivity showing more heterogeneous behavior. As the global communication graph transitions between sparse and dense configuration distance statistics span their widest range across robots due to large variance in the size of each per-robot neighborhood. This variance affects distance features disproportionately relative to graphlet features as continuous distance statistics shift gradually as opposed to discrete graphlet statistics representing integer density. 10D combined features show similar variance patterns throughout, most likely due to the inclusion of the 5D distance component within the combined vector, which inherits the same sensitivity to weight initialization at intermediate connectivity.

We additionally observe that 5D structural features exhibit different training dynamics from all other configurations at LARGE radius, with a sharp flattening within the first 50 epochs rather than a gradual descent to a stable plateau. This early and abrupt plateau is not a symptom of underfitting or insufficient training duration but reflects a fundamental ceiling imposed by feature collapse at high connectivity. This specific collapse will be further described in Section 6.2.

6.2 Swarm Classification Accuracy

Figure 6.2 reports the swarm-level classification accuracy across all nine configurations. A prediction is counted as correct only if every robot in the swarm independently outputs the correct label.

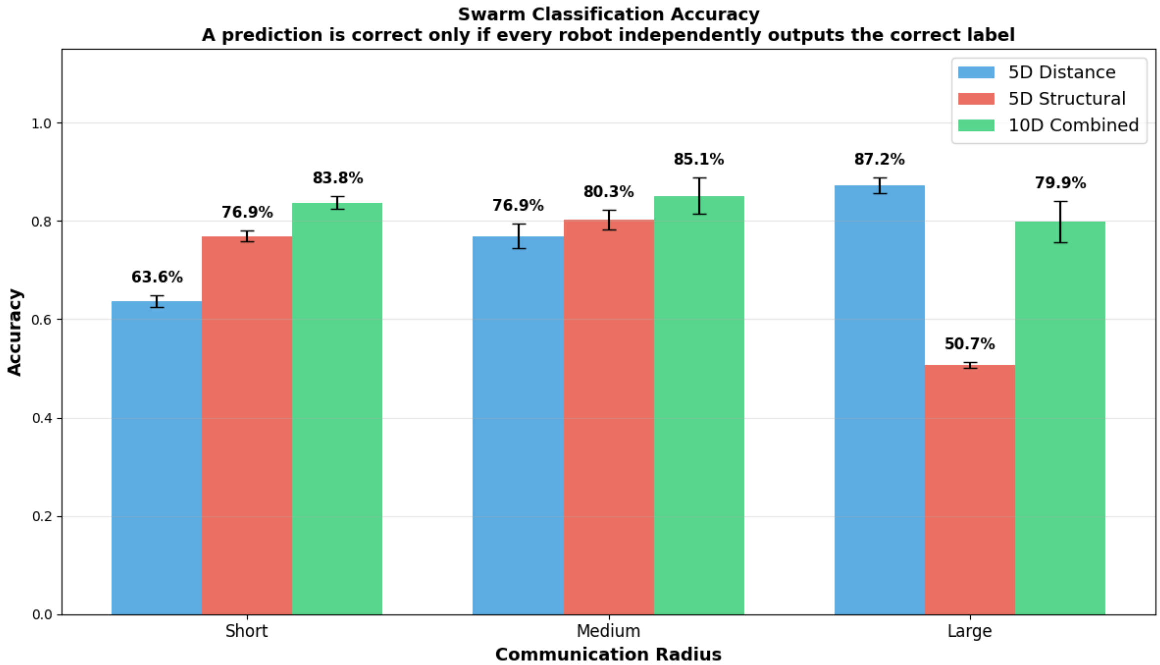


Figure 6.2: Swarm classification accuracy for all nine model configurations. Error bars show ± 1 standard deviation across three seeds. A swarm prediction is correct only if every robot independently outputs the correct label.

5D structural features and 10D combined features outperform 5D distance features by roughly 13% and 20% respectively at SHORT radius. At this radius, the communication graph is sparse, and global prediction relies on robot information gathered from a very small subset of neighbors. Structural features are able to encode information beyond their immediate neighborhood, and therefore have an upper hand on distance features strictly limited to scalar information regarding their direct neighbor. A contributing factor of this significant increase in accuracy is further described in the analysis of ring shapes in Section 6.4.

We observe a general increase in accuracy as communication radius increases for 5D Distance features. These features naturally encode more information as the density of edges within the communication graph grows as both the mean and standard deviation benefit from greater global knowledge. While 5D distance at LARGE radius exhibits the highest swarm-level accuracy across all configurations, it is important to note at such high connectivity the communication graph becomes complete, with the per-robot neighborhood consisting of every robot in the swarm. This type of connectivity is unlikely in distributed multi-robot settings, and merely proves the importance of global knowledge.

This general increase is not observed in structural features. 5D structural features entirely collapse at LARGE radius and 10D combined features show a slight reduction in accuracy, most likely held up by the concatenated 5D distance features. At such high communication radius, as displayed in Figure 5.2, the features become diluted. With every robot sharing a direct edge to all other robots in the communication graph $\lambda(v)$ reduces to 0, as no pairs of neighbors are not directly connected. Features $\eta(v)$ and $\rho(v)$ additionally become uninformative, causing features $\delta(v)$ and $\gamma(v)$ to carry all the available information. The only structural information therefore available is the total number of robots in the swarm.

At MEDIUM radius, we observe a gradual increase in accuracy from 5D distance to 5D structural to 10D combined features. At MEDIUM radius the communication graph is neither so sparse that each robot sees only immediate ring neighbors, nor so dense that all local structure collapses, allowing both distance and structural feature embeddings to encode useful information. This information, when then concatenated to form the 10D combined feature vector, combines to achieve 85% mean accuracy. Each feature contributes complementary discriminative signals simultaneously, producing a representation that is more powerful than either component alone.

Finally, we observe the 10D combined vector performing well across all radii, regardless of the collapse of either of its components. It seems as though the strong signals provided by either feature when correct outweigh negative signals from the other.

6.3 Per-Class Confusion Analysis

Figure 6.3 shows the confusion matrices for all nine configurations, with counts summed across the three seeds to produce more reliable estimates of per-class error patterns.

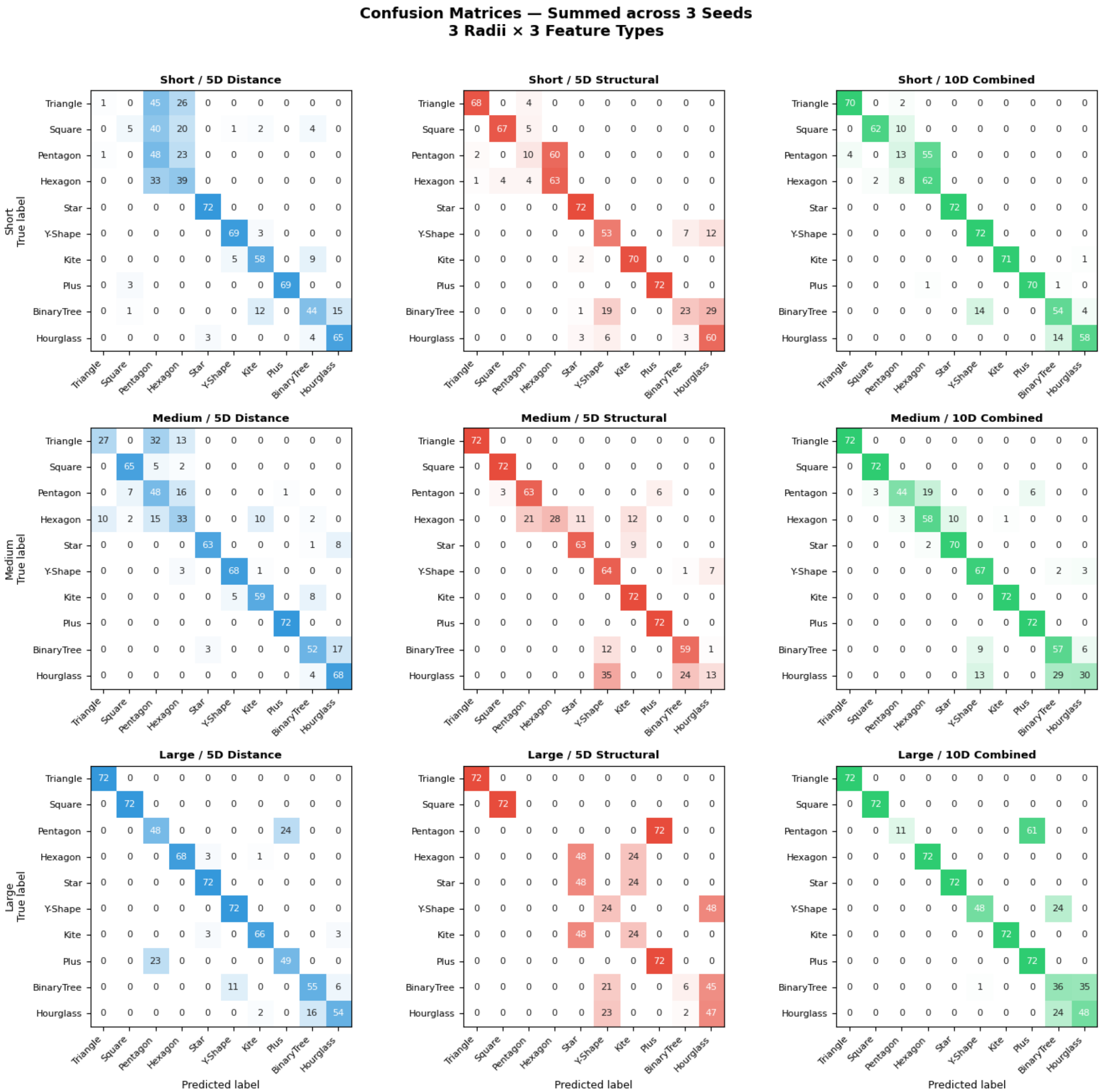


Figure 6.3: Confusion matrices for all nine model configurations, with counts summed across three seeds. Rows indicate true labels; columns indicate predicted labels.

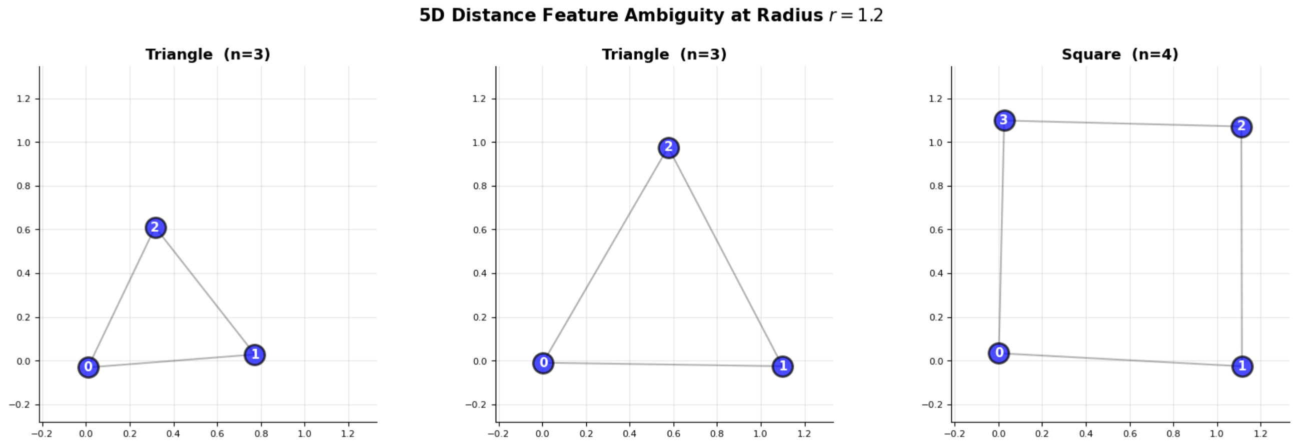
Firstly, we observe an inability to classify ring structures at SHORT radius. 5D distance features are unable to distinguish between any of the ring-like shapes and 5D structural and 10D combined features struggle specifically with Pentagon and Hexagon confusion. Further analysis into the difficulties of this problem and the advantages of structural embeddings are provided in Section 6.4.

The shapes Star, Plus, and Y-Shape are consistently well-classified across almost all radii and feature types. This is most likely due to their distinctive hub-and-spoke topologies, which are reflected both in distance and structural features. On the other hand, Binary Tree and Hourglass exhibit consistent cross confusion, especially at SHORT and MEDIUM radii, as their tree-like sequences produce similar local neighborhoods most directly evident under sparse connectivity.

We once again see a complete collapse in accuracy at LARGE radius for 5D structural features. It is once again evident that the features provide no information outside of the number of robots in the global swarm, as shapes Triangle and Square (the only shapes with n=3 and n=4 total robots respectively) are classified with 100% accuracy. All misclassified shapes are only misclassified within the subgroup of shapes sharing their total number of robots.

6.4 Ring Shape Analysis

Four of the ten total shapes share a pure ring topology, in which each robot has exactly 2 neighbors at SHORT radius. This topology eventually breaks at larger communication radii, but an analysis of the feature differences at smaller communication radii may differentiate the ability to encode information between pure scalar distance information and structural densities, as well as the potential limitations of both approaches.



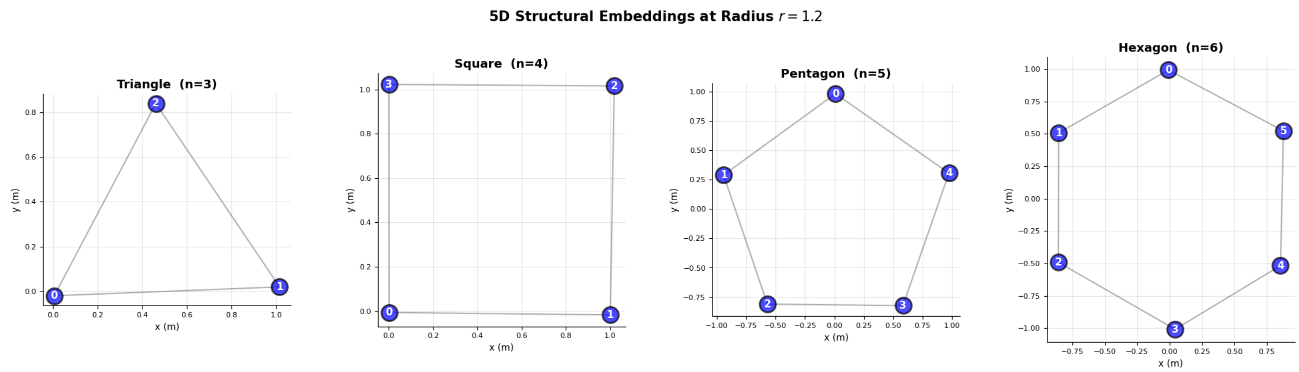
Robot	mean	min	max	std	deg
0	0.739	0.711	0.766	0.027	2
1	0.753	0.74	0.766	0.013	2
2	0.726	0.711	0.74	0.014	2
—	—	—	—	—	—

Robot	mean	min	max	std	deg
0	1.118	1.096	1.14	0.022	2
1	1.113	1.096	1.13	0.017	2
2	1.135	1.13	1.14	0.005	2
—	—	—	—	—	—

Robot	mean	min	max	std	deg
0	1.089	1.065	1.114	0.024	2
1	1.105	1.097	1.114	0.008	2
2	1.091	1.085	1.097	0.006	2
3	1.075	1.065	1.085	0.01	2

Figure 6.4: Differences in 5D Distance initial embeddings between two Triangles at varying scales and a Square, all at $r = 1.2$.

Figure 6.4 depicts the differences in initial 5D Distance embeddings between a Triangle at two different scales and a Square. Each robot is limited to the distance information of its two neighbors and, due to the uniformity of ring-like structures, is unable to gain helpful information even through multiple rounds of aggregation. We further observe the limitations the scale parameter on feature differentiation. A Triangle at a certain scale will have feature embeddings more similar to a differing ring-like shape at the same scale than a separate Triangle at smaller or larger scale. This is not a failure of the feature computation but a fundamental consequence of the ring topology under sparse communication.



Robot	δ	λ	γ	ρ	η
0	2	0	1	0	0
1	2	0	1	0	0
2	2	0	1	0	0
—	—	—	—	—	—
—	—	—	—	—	—
—	—	—	—	—	—

Robot	δ	λ	γ	ρ	η
0	2	1	0	1	0
1	2	1	0	1	0
2	2	1	0	1	0
3	2	1	0	1	0
—	—	—	—	—	—
—	—	—	—	—	—

Robot	δ	λ	γ	ρ	η
0	2	1	0	2	0
1	2	1	0	2	0
2	2	1	0	2	0
3	2	1	0	2	0
4	2	1	0	2	0
—	—	—	—	—	—

Robot	δ	λ	γ	ρ	η
0	2	1	0	2	0
1	2	1	0	2	0
2	2	1	0	2	0
3	2	1	0	2	0
4	2	1	0	2	0
5	2	1	0	2	0

Figure 6.5: Differences in 5D Structural initial embeddings between Triangle, Square, Pentagon, and Hexagon at $r = 1.2$.

Figure 6.5 depicts the differences in initial 5D Structural embeddings between the four ring-like shapes. We observe the embeddings for all robots in shape Triangle share density $\lambda = 0$ and $\gamma = 1$, as all robots in this shape are connected. This connectivity therefore differentiates the initial embeddings from later ring structures. Square, Pentagon, and Hexagon, however, each produce a ring with structure $\delta = 2$, $\lambda = 1$, $\gamma = 1$, as outside of its two direct neighbors each robot remains unconnected from the graph outside of its communication radius. Robots within a Square differentiate from a Pentagon or a Hexagon with $\rho = 1$ rather than 2, as there is only one remaining robot outside of their neighbor list. These two differences allow both 5D structural and 5D combined embeddings to recognize shapes Triangle and Square from the remaining ring shapes using only a one-hop message passing system.

It is worth noting, however, that shapes Pentagon and Hexagon share the exact same initial embeddings and are therefore consistently confused at SHORT radius. In fact, every ring-like shape following $n = 4$ robots will share this same confusion. To adapt to this issue the specific 5D structural features could be altered to better suit the issue of ring topology. This highlights a broader limitation of structural features: their discriminative power is inherently tied to prior knowledge of the formation vocabulary.

6.5 Discussion

The results reveal a consistent and interpretable interaction between communication radius and feature type. Structural features are best suited to identify a robot's role within its local formation and therefore recognize shapes in which robots contribute various perspectives, while distance features capture the physical scale and spread of the neighborhood. At LARGE radius we observe distance features alone to offer the greatest ability to distinguish robot formation, but this requires a highly connected environment not offered in many practical swarm settings. Many distance features are incredibly informative when given access to a full communication graph, such as minimum and maximum neighbor distance, but at sparse radii a robot may have only one or two neighbors, rendering these statistics insufficient to characterize the broader formation. Structural features fill this gap precisely because they are designed to extract meaning from local topology regardless of scale.

The three specific design choices of this paper – local feature design, gossip-based pooling, and per-robot classification – address the same underlying constraint. The lack of global context requires that every stage of the pipeline operate exclusively on information a robot can obtain through bounded local communication. Prior distributed approaches either require global localization to construct meaningful node features or rely on centralized aggregation to produce a classification. Our work addresses both simultaneously, demonstrating that formation self-awareness can emerge entirely from peer-to-peer communication among robots with no global knowledge at any stage of inference.

Chapter 7: Conclusion & Future Work

This thesis addressed the problem of distributed swarm formation classification under strict local communication constraints with no access to a global positioning system or base station. We formalized the swarm as a communication graph and proposed a four-stage distributed pipeline: local feature computation, shared-weight local GNN message passing, and gossip averaging as a substitute for global pooling, terminating in a per-robot classifier that produces an independent shape prediction on every robot simultaneously.

We designed and evaluated three local feature representations: the 5D distance vector encoding scalar statistics of each robot's one-hop neighborhood, the 5D structural vector encoding graphlet-inspired topological counts over the same one-hop neighborhood, and the 10D combined concatenated vector of both representations. All three vectors are computable with no global graph access, entirely from each robots neighbor list and the neighbor lists of those same robots.

We demonstrated that structural and combined features consistently outperform distance features at SHORT radius, while distance features dominate at LARGE radius where the communication graph becomes complete. We additionally observed the 10D combined vector to achieve the strongest overall performance at MEDIUM radius, the radius most similar to a communication graph at a real world deployment level. We further showed the failure modes of distance and structural features to be distinct, and that their combined vector remained competitive across all radii regardless of individual embedding collapse due to an ability to draw upon the stronger signal.

7.1 Limitations

Several limitations constrain the generality of the current system. First, the evaluation is restricted to static formations of fixed size, in which robots are placed at set positions and then subject to small noise and scale variation. The system has not been evaluated on a dynamically evolving formation nor formations undergoing continuous reconfiguration. Second, the Pentagon and Hexagon confusion described in Section 6.4 highlights a theoretical limitation of structural features who requires an understanding of the predefined shape classes in their design. The successes of structural features may be attributed to their design specifically suited for the problem at hand rather than as learned-heuristics. Fourth, all experiments were conducted in simulation under an idealized communication assumption, in which no packet loss or hardware noise was present. It is additionally unclear how the simulation would scale to formation classes ranging to larger swarms.

7.2 Future Work

Several directions present themselves as natural extensions of this work. The most immediate is hardware deployment on physical Kilobot systems, which would introduce realistic sources of noise including range estimation error, message collision, and timing variability, and would provide a direct test of whether the trained models transfer to the physical domain without retraining. A second direction is extension to dynamic formations, in which robots continuously update their shape classification as the swarm reconfigures, requiring either online adaptation of the GNN weights or a recurrent formulation that tracks formation identity through time.

The ring shape collapse analysis motivates a third direction of learned-heuristics. The structural feature vectors are currently designed on predefined shape classes to provide additional structural information. However, future work could explore a graphlet design process learned through deployment. This would allow the model to discover structural features beyond the fixed vocabulary, potentially resolving confusions that the current feature set cannot address.

Finally, scaling the evaluation to larger swarms and broader formation vocabularies would provide a more complete picture of the practical limits of distributed formation recognition as proposed in our algorithm.

Bibliography

[1] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2012, pp. 3293–3298.

[2] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[3] M. Gauci, R. Nagpal, and M. Rubenstein, "Programmable self-disassembly for shape formation in large-scale robot collectives," in *Proc. Int. Symp. Distributed Autonomous Robotic Systems (DARS)*, 2018, pp. 573–586.

[4] D. March-Pons, J. Múgica, E. E. Ferrero, and M. C. Miguel, "Honeybee-like collective decision making in a kilobot swarm," *Physical Review Research*, vol. 6, no. 3, 2024.

[5] I. Slavkov et al., "Morphogenesis in robot swarms," *Science Robotics*, vol. 3, no. 25, p. eaau9178, 2018.

[6] M. S. Talamali, A. Saha, J. A. R. Marshall, and A. Reina, "When less is more: Robot swarms adapt better to changes with constrained communication," *Science Robotics*, vol. 6, no. 56, p. eabf1416, 2021.

[7] M. Otte, "An emergent group mind across a swarm of robots: Collective cognition and distributed sensing via a shared wireless neural network," *The International Journal of Robotics Research*, vol. 37, 2018.

[8] M. Pluhacek, S. Garnier, and A. Reina, "Decentralised construction of a global coordinate system in a large swarm of minimalistic robots," *Swarm Intelligence*, vol. 19, pp. 333–360, 2025.

[9] J. P. Desai, "A graph theoretic approach for modeling mobile robot team formations," *Journal of Robotic Systems*, vol. 19, pp. 511–525, 2002.

[10] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, 2007.

[11] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Information Processing in Sensor Networks (IPSN)*, 2005, pp. 63–70.

[12] Z. Zhang, X. Zhao, B. Tao, and H. Ding, "Distributed gossip-triggered control for robot swarms with limited communication range," *IEEE Transactions on Industrial Electronics*, 2023, doi: 10.1109/TIE.2023.3239876.

[13] H. Cao, S. Shreedharan, and N. Atanasov, "Multi-robot object SLAM using distributed variational inference," *arXiv preprint arXiv:2404.18331*, 2024.

[14] S. Chinchali et al., "Network offloading policies for cloud robotics: A learning-based approach," *arXiv preprint arXiv:1902.05703*, 2019.

[15] A. J. Genot et al., "Hearing the shape of an arena with spectral swarm robotics," *arXiv preprint arXiv:2403.17147*, 2024.

[16] J. T. Kim, J. Park, S. Choi, and S. Ha, "Learning robot structure and motion embeddings using graph neural networks," *arXiv preprint arXiv:2109.07543*, 2021.

[17] R. Miao, Q. Jia, F. Sun, G. Chen, H. Huang, and S. Miao, "Semantic representation of robot manipulation with knowledge graph," *Entropy*, vol. 25, no. 4, p. 657, 2023.

[18] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Proc. Conf. Robot Learning (CoRL)*, 2020, pp. 671–682.

[19] W. Gosrich, S. Mayya, R. Li, J. Paulos, M. Yim, A. Ribeiro, and V. Kumar, "Coverage control in multi-robot systems via graph neural networks," *arXiv preprint arXiv:2109.15278*, 2021.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

[21] L. Ruiz, F. Gama, and A. Ribeiro, "Graph neural networks: Architectures, stability and transferability," *arXiv preprint arXiv:2008.01767*, 2021.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2018.

[23] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, 2019, pp. 6861–6871.

[24] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, 2018, pp. 5453–5462.

[25] A. Zhao et al., "RoboGrammar: Graph grammar for terrain-optimized robot design," *ACM Transactions on Graphics*, vol. 39, pp. 1–16, 2020.

[26] L. Steels, "The emergence of grammar in communicating autonomous robotic agents," in *Proc. 14th European Conf. Artificial Intelligence (ECAI)*, 2000, pp. 764–769.

[27] G. Stiny, "Introduction to shape and shape grammars," *Environment and Planning B: Planning and Design*, vol. 7, pp. 343–351, 1980.

[28] N. Pržulj, D. G. Corneil, and I. Jurisica, "Modeling interactome: Scale-free or geometric?" *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, 2004.

[29] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.

[30] N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Proc. 12th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2009, pp. 488–495.

[31] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein, “Improving graph neural network expressivity via subgraph isomorphism counting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 657–668, 2023.