Emergent Molecular Communication: Preliminary Results with Graph Neural Networks and Diffusion Channels

Alberto Archetti[1], Gabriele Giusti[1], Karthik Reddy Gorla[1,3], Stefano Caputo[2], Maurizio Magarini[1], Matteo Matteucci[1], Lorenzo Mucchi[2], Massimiliano Pierobon[3]

***Abstract*—Biological organisms have developed sophisticated communication mechanisms to enhance their adaptability and survival, ranging from molecular signaling to complex linguistic structures. Understanding the evolution of these mechanisms can provide insights into both biological processes and artificial communication system design. As a first attempt in this direction, a preliminary system based on Graph Neural Networks (GNNs) and molecular diffusion models is presented in this paper to observe the emergence of Molecular Communication (MC) protocols among artificial agents. In particular, these agents interact according to a Lewis-game framework by exchanging molecules that diffuse in a fluidic environment. In this framework, a sender encodes and transmits symbols to a destination with the help of intermediate relay nodes, subject to the stochastic behavior of diffusion-based MC, which affects message integrity. Each of these nodes can tune its internal parameters to influence communication, mimicking the evolution of biological organismssuch as bacteria. Preliminary numerical results suggest that this system can learn to evolve an effective MC protocol. The findings highlight the potential of such a system for investigating emergent MC, possibly leading to a better understanding of molecular biological systems and the optimization of bio-based and bio-inspired networked systems.**

***Index Terms*—Lewis game, graph neural networks, emergent communication, diffusion-based communication.**

I. Introduction

The emergence of communication in biological systems is a fundamental phenomenon that has shaped the evolution of life. Organisms develop intricate signaling mechanisms, from chemical diffusion at the microscopic scale to complex linguistic structures, enabling adaptability to dynamic environments. Understanding these communication processes provides insights not only into biological evolution but also into artificial communication system design. While bio-inspired approaches have significantly influenced areas such as neural networks and artificial intelligence, their potential for guiding adaptive communication in engineered systems remains underexplored [1].

One of the key challenges in multi-agent communication is adaptability in uncertain and evolving environments. Handcrafted protocols often struggle when real-world conditions deviate from predefined constraints. In biological systems, organisms modify their communication strategies in response to environmental changes, ensuring robustness and survival [2]. For instance, chemotactic bacteria adjust their signaling based on environmental cues, optimizing information flow for survival [3]. Inspired by these principles, this paper explores the emergence of Molecular Communication (MC) protocols in artificial agents using a novel framework based on Graph Neural Networks (GNNs), a communication channel modeling molecular diffusion, and the Lewis signaling game.

The proposed approach models communication as an iterative message-passing process over a graph, where nodes exchange molecular signals that diffuse through a fluidic medium. The Lewis signaling game is applied to study how communication systems evolve and how signals acquire meaning [4]. Unlike prior work that applies the Lewis game to semantic communication in engineered systems [4], this study extends it to a biologically inspired broadcast setting based on the Brownian motion of information-bearing molecules. Agents dynamically adjust their internal parameters, mimicking evolving biological entities. The key contributions of this work include: (i) A novel adaptation of the Lewis game to model multi-node broadcast of MC messages via molecule diffusion; (ii) The design and development of a diffusion-based stochastic communication channel compliant with backpropagation. While the former results in a framework that can be extended to other broadcast transmission scenarios, even beyond MC systems, the latter ensures the differentiability of the network model, essential for successfully supporting a Lewis game.

This study represents the first step within a broader projects aimed at understanding the conditions that drive evolution towards the emergence of communication strategies. Preliminary results indicate that this approach enables the emergence of effective MC strategies, shedding light on how adaptive biological communication may evolve. These findings have implications for understanding natural MC processes and optimizing bio-inspired communication networks for applications in synthetic biology, bio-nanotechnology, and future biohybrid intelligent systems.

The rest of the paper is organized as follows. Section II introduces some background necessary to understand the proposed framework. In Sec. III we describe the assumptions and models underlying the proposed computational framework. In Sec. IV, we describe the experimental setup and discuss the obtained preliminary numerical results. Finally, in Sec. VI we conclude the paper.

II. Background

A. Emergent Communication in Artificial Intelligence

Emergent Communication (EC) explores how communication protocols naturally arise as artificial agents evolve

[1]Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Milan, Italy

[2]Department of Information Engineering, Università degli Studi di Firenze, Florence, Italy

[3]School of Computing, University of Nebraska-Lincoln, Lincoln, NE, USA

their interactions to solve a task, aiming to understand the underlying principles that drive the emergence of artificial language [5]. Alongside emergence itself, EC examines the properties of these languages and how they relate to constraints imposed by the communication environment or channel [6]. The foundational framework for EC is the Lewis Signaling Game [7]. In this setup, two or more agents collaborate to solve a task, such as instance classification, by using a communication channel. A standard Lewis game involves two agents, a sender and a receiver. The sender observes an object and sends a message to the receiver, who must then select the correct object from a set of candidates based on the received message. Positive rewards are given only for correct classifications. The key aspect of the Lewis games is that the communication protocol is not predefined; it must be learned by the agents during the experiment using optimization algorithms [8], [9].

B. Graph Neural Networks

GNNs [10] are a class of deep learning models designed to process graph-structured data. The fundamental operation in GNNs is graph convolution, which iteratively aggregates information from neighboring nodes and generates messages that are propagated throughout the graph. This mechanism enables GNNs to effectively leverage neighboring information, exploiting the relationships between adjacent nodes. GNNs have been successfully employed in various applications, including social network analysis, molecular chemistry, and recommendation systems [11].

C. Molecular Communication

MC is an interdisciplinary research field that explores the transmission and processing of information using molecules as carriers, inspired by biological communication systems at the cellular and nanoscale levels [12]. One of the most studied paradigms within MC is based on molecule diffusion, where information is encoded in the amount, type, or release timing of molecules that propagate through a fluid medium via Brownian motion. This mode of communication is stochastic in nature, with signal arrival and detection governed by diffusion equations and first-passage time statistics [13]. Diffusion-based MC is crucial for understanding and engineering synthetic biological networks, targeted drug delivery, and the Internet of Bio-Nano Things (IoBNT) [14], where molecules act as information carriers to coordinate functions at the microscopic scale.

III. Network Architecture

We propose to address a Lewis game by integrating key aspects of MC via diffusion. For this, a preliminary investigation is presented in this paper based on a neural network architecture. This architecture extends traditional GNNs to capture the inherent randomness of diffusion channels while maintaining the end-to-end differentiability necessary for its training process.

A. Assumptions

Our network architecture, with reference to Fig. 1, is based on the following assumptions: a set of nodes distributed in the space, and indexed as $i = 0, \dots, N-1$, form a fully connected graph; nodes are static, i.e., they do not change position at training or inference time; each node can serve as both point-wise transmitter and receiver of an MC message; all interactions between nodes occur in discrete time intervals indexed by τ; each MC message vector $\mathbf{s}_i^\tau$ sent by node i (transmitter) at the beginning of each time interval τ is composed of K instantaneous emissions of molecules of different species; each node of the network is capable of sensing (receiving) an incoming quantity of molecules distinctly for each of the K species (K independent channels) and they are perfect absorbers of the received molecules; a message $\mathbf{s}_i^\tau$ sent by a node i is received by node j ($i \neq j$) within the same time interval τ according to a distance-dependent stochastic weight w_{ij}^τ, as detailed in Sec. III-B2; at the beginning of each time interval τ, all molecules emitted by nodes at the previous time interval have degraded and do not interfere with the next emissions. All nodes are assumed synchronized. As our work progresses, we plan to revise and remove some of these preliminary assumptions.

B. Communication Model

1) Diffusion-based detection probability: Consider two nodes i and j at a distance d_{ij}. When i emits molecules, each diffuses according to Brownian motion through the medium. The probability density function of the first-passage time $f(t)$ for one molecule to reach node j for the first time [12] is

$$f(t) = \frac{d_{ij}}{\sqrt{4\pi D\, t^3}} \exp\left(-\frac{d_{ij}^2}{4D\, t}\right), \tag{1}$$

where D is the diffusion coefficient and t is the time elapsed from the molecule emission. Thus, the probability p_{ij} that a molecule reaches node j within time t is

$$p_{ij} := P\{T \leq t\} = \operatorname{erfc}\left(\frac{d_{ij}}{\sqrt{4D\, t}}\right), \tag{2}$$

where $\operatorname{erfc}(\cdot)$ is the complementary error function [12], [13]. For simplicity, in the scope of this study we assume that all molecule species are characterized by the same diffusion coefficient $D = 1$ and that molecule emission and receptor at each time interval τ happens for a time duration of $t = 1$. As a result of these assumptions, $p_{ij} = \operatorname{erfc}(d_{ij}/2)$.

2) Stochastic molecular detection: The detection of multiple molecules of each of the K species received at node j and emitted by node i can be modeled as a binomial distribution with number of trials corresponding to the number of emitted molecules and probability of success equal to p_{ij}. Assuming a large number of emitted molecules, the random variable $\mathbf{r}_{ij}^\tau$ denoting the number of molecules detected by j upon an emission s_i^τ of K species from i at time interval τ can be approximated by a multivariate Gaussian distribution as

$$\mathbf{r}_{ij}^\tau \sim \mathcal{N}\Big(p_{ij}\, \mathbf{s}_i^\tau,\, \operatorname{diag}\big(p_{ij}\,(1-p_{ij})\, \mathbf{s}_i^\tau\big)\Big). \tag{3}$$

For multiple senders, the total number of molecules detected by node j for each of the K species is

$$\mathbf{r}_j^\tau = \sum_{\substack{i\in\{1,\dots,N\},\\ i\neq j}} \mathbf{r}_{ij}^\tau. \tag{4}$$

To enable backpropagation through stochastic samples, we utilize the reparametrization trick from variational autoencoders [15]. This technique reformulates stochastic variables in a way that separates the source of randomness from the parameters to be optimized, allowing gradients to propagate though the network. Specifically, a Gaussian-distributed variable can be rewritten as the sum of its mean and the product of its standard deviation with a noise term sampled from a fixed distribution. Let $\varepsilon_{ij}^\tau \sim \mathcal{N}(0,1)$, then

$$\mathbf{r}_j^\tau = \sum_{\substack{i\in\{1,\dots,N\}\\ i\neq j}} \Big(p_{ij}\,\mathbf{s}_i^\tau + p_{ij}(1-p_{ij})\,\varepsilon_{ij}^\tau\,\mathbf{s}_i^\tau\Big) = \sum_{\substack{i\in\{1,\dots,N\}\\ i\neq j}} w_{ij}^\tau\,\mathbf{s}_i^\tau, \tag{5}$$

where w_{ij}^τ is the distance-dependent stochastic weight that controls the number of molecules detected by j.

3) Message-passing interpretation: As depicted in Fig. 1, at iteration τ, node j receives $\mathbf{r}_j^{\tau-1}$ molecules of each of the K species. Then, a neural network ϕ_j maps these detected amounts of molecules to the emissions at the next iteration τ as

$$\mathbf{s}_j^\tau = \phi_j\big(\mathbf{r}_j^{\tau-1}\big). \tag{6}$$

According to our assumptions, each node j broadcasts at time interval τ the same message with the emission $\mathbf{s}_j^\tau$ to all other nodes. This construction is equivalent to a GNN message passing step, where the update of each node depends on aggregating messages from all neighbors. The distance-dependent stochastic weight w_{ij} in (5) acts as a weighted edge in the GNN, modulating the probability of successful communication.

4) Lewis game integration: To simulate a Lewis game, we add an encoder $\mathcal{E}$ to map input features $\mathbf{x}$ at every time interval τ into a K-dimensional message vector $\mathbf{s}^0$, as shown in Fig. 1. We designate two nodes, a sender and a receiver, as the pair furthest apart in the fully connected graph. Initially, the sender is assigned message $\mathbf{s}^0$, while all other nodes start with no messages to be transmitted. We then perform T message-passing iterations of the above molecular communication process. Finally, a decoder $\mathcal{D}$ outputs class probabilities from the receiver node. The entire system is trained end-to-end using backpropagation and a categorical cross-entropy loss. This way, the neural network parameters are driven to learn a message encoding that maximizes classification performance, effectively solving the Lewis game.

The described GNN-based network architecture enables the integration of communication channels based on MC via diffusion, and the aforementioned communication models make it both stochastic and differentiable, enabling the emergence of a communication protocol as part of a learning task in a Lewis game. In Fig. 2 we show an example of a trained 6-node network resulting from a Lewis game, as described in Sec. IV.

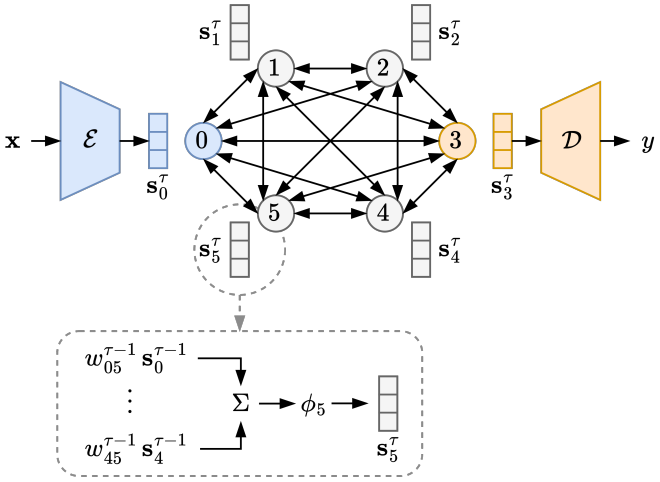

Fig. 1. Example of the proposed architecture on a 6-node network. The input signal $\mathbf{x}$ is encoded into a message $\mathbf{s}_0$ at sender node 0 with dimension $K = 3$, which propagates through the network following stochastic diffusion-based weights. Through multiple rounds of message passing updates, information propagates until reaching receiver node 3, where the final message $\mathbf{s}_5^T$ is decoded into the output classification y.

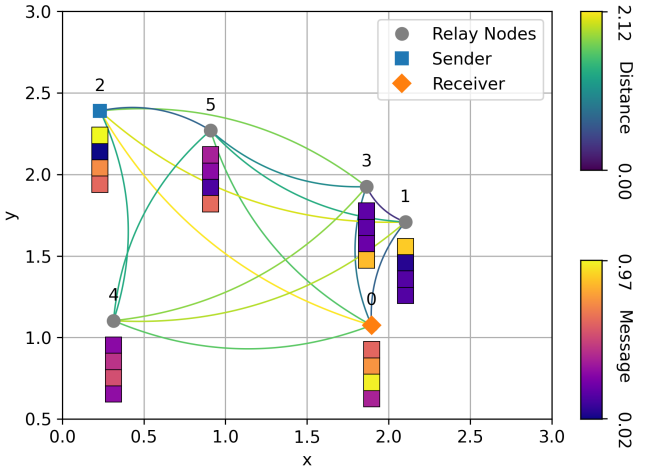

Fig. 2. Example architecture with $N = 6$ and $K = 4$. Node distances are depicted as color-coded arcs. Message contents are drawn below each node with each square corresponding to the k-th entry.

C. Computational Complexity

To characterize the computational complexity of the proposed solution, we analyze its scaling behavior with respect to the input parameters. We begin by assuming a fully connected graph of N nodes, where each node exchanges a K-dimensional vector with every other node. The message-passing operation thereby amounts to $\mathcal{O}\big(N^2K\big)$ in total, since each of the $\mathcal{O}\big(N^2\big)$ edges involves an $\mathcal{O}\big(K\big)$ operation. Next, each node updates its internal state via a small neural network with H hidden units. The forward pass of such a network can be approximated by $\mathcal{O}\big(KH\big)$. Because there are N nodes, this amounts to an additional $\mathcal{O}\big(NKH\big)$ per message-passing round. Since the message passing and updating procedures repeat for T iterations, the total computational complexity is

$$\mathcal{O}\big(T\,\big(N^2K + NKH\big)\big). \tag{7}$$

In typical settings where K and H are small with respect to N and T, the dominant cost arises from the quadratic term N^2. The overall scaling is therefore largely determined by the number of nodes N and the number of iterations T.

IV. Numerical Results

A. Data and Parameters

We assess classification performance of our model on three widely used tabular datasets from the UCI Machine Learning Repository [16]: Breast Cancer (569 samples, 30 features, 2 classes), Iris (150 samples, 4 features, 3 classes), and Wine (178 samples, 13 features, 3 classes). For each dataset, we allocate 70% of the samples for training, 10% for validation, and 20% for testing.

In every experiment, communication nodes are placed randomly in a two-dimensional area at a uniform density of one node per unit space and remain fixed during training. While the density is constant, we vary the number of nodes N and the number of message-passing iterations T across experiments. The number of independent communication channels K is set to four.

Our trainable architecture comprises encoders, decoders, and message-transformation functions implemented as feedforward neural networks. Each network has three fully connected layers—a data input layer, a hidden layer with four neurons, and an output layer—each followed by a ReLU activation. The encoders and decoders are pretrained on their respective datasets and their weights are frozen during the molecular communication training phase; allowing these weights to remain trainable led to similar performance. We train all networks for 2000 epochs using the Adam optimizer [17], initializing the learning rate at 0.01 and using a batch size of 256. After two-thirds of the training, we reduce the learning rate by a factor of ten. The final model parameters correspond to the epoch with the lowest validation loss. Training and evaluation are carried out using the `PyTorch`[1] library, and the fully reproducible code is available on GitHub[2].

B. Results

Our empirical findings support the feasibility of using a diffusion-based communication model within a GNN framework to solve a Lewis game. Here, we summarize the main observations from our experiments, emphasizing that the reported results serve as a proof of concept and are not intended to be conclusive.

1) Results Overview: Figure 3 collects accuracy results under ideal and diffusion channels. The left column represents an upper performance bound, where nodes receive perfect message copies across a noise-free channel. The right column illustrates results using the proposed noisy diffusion channel. Rows correspond to the three datasets. In each plot, the x-axis denotes the number of nodes N and the y-axis shows classification accuracy at the final iteration. The black dashed line marks the encoder–decoder baseline (no relay nodes),

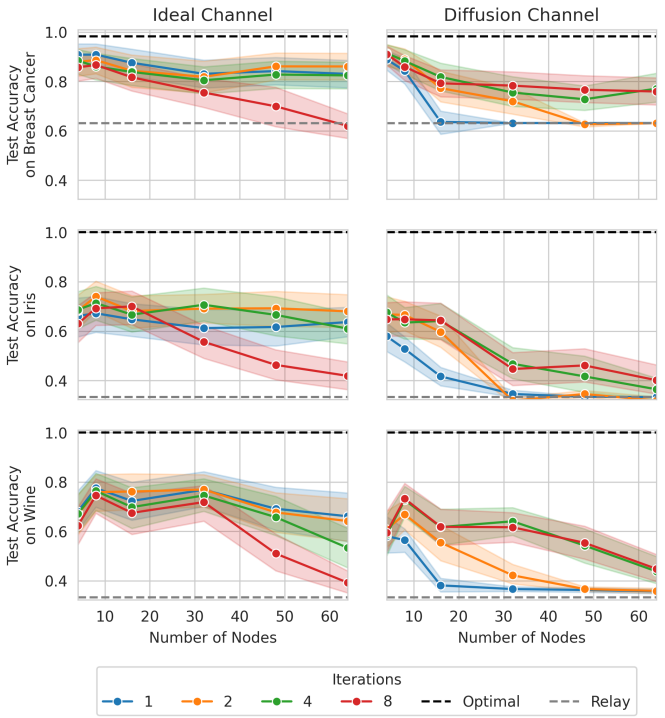

Fig. 3. Test accuracy across all configurations. Left column shows results with a noise-free ideal channel, and right column with the proposed diffusion channel. Each row corresponds to one of the three datasets (Breast Cancer, Iris, Wine), and each curve plots accuracy versus number of nodes N, averaged over 30 runs.

while the gray line indicates non-trainable relay nodes that simply forward incoming messages.

2) Effect of Node Count and Iterations: Focusing on the diffusion channel results, two trends emerge. First, performance generally degrade with an increasing number of nodes. This is expected, as the classification problem becomes more difficult with more nodes training and communicating, as each of these nodes must learn how to send meaningful messages to its neighbors. The second observation, however, is that the degree of degradation is much lower for experiments with a higher number of iterations—4 to 8. This finding suggests that, as the network of communicating nodes grows larger, it necessitates several communication rounds to provide meaningful classification information to the receiver. This highlighting that a communication scheme has emerged from training the GNN.

3) Ideal vs. Diffusion Channel: Comparing each dataset (row-wise), we observe that, although diffusion channels underperform relative to the ideal case, the gap remains small. This suggests our model approaches the upper-bound performance even in the presence of realistic diffusion noise. Interestingly, under the ideal channel, accuracy decreases with more iterations—especially for large networks—since redundant communication adds no benefit in the absence of noise and contributes only to information degradation. Conversely, with diffusion noise, additional iterations improve accuracy for large networks. This implies that the model is capable of learning robust messaging schemes to cope with diffusion

[1]https://pytorch.org/
[2]https://github.com/archettialberto/commence-tmbc

noise.

4) Overall Performance: Across all configurations, our models consistently perform between the relay-only baseline and the encoder–decoder upper bound. It is only in scenarios combining many nodes with few iterations under the diffusion channel that accuracy approaches random guessing. However, this is expected, as diffusion degrades useful information between senders and receivers far from each other. These results highlight both the challenges and the promise of training GNNs with diffusion-based communication.

5) Ablation Study: In addition to the core architecture, we evaluated several modifications that produced negligible performance gains and are omitted from the main discussion. Specifically, we varied network depth (0–3 hidden layers) and width (4–32 neurons per layer). We also tested skip connections, batch normalization, dropout, and sigmoid activations in place of ReLU. Furthermore, we introduced a simple memory mechanism at each node by concatenating the previous output message with the current input and tested several values for the number of channels K. Finally, we experimented with training the encoder and decoder weights alongside the network. None of these techniques significantly affected the results.

V. Discussion and Future Work

The results presented in this work serve as an initial exploration within a broader research aimed at understanding, computationally characterizing, and leveraging the evolutionary mechanisms underlying emergent communication. Using the proposed GNN-based framework as a foundation, several potential enhancements and extensions can be investigated. First, more extensive experiments are needed to fully assess the strengths and limitations of the approach, particularly under different parameter settings and network topologies. These additional investigations will provide a clearer picture of the model robustness and its adaptability to various molecular communication environments.

Beyond refining the current results, a key extension involves incorporating more biologically realistic cellular behaviors. For instance, introducing agent mobility and energy constraints would allow a more dynamic simulation of molecular communication systems. Specifically, neural networks could generate additional outputs representing movement actions, enabling nodes to reposition based on learned strategies. Since such movement dynamics can introduce non-differentiable operations, future iterations may need to incorporate reinforcement learning or genetic programming techniques, which can accommodate discrete decision-making but may also increase convergence complexity.

Another interesting addition is the inclusion of an energy management mechanism. By associating an energy cost with molecular transmissions and action, evolutionary pressure can be simulated by penalizing inefficient communication strategies. A possible implementation could involve periodically resetting the weights of nodes with lower classification accuracy, thereby simulating survival-of-the-fittest dynamics.

Ultimately, these enhancements will contribute to a deeper understanding of how learning-based protocols can drive bio-inspired emergent communication. The benefits are twofold: first, understanding the evolutionary pressures and conditions that enable the emergence of communication in natural environments, and second, applying these insights to the design of more efficient, self-organizing, and adaptive communication protocols for engineering applications.

VI. Conclusion

This paper presented a novel approach for studying the emergence of diffusion-based MC protocols in artificial agents using a GNN-based framework and the Lewis signaling game. Our preliminary findings suggest that artificial agents can learn to develop effective MC protocols, mirroring the adaptive communication strategies observed in biological systems. The implications of this work extend beyond theoretical exploration, providing a foundation for designing robust bio-based and bio-inspired communication networks.

Acknowledgments

This paper is supported in part by the FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, investment 1.3, line on Artificial Intelligence) and in part by the Italian Ministry of Foreign Affairs and International Cooperation, grant number US23GR04 (CUP: D43C23000350001).

References

[1] A. K. Kar, Bio inspired computing – a review of algorithms and scope of applications, Expert Systems with Applications 59 (2016) 20–32.

[2] Z. Chelly Dagdia, P. Avdeyev, M. S. Bayzid, Biological computation and computational biology: survey, challenges, and discussion, Artificial Intelligence Review 54 (6) (2021) 4169–4235.

[3] L. Brand, Y. Wang, M. Magarini, R. Schober, S. Lotter, Semantic information in mc: Chemotaxis beyond shannon, To appear in Proc. GLOBECOM 2024. Preprint arXiv:2402.18465 (2024).

[4] J. Choi, J. Park, Semantic communication as a signaling game with correlated knowledge bases, in: VTC2022-Fall, IEEE, 2022, pp. 1–5.

[5] B. Boldt, D. Mortensen, A review of the applications of deep learning-based emergent communication, Preprint arXiv:2407.03302 (2024).

[6] B. Carmeli, R. Meir, Y. Belinkov, Emergent Quantized Communication, Proc. AAAI Conference on Artificial Intelligence 37 (10) (2023) 11533–11541. doi:10.1609/aaai.v37i10.26363.

[7] D. Lewis, Convention: A philosophical study, John Wiley & Sons, 2008.

[8] R. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (2004) 229–256.

[9] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144 (2016).

[10] B. Sanchez-Lengeling, E. Reif, A. Pearce, A. B. Wiltschko, A gentle introduction to graph neural networks, DistillHttps://distill.pub/2021/gnn-intro (2021).

[11] J. Zhou, et al., Graph neural networks: a review of methods and applications, AI Open 1 (2020) 57–81.

[12] N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, W. Guo, A comprehensive survey of recent advancements in molecular communication, IEEE Communications Surveys & Tutorials 18 (3) (2016) 1887–1919. doi:10.1109/COMST.2016.2527741.

[13] M. Pierobon, I. F. Akyildiz, A statistical–physical model of interference in diffusion-based molecular nanonetworks, IEEE Trans. Commun. 62 (6) (2014) 2085–2095. doi:10.1109/TCOMM.2014.2314650.

[14] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, Y. Koucheryavy, The internet of bio-nano things, IEEE Communications Magazine 53 (3) (2015) 32–40. doi:10.1109/MCOM.2015.7060516.

[15] C. Doersch, Tutorial on variational autoencoders, arXiv preprint arXiv:1606.05908 (2016).

[16] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository (2017).
URL http://archive.ics.uci.edu/ml

[17] D. P. Kingma, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).