# DESIGNING NEURAL NETWORK COLLECTIVES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Artificial neural networks have demonstrated exemplary learning capabilities in a wide range of tasks, including computer vision, natural language processing and, most recently, graph-based learning. Many of the advances in deep learning have been made possible by the large design-space for neural network architectures. We believe that this diversity in architectures may lead to novel and emergent learning capabilities, especially when architectures are connected into a collective system. In this work, we outline a form of *neural network collectives* (NNC), motivated by recent work in the field of collective intelligence, and give details about the specific sub-components that an NNC may have.

## 1 INTRODUCTION

Deep learning methods, characterised by artificial neural networks (ANNs), have demonstrated high performance at many complex tasks including computer vision, natural language processing, and graph-based learning Voulodimos et al. (2018); Cambria & White (2014); Zhou et al. (2020). The number of ANNs in deployment is expected to increase dramatically as the design of ANNs becomes more accessible due to dedicated software packages, and as computational power of modern processors increases. Due to the importance and societal implications of widespread distributed learning systems, we consider here the engineering question of how to design artificial learning systems that perform robustly, adaptively and in a decentralised manner. We focus on the methods and principles of collective intelligence Moussaid et al. (2009); Malone & Bernstein (2015), outlining learning motifs that we take to be important for designing systems of ANNs, so-called 'neural network collectives' (NNCs).

NNCs are defined by structuring computational ANN units in a collective, with the aim of establishing forms of emergent intelligence. An example schema for NNCs is shown in Figure 1. At the first layer, traditional ANNs architectures comprise individual units. The outputs produced by these units are connected together at the second layer and are allowed to interact and exchange information (such as output values, network weight values or latent space probabilistic belief representations) according to the principles of collective intelligence. These interactions produce emergent behaviours at the third layer. Meta-learning and adaptation occurs as information from higher layers feeds back to optimise lower level structures, for example through modulation of individual network structure and behaviour at layer one, or through restructuring of interactions and information exchange at layer two.

We believe that the implementation of large-scale NNCs will bring several benefits. For example, NNCs would give invaluable insight into the advantages and limitations of large-scale collective learning infrastructures. These could dramatically increase the security of sensitive data used in government and medicine. Discussions about distributed collective intelligence systems are already highly active, especially in the context of distributed ledger technologies Saldamli et al. (2020). NNCs would add an 'artificial learning' dimension to this area, for example, by supplementing crowd-sourced learning initiatives with synthetic learning agents Li et al. (2018); Ducrée et al. (2020).

Additionally, we believe that NNCs will be worth studying as a complex system in their own right, and used for investigating the conditions under which collective intelligence emerges. In this setting, we highlight the possibility of performing active experiments on NNCs, ablating parts of the network to test hypothesis on collective intelligence or causal reasoning, or adding adversarial units to affect convergence, such as introducing zealots.

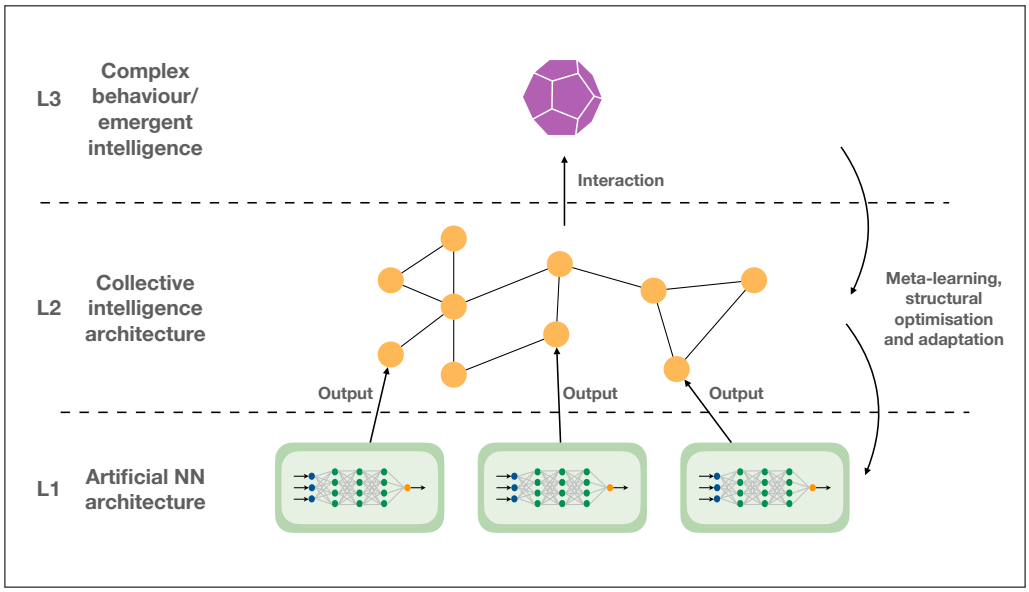


Figure 1: Multi-layered collective artificial intelligence schema

Finally, NNCs may address some of the limitations of current deep learning models. Many of the existing language models are extremely large, training networks with several billion parameters Floridi & Chiriatti (2020). These models are prohibitively expensive to train. NNCs that distribute training over consumer electronics could considerably reduce training costs while still providing strong performance.

This paper is structured as follows; we first outline the principles and concepts of collective intelligence that are relevant to the design of NNCs. We then outline categories of artificial collective intelligence, including multi-agent systems, embodied AI in robotics and various forms of ANNs. Having discussed significant related works, we outline our vision for NNCs, giving details about features at all three layers. We highlight choices to be made at the node level, collective level, and possible features that will emerge at the level of collective intelligence. Finally, we make our concluding remarks.

## 2 Related Work

### 2.1 Collective intelligence and self-organisation

Collective or swarm intelligence is widely observed across scales in nature Camazine et al. (2001). Here, groups of simple, autonomous individuals display the ability to collectively make decisions and solve problems in their environment, such as collective environmental sensing Berdahl et al. (2013), foraging and search Falcón-Cortés et al. (2019); Nakagaki (2001), construction of complex structures Werfel & Nagpal (2006), value-dependent decision making Seeley et al. (2006), as well as task allocation and division of labour Theraulaz et al. (1998). Importantly, completion of these tasks is beyond the capacity of single individuals making up the group, and must be completed in a distributed and self-organised fashion, relying on noisy peer-to-peer communication, and no external guidance. Through specific sets of interactions and information exchange, self-organising systems may then display emergent behaviours and intelligence at the group level Moussaid et al. (2009); Garnier et al. (2007). Natural swarm systems are also robust to environmental noise and are able to adapt their functioning to respond to changes in environmental conditions over time Kitano (2004; 2007). These systems have also demonstrated the ability to collectively learn over time, transferring information between generations during the formation of culture on various time-scales Biro et al. (2016); Sasaki & Biro (2017).

The study and mathematical characterisation of this self-organisation has led to engineering solutions and applications Bonabeau et al. (1999); Dorigo et al. (2021); Bayındır (2016). The motivation

of these algorithms is to reproduce emergent properties of distributed complex systems observed in nature. These properties are robustness to individual unit failure, scalability of operation to large numbers of units (computational load can be distributed across many entities), and flexibility of the type of tasks that systems can perform Şahin (2004); Brambilla et al. (2013). While these principles have been applied to develop artificial collective intelligence in both software Bonabeau et al. (1999); Brabazon et al. (2018); Dorigo et al. (2006); Kennedy & Eberhart (1995) and embodied AI systems Dorigo et al. (2021), their application to systems of agents with higher levels of intelligence has been limited. We propose that the principles of distributed self-organisation can be applied to neural network architectures in order to enhance and extend their range of applications.

#### 2.1.1 Types of structures in CI

Multi-agent systems may be characterised according to the way that data and decisions are shared between the different interacting units. Here, we use the generally agreed-upon definition in the distributed engineering Saeedi et al. (2016) and computation Coulouris et al. (2005); Attiya & Welch (2004); Garg (2002) communities.

Centralised systems are ones in which information is processed by one central, predetermined unit that makes decisions by taking into account global system information. Centralised systems are relatively simple to design, however these structures face operational limitations on computational resources and memory, and are vulnerable to unit failure of the central component.

In decentralised systems, computational load and decision-making are shared across several but not necessarily all units. Information obtained by the different units can then be combined locally in order to generate a more coherent and whole view of the environment. Because such systems contain several separate and autonomous decision-making entities, they typically contain some redundancy, with multiple agents performing the same or similar tasks. This redundancy endows decentralised systems with robustness to environmental noise and individual unit failure Kitano (2004).

Finally, distributed systems are ones in which access to data, computation and decision-making is completely local and asynchronous for each individual unit. Importantly, information is communicated using only local message passing Coulouris et al. (2005), also referred to as gossiping Dimakis et al. (2010). Hence, all distributed systems are decentralised, but not all decentralised systems are distributed. Distributed approaches are especially effective when agents are separated in space or have partitioned access to different data sources, making them scalable and secure for engineering purposes Coulouris et al. (2005); Garg (2002). Collective behaviour and intelligence in biological systems is almost always distributed, as individuals do not share global information and decisions are made autonomously and asynchronously.

#### 2.1.2 Interactions for self-organisation

Collective intelligence is thought to arise from four basic principles Moussaid et al. (2009); Garnier et al. (2007). These are (1) repeated interactions, (2) stochasticity, and (3) positive and (4) negative feedback. Firstly, information must be exchanged between agents in the system through repeated interactions. These interactions provide feedback into the system dynamics which is either positive, creating structure, or negative, preventing structure from forming. Finally, there should be an element of randomness in interactions and agent behaviours. This stochasticity gives self-organising systems flexibility and prevents unstable structures from forming.

Interactions between agents can be further classified as attractive or repulsive, promoting homogeneity or heterogeneity, respectively, across individuals in the group. Consensus is the process of combining local estimates made by separate units in order to reach global agreement. From bees Seeley et al. (2006) to humans Dyer et al. (2008); Momennejad (2022), models and experiments have repeatedly demonstrated the ability of groups to effectively represent and respond to their environment using a variety of consensus forming methods. Several models of consensus have been studied Flache et al. (2017), where system states are represented by discrete and continuous scalar or vector variables Sîrbu et al. (2017); Dong et al. (2018).

The primary goal of consensus algorithms is to make use of a 'wisdom of the crowd' to arrive at a global state estimate based on distributed and noisy local estimates. Agents in distributed systems may have access to the same information source with various degrees of unknown measurement un-

certainty. They may also have access to different samples of data drawn from the same distribution, with the goal of estimating the distribution. Consensus can be centralised (passing through a central processor that synchronously combines all estimates) or distributed, whereby individual agents rely purely on local interactions to propagate information through the system. Centralised consensus eliminates estimation errors in individual measurements by producing a weighted sum of individual estimates. The weighting on each term determines the contribution of the individual estimate to the global estimate. Distributed consensus relies on message passing interactions which are attractive, and lead to an eventual convergence of agent states Dimakis et al. (2010); Flache et al. (2017).

Consensus is also used to perform collective decision making between discrete alternatives, for example in the the best-of-$n$ problem, where a choice must be made between $n$ different alternatives of varying quality. Quorum sensing Sumpter & Pratt (2009); Marshall et al. (2019) is employed by natural systems, and various other probabilistic opinion pooling methods addressing this problem have also been developed Dietrich & List (2016). We posit that similar distributed approaches can be utilised by neural network collectives for tasks such as data classification.

Consensus amongst agents may not always be desirable, however. Instead, agent differentiation may be required in order to organise behaviour and collective action. These methods make use of repulsive positive reinforcement to differentiate between agents. The positive reinforcement means that existing differences between agents are amplified during interactions. This results in agent states being differentiated across a state-space. For example, insects have been shown to use adaptive response thresholds to flexibly perform allocation of tasks and division of labour Bonabeau et al. (1998). Agent differentiation for the purpose of self-organisation has also been studied in the context of social hierarchy formation and organisation Bonabeau et al. (1996), where agent heterogeneity is used for conflict resolution Brush et al. (2018).

### 2.2 Artificial Collective Intelligence

So far, we have discussed some of the key concepts and principles leading to naturally occurring collective intelligence and self-organisation. The mathematical characterisation of these principles has made it possible to design systems displaying artificial collective intelligence, with applications in multi-agent robotics Dorigo et al. (2021); Schranz et al. (2020), distributed sensor technology Zhang & Zhang (2012), and software algorithm design Bonabeau et al. (1999); Brabazon et al. (2018); Dorigo et al. (2006); Kennedy & Eberhart (1995). These applications have been motivated with designing artificial collective systems that share desirable features of self-organising systems in the natural world, including robustness, adaptation, and higher-order emergent structures. Furthermore, these algorithms allow for a distribution of data, computational resources and memory across different processors in the system.

For example, one use-case of artificial collective intelligence has been to enhance the performance of artificial agents in distributed and embodied AI. These include robotic systems and distributed sensor networks. Consensus and distributed filtering methods are used to combine noisy estimates made by individual sensors Yu et al. (2009). Distributed filtering is closely related to consensus formation (see subsubsection 2.1.2). The goal of a filtering task is to estimate the state of a (possibly dynamical) system in the presence of measurement noise. In distributed filtering, data is partitioned across different sensors or computational units. The data is processed separately by the different units, and individual state estimates are refined through local communication. This data integration is usually implemented using Bayesian methods, as they explicitly take the uncertainty of estimates into account. A physically distributed application can be found in multi-agent Simultaneous Localisation and Mapping (SLAM) Thrun & Liu (2005); Cadena et al. (2016); Saeedi et al. (2016), in which agents combine Bayesian models of their physical environment. For further information on distributed state estimation, we refer the reader to Ortiz et al. (2021), as well as Battistelli & Chisci (2014; 2016) for probability density estimate fusion using consensus Kalman Filters.

Self-organised agent differentiation has been applied to problems of task allocation and scheduling in Cicirello & Smith (2004); Castello et al. (2013) by modifying the probability with which agents respond to task stimuli in their environment. More computationally sophisticated swarm intelligence approaches to task scheduling have also been developed Garattoni & Birattari (2018). Additionally, models for self-organising social hierarchies have recently been applied to design spatial deployment

algorithms for multi-robot collective search and coverage tasks in large, unstructured environments Neu & Giuggioli (2022).

Finally, the principles of swarm intelligence have been used to design widely used computational optimisation algorithms, such as *ant colony optimisation* Dorigo et al. (2006) and *particle swarm optimisation* (PSO) Kennedy & Eberhart (1995), amongst others Brabazon et al. (2018). These methods utilise distributed computation methods used by natural swarms to converge to optimal solutions in large search spaces. We note that PSO relies on a form of model ensembling, similar to what we believe may occur in NNCs (see subsubsection 3.2.2).

### 2.3 Applications to Artificial Neural Networks

One of the most popular and powerful paradigms for artificial collective intelligence is now emerging in the field of deep learning. This is the application of artificial neural networks (ANNs) to collective systems. This field is extremely large and developing quickly. As such, we discuss only salient research topics as relevant to NNCs and refer the reader to Ha & Tang (2021) for a comprehensive review.

ANNs take inspiration from one of the most successful examples of collective intelligence, the structure of the brain. The original form of an ANN is known as the multi-layer perceptron (MLP) which is an example of a feed-forward ANN. MLPs typically have three or more hidden layers (where any ANN that includes more than one hidden layer is known as a 'deep' ANN). Data is propagated through the hidden layers, and the output from the hidden layers is used to back-propagate the error and update the model by adjusting the weights in each layer. For further information, see Skansi (2018); Chollet (2021).

Over twenty years ago, there was discussion on how MLPs could be made to interact, such as through the application of statistical physics Metzler et al. (2000); Kinzel et al. (2000); Kanter et al. (2002) or, more recently, in mobile agents Hou et al. (2009); Wen et al. (2016). ANNs have also been designed that interact at the scale of network architecture. One of the most popular examples of this is a generative adversarial network (GAN). In their simplest form, GANs are composed of two ANNs, one 'generative' network which generates new candidate instances of the data and a second 'discriminative' network which evaluates them as either arising from the true or generated datasets. The use of a discriminator allows GANs to avoid the task of comparing the generator distribution to the real data distribution, both of which are difficult to estimate from samples Goodfellow et al. (2014). GANs have been extremely successful at generative tasks, providing some of the earliest examples of 'deep fakes'. Subsequent research has considered many different variations of the GAN architecture Creswell et al. (2018).

One important extension to the GAN architecture has been multi-GANs, which use multiple generators or discriminators. One of the main motivations for the multi-GAN architecture has been to overcome the mode collapsing problem, whereby GANs converge to a single optimum within a multi-modal latent space distribution Lala et al. (2018). Multi-GANs have been designed to address this by using different partitioned generators that sample different regions of the feature space. For example, in MG-GAN Hoang et al. (2018), several generators are trained simultaneously to produce a partitioning of the latent space. This enables multiple generators to collectively encode a multi-modal latent space, avoiding the mode collapsing problem by introducing a classifier whose goal is to determine which generator a sample came from. The objective function of the generators then aims to beat the discriminator (as in traditional GANs), whilst at the same time maximising the difference between generators by maximising the entropy of the classifier. The use of the classifier in this case avoids the task of inferring the probability distributions of the generators directly from samples, similarly to the role of the discriminator in single GAN models. As we will discuss, GANs and especially multi-GANs, represent typical examples of NNCs.

Other research has explored the behaviour of individual ANNs which interact spatially, rather than at architecture level. These include Neural Cellular Automata (NCA), such as in Gilpin (2019) which replace the cells of a cellular automata system with ANNs, multi-agent systems such as in Kajić et al. (2020), and, more recently, systems of many thousands to millions of interacting ANNs such as in Zheng et al. (2018); Suarez et al. (2021). We believe these works are early examples of NNCs.

One means for addressing computational bottlenecks is to distribute the ANNs across computational nodes. These methods can be broadly classified under the header of federated learning (FL) Li et al. (2020). FL consists of sets of ANNs with identical architecture but different weight updates. A training dataset is distributed across the ANNs, so that each ANN is trained on a subset of the total dataset. The ANNs then update their weights, which are summed and passed to a central ANN. The amount by which each network contribution is weighted in the sum approximates the certainty of its weight parameter estimates. For example, this could be given by the proportion of data used to train the corresponding network at each epoch, as in Zhu et al. (2021). Research in this area has been motivated by concerns with data privacy and protection and many different approaches to FL have been proposed. For a review of this area, see, for example Li et al. (2020). FL highlights the impact of structure, such as whether the individual ANNs are centralised, decentralised or distributed, as we discuss in subsection 3.2.

Finally, we note that typical collective systems are inherently relational. Geometry deep learning and graph neural networks (GNNs) are those that are designed to operate over graphical or relation data Bronstein et al. (2017); Zhou et al. (2020). Recent work has shown that GNNs can be effectively applied to collective systems, including in learning both local and global interaction rules Battaglia et al. (2018); Cranmer et al. (2020). As our conception of NNCs is inspired by collective systems and, as such, is also relational, one may consider how NNCs differ GNNs. Our conception of NNCs is similar to GNNs, and may even be considered as a form of GNNs, in that they operate over graph-like structures (as with FL). However, the distinction between GNNs and NNCs is that GNNs perform complex computations on graph-structured data whereas NNCs structure their computational components on a graph. For example, a GAN might be considered as a two-node NNC, where one node is the discriminator and one is the generator. However, current combinations of GANs with GNNs typically use GNNs to generate graph structures that are used downstream by a GAN. In the next section, we discuss the salient features of NNCs in more detail.

## 3 Neural Network Collectives

Neural network collectives (NNCs), shown in Figure 1, use collective intelligence to design networks of artificial neural networks (ANNs). The defining components of an NNC are the nodes, edges, and information passed between nodes. We posit that, under appropriate conditions, NNCs will display novel learning structures, as is commonly observed in complex systems. We refer to these learning structures as *emergent intelligence* and discuss potential forms that emergent intelligence may take. We discuss each aspect in more detail in what follows.

### 3.1 Layer 1: Artificial Neural Networks as Nodes

The first layer of NNCs are the individual ANNs which comprise the nodes of the collective. The nodes of the NNC are extremely complex and inhabit the largest design space. They are composed of individual ANNs, hence inheriting all of the complexity of deep learning, including variations in architecture, learning algorithms and loss functions. As with ANNs, we expect them to be stochastic (such that ANNs with identical input and architecture will result in differing outputs) and nonlinear. They may range from simple MLPs to more complex generative models such as GANs.

As there are multiple nodes across the collective, there are additional design choices that need to be considered such as the degree of similarity or diversity in nodes, what input data is passed to the nodes, and what output data is shared between nodes. For example, all the ANNs may be identical, as in Siamese neural networks (SNNs), where copies of an input are passed to two or more sub-networks that have the same ANN configuration, parameters and weights Chicco (2021). These sub-networks are used to perform classification by considering similarities in feature vectors, through the use of a specific loss function (such as a contrastive Wang & Liu (2021) or triplet loss Dong & Shen (2018)). SNNs, like GANs, can be considered as a form of NNCs where nodes are identical but input data is varied and top-level tasks (such as classification) are performed by a pooling operation.

Alternatively, nodes may comprise of ANNs with differing architectures, as exemplified by GANs, which include either one generator-discriminator pair, or multi-GANs that make use of multiple generators or discriminators. Multi-GANs distribute computational complexity and are typical NNCs in

that respect. For example, in Hardy et al. (2019), a multi-GAN model is proposed which consists of one generator and multiple discriminators. Each discriminator has a subset of test data, and the discriminators exchange information amongst themselves. In Nguyen et al. (2017), two discriminators are used which optimise two different cost functions in order to avoid the mode-collapse problem. The work in Choi & Han (2021) also explores the use of multiple discriminators to promote diversity amongst discriminators, specialising in different clusters of a dataset by using 'multiple choice learning'. In this way, a mixture of discriminators can be used to collectively differentiate between different data-sources.

Both SNNs and GANs highlight how similarity and diversity in connected ANNs can dramatically change both their learning performance and their suitability for certain tasks. Aspects of consensus and differentiation, as outlined in subsubsection 2.1.2, are likely to be critical to NNC design principles. Future research might consider how NNCs perform with a large number of identical NNs, such as in *MAgent* Zheng et al. (2018), or mixed sub-populations of ANNs optimised for different tasks, as with multi-GANs. We expect that the choice of similarity or diversity in a NNC will be largely task specific except in those cases where NNCs are constructed as objects of study themselves.

### 3.2 Layer 2: Structure of a Collective

#### 3.2.1 On Connecting Artificial Neural Networks

The structure of the collective can be affected both locally and globally. Locally, one can specify the directionality of nodes, such as whether nodes can both send and receive shared data. Equally, one may consider the respective strengths of the connection, whether there is error in the system, and whether there are open connections between individual nodes and their contextual environment. For example, NCAs are bi-directional, where local information is both shared and received by individual ANN cells.

Broadly speaking, the global structure of a collective can be understood within the paradigm of centralised versus distributed approaches, as discussed in subsubsection 2.1.1. Most current neural network implementations consider centralised systems where data is aggregated and processed using a central component. An example of a centralised system in which training data is partitioned across different worker sub-networks can be found in Zhu et al. (2021), whereby each MLP worker processes the data independently of others. A single central processor then aggregates the network weight estimates and re-distributes the updated weights back to individual workers. The work in Hu et al. (2019) provides a FL example of a decentralised algorithm. Using a segmented gossip method, worker network weights are locally aggregated using several computational nodes. Finally, MD-GAN Hardy et al. (2019) gives an example of a distributed system, where no centralised processors exist. Instead, a local gossip algorithm is applied across a set of distributed discriminators.

The global structure of NNCs give additional design flexibility. For example, one can use many of the traditional graph structures such as fully connected or hierarchical graphs in designing NNCs. Additionally, one may consider structures based on some underlying geometry such as grid-based nearest neighbours as with NCAs Gilpin (2019). However, future NCAs may not require pre-determined structures, choosing instead to alter and optimise the links dynamically, as occurs with ANNs.

#### 3.2.2 On Communicating Artificial Neural Networks

It is important to consider what information is being passed across the collective. At the most basic level, this could be feature data based on local computations by the nodes. However, higher dimensional data may also be passed, such as sound, text, or image data. More interesting is the possibility that NNCs pass multi-modal data, with specific nodes receiving, transforming, and sending data based on their position within the collective.

It is important to note that errors in the initial receiving and distribution of data, as well as the computation performed by each node, can propagate and amplify as data is shared across the system. One way to address this is to account for error and uncertainty in the passed data. This could be achieved by, for example, quantifying uncertainty such that nodes with higher certainty on output are given more weight. Alternatively, individual nodes could pass a full probability density that describes their respective posterior distribution over their received input or output as a form of Bayesian NNCs.

We have so far been describing traditional communication content between nodes in a graph, such as features, images, text, and even densities. However, one of the benefits of NNCs is that individual nodes also have access to their own properties. As a result, NNCs could be designed such that individual nodes share their own weights. This would mean that NNCs are passing ANNs between nodes, possibly to a central processing unit as has been proposed in, for example, methods of federated learning Zhu et al. (2021).

### 3.3 Layer 3: Emergent Intelligence

Above, we gave our initial perspectives on how NNCs can be designed with respect to the nodes and edges of the collective. However, the central benefit of using a NNC is the collective intelligence gained by ensembling multiple varied models, leveraging the wisdom of artificial crowds. Below, we discuss why we believe emergent intelligence is possible in the context of NNCs, and what form it might take.

In order to best make use of the artificial crowd, careful choices should be made about how information passed between nodes is ensembled. This might be a simple aggregator (as in federated learning), or more complex operations such as the message passing protocol used in GNNs Battaglia et al. (2018). Alternatively, methods from collective intelligence such as voting-based consensus algorithms might be applied Dimakis et al. (2010). Most intriguing of all is the possibility that, under certain conditions, a NNC will ensemble models in such a way that the collective will demonstrate higher-order learning capabilities. These capabilities would be more successful than that of the individual nodes but also potentially different in other meaningful ways. We believe this is possible for two reasons.

First, recent analysis on neural networks have highlighted the existence of circuits of networks that develop task-specific expertise. For example, CNNs contain emergent receptive fields that are sensitive to sub-features of image data Luo et al. (2016); Cammarata et al. (2020). Secondly, we note the many aforementioned examples of complex systems which display emergent computational abilities, along with Ramos-Fernandez et al. (2020) and references therein. Hence, we believe that computation scales and that intelligence structures will emerge from scaled computation.

If emergent intelligence is observed in NNCs, we expect it to feedback into the lower units of the collective. This form of top-down feedback has been identified as a potential mechanism in collective intelligence, see, for example, Biro et al. (2016); Flack (2017); Diaconescu et al. (2019; 2021), and may end up being a form of reinforcement which strengthens the emergent intelligence or maintains the critical conditions under which it emerges. Moreover, the high-order intelligence may allow for different forms of learning, whereby the collective learns how to learn. This type of meta-learning could, for example, be formed by certain regions of the collective being used to retain information, as has recently been observed in transformers Elhage et al. (2021). Alternatively, entirely new forms of learning and intelligence may be discovered, possibly through the application of meta-heuristic search algorithms such as evolutionary or genetic algorithms Stanley & Miikkulainen (2002).

## 4 Conclusion

We believe that NNCs could offer a rich and exciting new field of study from the perspectives of both deep learning and collective intelligence. We believe that designing these collectives will drive new research paradigms and lead to new applications of deep learning-based systems. Central to the design of NNCs is the question of how the principles of collective intelligence can be used to enhance the performance and range of applications of existing artificial neural network architectures. One immediate possibility is the development of algorithms that make use of distributed data processing, where data to be used for training is distributed across separate workers. As well as reducing the computational load required of each worker, these approaches offer enhanced user privacy and data security. Secondly, it may be possible for newly developed NNC algorithms to make use of emergent computational properties to overcome challenges experienced by single computational units, as in the example of multi-GANs. Finally, a speculative yet exciting opportunity is to implement meta-learning and self-tuning in NNCs by using the capacity of natural complex systems to adapt their behaviour to different unseen learning tasks as is expected for artificial general intelligence.

## REFERENCES


Hagit Attiya and Jennifer Welch. *Distributed computing: fundamentals, simulations, and advanced topics*, volume 19. John Wiley & Sons, 2004.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Giorgio Battistelli and Luigi Chisci. Kullback–leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, 50(3):707–718, 2014.

Giorgio Battistelli and Luigi Chisci. Stability of consensus extended kalman filter for distributed state estimation. *Automatica*, 68:169–178, 2016.

Levent Bayındır. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016.

Andrew Berdahl, Colin J Torney, Christos C Ioannou, Jolyon J Faria, and Iain D Couzin. Emergent sensing of complex environments by mobile animal groups. *Science*, 339(6119):574–576, 2013.

Dora Biro, Takao Sasaki, and Steven J Portugal. Bringing a time–depth perspective to collective animal behaviour. *Trends in ecology & evolution*, 31(7):550–562, 2016.

Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Mathematical model of self-organizing hierarchies in animal societies. *Bulletin of mathematical biology*, 58(4):661–717, 1996.

Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60(4):753–807, 1998.

Eric Bonabeau, Guy Theraulaz, Marco Dorigo, Guy Theraulaz, Directeur de Recherches Du Fnrs Marco, et al. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press, 1999.

Anthony Brabazon, Seán McGarraghy, et al. *Foraging-inspired optimisation algorithms*. Springer, 2018.

Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Eleanor R Brush, David C Krakauer, and Jessica C Flack. Conflicts of interest improve collective computation of adaptive social structures. *Science advances*, 4(1):e1603311, 2018.

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

Scott Camazine, Jean-Louis Deneubourg, Nigel R Franks, James Sneyd, Guy Theraula, and Eric Bonabeau. *Self-organization in biological systems*. Princeton university press, 2001.

Erik Cambria and Bebo White. Jumping nlp curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2):48–57, 2014.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: circuits. *Distill*, 5(3):e24, 2020.

Eduardo Castello, Tomoyuki Yamamoto, Yutaka Nakamura, and Hiroshi Ishiguro. Task allocation for a robotic swarm based on an adaptive response threshold model. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pp. 259–266. IEEE, 2013.

Davide Chicco. Siamese neural networks: An overview. *Artificial Neural Networks*, pp. 73–94, 2021.

Jinyoung Choi and Bohyung Han. Mcl-gan: Generative adversarial networks with multiple specialized discriminators. *arXiv preprint arXiv:2107.07260*, 2021.

Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

Vincent A Cicirello and Stephen F Smith. Wasp-like agents for distributed factory coordination. *Autonomous Agents and Multi-agent systems*, 8(3):237–266, 2004.

George F Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed systems: concepts and design*. pearson education, 2005.

Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33:17429–17442, 2020.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35 (1):53–65, 2018.

Ada Diaconescu, Louisa Jane Di Felice, and Patricia Mellodge. Multi-scale feedbacks for large-scale coordination in self-systems. In *2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pp. 137–142. IEEE, 2019.

Ada Diaconescu, Louisa Jane Di Felice, and Patricia Mellodge. Exogenous coordination in multi-scale systems: How information flows and timing affect system properties. *Future Generation Computer Systems*, 114:403–426, 2021.

Franz Dietrich and Christian List. Probabilistic opinion pooling. 2016.

Alexandros G Dimakis, Soummya Kar, José MF Moura, Michael G Rabbat, and Anna Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 459–474, 2018.

Yucheng Dong, Min Zhan, Gang Kou, Zhaogang Ding, and Haiming Liang. A survey on the fusion process in opinion dynamics. *Information Fusion*, 43:57–65, 2018.

Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

Marco Dorigo, Guy Theraulaz, and Vito Trianni. Swarm robotics: Past, present, and future [point of view]. *Proceedings of the IEEE*, 109(7):1152–1165, 2021.

Jens Ducrée, Max Gravitt, Ray Walshe, Sönke Bartling, Martin Etzrodt, and Tomás Harrington. Open platform concept for blockchain-enabled crowdsourcing of technology development and supply chains. *Frontiers in Blockchain*, pp. 47, 2020.

John RG Dyer, Christos C Ioannou, Lesley J Morrell, Darren P Croft, Iain D Couzin, Dean A Waters, and Jens Krause. Consensus decision making in human crowds. *Animal Behaviour*, 75 (2):461–470, 2008.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Andrea Falcón-Cortés, Denis Boyer, and Gabriel Ramos-Fernández. Collective learning from individual experiences and information transfer during group foraging. *Journal of the Royal Society Interface*, 16(151):20180803, 2019.

Andreas Flache, Michael Mäs, Thomas Feliciani, Edmund Chattoe-Brown, Guillaume Deffuant, Sylvie Huet, and Jan Lorenz. Models of social influence: Towards the next frontiers. *Journal of Artificial Societies and Social Simulation*, 20(4), 2017.

Jessica C Flack. Coarse-graining as a downward causation mechanism. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2109):20160338, 2017.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

Lorenzo Garattoni and Mauro Birattari. Autonomous task sequencing in a robot swarm. *Science Robotics*, 3(20):eaat0430, 2018.

Vijay K Garg. *Elements of distributed computing*. John Wiley & Sons, 2002.

Simon Garnier, Jacques Gautrais, and Guy Theraulaz. The biological principles of swarm intelligence. *Swarm intelligence*, 1(1):3–31, 2007.

William Gilpin. Cellular automata as convolutional neural networks. *Physical Review E*, 100(3): 032402, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

David Ha and Yujin Tang. Collective intelligence for deep learning: A survey of recent developments. *arXiv preprint arXiv:2111.14377*, 2021.

Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. In *2019 IEEE international parallel and distributed processing symposium (IPDPS)*, pp. 866–877. IEEE, 2019.

Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. Mgan: Training generative adversarial nets with multiple generators. In *International conference on learning representations*, 2018.

Zeng-Guang Hou, Long Cheng, and Min Tan. Decentralized robust adaptive control for the multiagent system consensus problem using neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3):636–647, 2009.

Chenghao Hu, Jingyan Jiang, and Zhi Wang. Decentralized federated learning: A segmented gossip approach. *arXiv preprint arXiv:1908.07782*, 2019.

Ivana Kajić, Eser Aygün, and Doina Precup. Learning to cooperate: Emergent communication in multi-agent navigation. *arXiv preprint arXiv:2004.01097*, 2020.

Ido Kanter, Wolfgang Kinzel, and Eran Kanter. Secure exchange of information by synchronization of neural networks. *EPL (Europhysics Letters)*, 57(1):141, 2002.

James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pp. 1942–1948. IEEE, 1995.

Wolfgang Kinzel, R Metzler, and I Kanter. Dynamics of interacting neural networks. *Journal of Physics A: Mathematical and General*, 33(14):L141, 2000.

Hiroaki Kitano. Biological robustness. *Nature Reviews Genetics*, 5(11):826–837, 2004.

Hiroaki Kitano. Towards a theory of biological robustness, 2007.

Sayeri Lala, Maha Shady, Anastasiya Belyaeva, and Molei Liu. Evaluation of mode collapse in generative adversarial networks. *High Performance Extreme Computing*, 2018.

Ming Li, Jian Weng, Anjia Yang, Wei Lu, Yue Zhang, Lin Hou, Jia-Nan Liu, Yang Xiang, and Robert H Deng. Crowdbc: A blockchain-based decentralized framework for crowdsourcing. *IEEE Transactions on Parallel and Distributed Systems*, 30(6):1251–1266, 2018.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

Thomas W Malone and Michael S Bernstein. *Handbook of collective intelligence*. MIT press, 2015.

James AR Marshall, Ralf HJM Kurvers, Jens Krause, and Max Wolf. Quorums enable optimal pooling of independent judgements in biological systems. *Elife*, 8:e40368, 2019.

R Metzler, W Kinzel, and I Kanter. Interacting neural networks. *Physical Review E*, 62(2):2555, 2000.

Ida Momennejad. Collective minds: social network topology shapes collective cognition. *Philosophical Transactions of the Royal Society B*, 377(1843):20200315, 2022.

Mehdi Moussaid, Simon Garnier, Guy Theraulaz, and Dirk Helbing. Collective information processing and pattern formation in swarms, flocks, and crowds. *Topics in Cognitive Science*, 1(3): 469–497, 2009.

Toshiyuki Nakagaki. Smart behavior of true slime mold in a labyrinth. *Research in Microbiology*, 152(9):767–770, 2001.

Zohar Neu and Luca Giuggioli. A sociogenesis movement model for cohesive collective motion in unbounded domains. 2022.

Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. *Advances in neural information processing systems*, 30, 2017.

Joseph Ortiz, Talfan Evans, and Andrew J. Davison. A visual introduction to gaussian belief propagation. *arXiv preprint arXiv:2107.02308*, 2021.

Gabriel Ramos-Fernandez, Sandra E Smith Aguilar, David C Krakauer, and Jessica C Flack. Collective computation in animal fission-fusion dynamics. *Frontiers in Robotics and AI*, pp. 90, 2020.

Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pp. 10–20. Springer, 2004.

Gokay Saldamli, Ardalan Razavi, et al. Swarm robotics meets blockchain to deploy surveillance missions. In *2020 32nd International Conference on Microelectronics (ICM)*, pp. 1–6. IEEE, 2020.

Takao Sasaki and Dora Biro. Cumulative culture can emerge from collective intelligence in animal groups. *Nature communications*, 8(1):1–6, 2017.

Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36, 2020.

Thomas D Seeley, P Kirk Visscher, and Kevin M Passino. Group decision making in honey bee swarms: When 10,000 bees go house hunting, how do they cooperatively choose their new nesting site? *American scientist*, 94(3):220–229, 2006.

Alina Sîrbu, Vittorio Loreto, Vito DP Servedio, and Francesca Tria. Opinion dynamics: models, extensions and external effects. In *Participatory sensing, opinions and collective awareness*, pp. 363–401. Springer, 2017.

Sandro Skansi. *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer, 2018.

Kenneth O Stanley and Risto Miikkulainen. Efficient evolution of neural network topologies. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pp. 1757–1762. IEEE, 2002.

Joseph Suarez, Yilun Du, Clare Zhu, Igor Mordatch, and Phillip Isola. The neural mmo platform for massively multiagent research. *arXiv preprint arXiv:2110.07594*, 2021.

David JT Sumpter and Stephen C Pratt. Quorum responses and consensus decision making. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1518):743–753, 2009.

Guy Theraulaz, Eric Bonabeau, and JN Denuebourg. Response threshold reinforcements and division of labour in insect societies. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1393):327–332, 1998.

Sebastian Thrun and Yufeng Liu. Multi-robot slam with sparse extended information filers. In *Robotics Research. The Eleventh International Symposium*, pp. 254–266. Springer, 2005.

Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

Guoxing Wen, CL Philip Chen, Yan-Jun Liu, and Zhi Liu. Neural network-based adaptive leader-following consensus control for a class of nonlinear multiagent state-delay systems. *IEEE transactions on cybernetics*, 47(8):2151–2160, 2016.

Justin Werfel and Radhika Nagpal. Extended stigmergy in collective construction. *IEEE Intelligent Systems*, 21(2):20–28, 2006.

Wenwu Yu, Guanrong Chen, Zidong Wang, and Wen Yang. Distributed consensus filtering in sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6): 1568–1577, 2009.

Shiwei Zhang and Haitao Zhang. A review of wireless sensor networks and its applications. In *2012 IEEE international conference on automation and logistics*, pp. 386–389. IEEE, 2012.

Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Hangyu Zhu, Haoyu Zhang, and Yaochu Jin. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems*, 7(2):639–657, 2021.