Jack of All Trades, or Master of One: Information Decomposition Reveals Distinct Features of Generalizable vs. Specialized Neural Representations

Authored by Alexandra Maria Proca

Supervised by
Dr. Matthew Crosby, Imperial College London
Dr. Pedro Mediano, University of Cambridge
Dr. Jun Wang, University College London

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in Machine Learning

Department of Computer Science
University College London
London, United Kingdom
2021

What is real is not the external form, but the essence of things. Starting from this truth it is impossible for anyone to express anything essentially real by imitating its exterior surface.

Constantin Brâncuși

Abstract

Although biological and artificial neural networks are widely studied in the fields of neuroscience and machine learning, there is active ongoing work trying to understand how both systems encode, compress, and transfer information. Partial information decomposition (PID), an extension of Shannon mutual information to the multivariate case, has shown to be a useful tool for studying various systems. PID partitions the mutual information between a set of source variables and a target variable into unique (information carried by one variable but not the other), redundant (mutually-shared information), and synergistic information (information only available from the presence of all sources). Previously, synergy has been strongly linked to complex human cognition and has been related to theories of consciousness based on information theory. However, these relations have had limited study in the context of machine learning, especially in relation to complex tasks; we still do not know precisely how and why synergy emerges in learning systems. To address this, we analyzed synergistic and redundant information in a supervised learning setting to study network dynamics in relation to task and to stochasticity of training method. We also studied reinforcement learning agents performing tasks of increasing complexity in the Animal-AI environment. Our work found that neural networks compress synergistic and redundant information into unique information and lower-order forms possibly due to the inefficiency of distributed representations. Furthermore, our results suggest that synergy is used by neural networks to generalize learned representations to new tasks, rather than re-learning unique information mappings. These findings provide a new interpretation of synergy: as a buffer for creating both higher-level representations and generalizing learned encodings to new settings, which presents a trade-off between efficient mappings within the neuronal information space.

Acknowledgments

I would like to thank Dr. Matthew Crosby and Dr. Pedro Mediano for their incredible supervision. I am very grateful to have had the opportunity to work on such an interesting and stimulating project, and have had their guidance, creative ideas and insights, and encouragement throughout the process. I express my gratitude to Professor Jun Wang for his support and supervision during my thesis. Additionally, I want to thank Andrea Luppi and Dr. Fernando Rosas for our insightful conversations. I also want to thank the UCL Friends and Alumni Association for supporting my degree. Thank you always to my parents and brothers, Andrei and Adrian, for their unconditional love and support. Finally, I would like to thank Julian Coda-Forno, Hannah Teufel, and Umais Zahid for their friendship and advice, and for our many hours in the UCL Student Centre spent discussing neuroscience, consciousness, intelligence, and morality.

Contents

List of Figures

Chapter 1

Introduction

Understanding how the brain processes information to learn, behave intelligently, and give rise to conscious experience is a subject of wide-study within the field of neuroscience. It has also been of great interest in machine learning research, as a model and source of inspiration for new developments. However, both fields currently have a limited understanding of exactly how these processes occur. The brain is considered the most complex known object in the universe, containing 80 billion neurons with up to 15,000 synapses each [1]. Even with the refinement of various techniques and experiments [2], much knowledge is still left to discover. Paralleling the difficulties of studying the brain, artificial neural networks are also currently poorly understood. Much of deep learning is grounded upon empirical evidence of performance in particular tasks, rather than a strong theoretical basis. We still do not fully know why neural networks perform as well as they do, what exactly occurs during the learning process, or how to interpret the computations and decisions made by our models [3, 4]. A point of commonality in both biological and artificial neural networks, is that information is transmitted from sources (groups of neurons) to targets (other groups of neurons). This information is processed through some form of computation and gives rise to representations which are used for some function. Although artificial neural networks are not a perfect model of the brain, understanding the same flow of information can contextualize the behavior of groups of neurons, allowing us to interpret different scales of neural behavior.

Partial information decomposition (PID) [5], an elegant framework from the field of information theory which divides Shannon mutual information beyond two variables into redundant (mutually-shared information), unique (information carried by one variable but not the other), and synergistic (information only available from the presence of all variables) information, has been used to study various systems. PID allows for the quantification and description of the interactions occurring within a group of sources as they are collectively transmitted to a given target. This idea can be leveraged to analyze the way information is encoded, compressed, and transferred within a system. One of PID's measures, synergy, quantifies information that only exists as a result of an entire set of sources being considered collectively, being more than the sum of its parts. Previous work has used synergy

to observe the way collective behavior gives rise to complexity, studying it in the interrelated contexts of artificial neural networks, emergence, and the brain; we review some of this research in Chapter 3. In this realm, one line of work [6, 7] has shown that synergy is strongly linked to complex human cognition and has used it to bridge two major theories of consciousness, namely Global Neuronal Workspace Theory [8, 9, 10] and Integrated Information Theory [11, 12, 13]. Further studying synergy through computational methods, such as reinforcement learning, can provide better understanding of its relation to learning and yield new insights for both neuroscience and machine learning.

In this work, we seek to find computational basis for neuroscientific findings of synergy's importance to complex cognition and leverage the tools of machine learning to explore the relationship of synergy to learning. More specifically, we ask the following questions: How does synergy behave in reinforcement learning agents as task complexity increases? Are levels of synergy correlated with particular tasks or processes? Do agents with increased levels of synergy perform differently than those with lower levels? To do this, we analyze synergistic interactions in both supervised learning and reinforcement learning settings. We use the Animal-AI Environment [14], a reinforcement learning environment inspired by research in animal cognition, to create new tasks and test for particular abilities.

Our work discovered a number of different findings for which we provide detailed analysis. Both sets of experiments revealed that the information decomposition of neural networks is strongly influenced by the task being performed, such that some tasks are highly-synergistic and others are not. We also found that neural networks compress distributed representations (i.e., synergy and redundancy) into unique and lower-order synergistic/redundant information, and that lower-order information decomposition is correlated with full-order information decomposition in small networks. We showed that dropout can be seen as a form of regularization which increases redundancy of important information and reduces synergy of unimportant information, but preserves essential synergistic information. Finally, our results suggest that synergy is used by networks to transfer learned representations to new tasks by extracting additional information from existing mappings, rather than re-learning unique representations.

This thesis is structured as follows. First, in Chapter 2, we introduce relevant preliminary background information about the fields of reinforcement learning and information theory, as well as describe the Animal-AI Environment in detail. Next, in Chapter 3, we review previous literature which has used synergy and conceptually-related measures (namely, integrated information) to study neural networks, emergent behavior, and theories of cognition and consciousness; as we will describe, these subjects can all be related to one another and are an important basis for our studies. In Chapters 4, 5, and 6, we describe our experimental methodology and provide thorough analysis of our results. Finally, in Chapters 7 and 8, we relate our findings to ongoing research and discuss future directions for our work.

Chapter 2

Background

The current work is built off the fields of reinforcement learning and information theory. First, Section 2.1 summarizes relevant reinforcement learning background information. Following that, Section 2.2 introduces the Animal-AI Environment used for this project. Finally, Section 2.3 provides an introduction to information theory with a particular focus on partial information decomposition.

2.1 Reinforcement Learning

This section is meant to serve as a preliminary background for the reinforcement learning problem and methods for approaching it. We begin with a basic summary of reinforcement learning, predominantly based off of Sutton and Barto's book [15]. We then continue with an introduction to policy gradient methods and the Proximal Policy Optimization algorithm we use in our study. Readers familiar with deep reinforcement learning can skip to Section 2.2, as we do not introduce anything novel in this section.

2.1.1 Introduction

Reinforcement learning (RL) provides a strong framework for modeling cognition, using intrinsic states and goals, and extrinsic rewards and environments. In contrast to other areas of machine learning, RL is based on a learner's (*agent's*) interaction with its surroundings (*environment*) as the means by which it learns, forcing it to be active in its acquisition of information rather than dependent on information provided from an external source. This allows for generalization to complex behavior and tasks that may not be easily-defined outside of the need to reach some goal.

2.1.2 Preliminaries

Markov Decision Processes, later described in this section, can be formally defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P)$, where

- $\mathcal{S}$ is the set of all possible states
- $\mathcal{A}$ is the set of all possible actions
- $\mathcal{R}$ is the set of all possible rewards, such that $\mathcal{S} \times \mathcal{A} \to \mathcal{R}$
- $P(R_{t+1}, S_{t+1}|S_t, A_t)$ is the set of transition probabilities

In a RL paradigm, at each timestep t an agent is in a state $S_t \in \mathcal{S}$ and receives a *reward* $R_t \in \mathcal{R}$ from its environment and produces an *action* $A_t \in \mathcal{A}$. The observation relays some information about the state of the environment and the reward provides feedback about the agent's relative performance at that timestep. The behavior of the agent is dictated by its *policy*. More specifically, an agent's policy π is a mapping from its state s to an action a. It can either be a deterministic action given the state, $\pi(s)$, or a stochastic sampling from a distribution of actions given a state, $\pi(a|s) = p(a|s)$. The goal of the agent is to maximize its expected cumulative reward, the expected *return* G_t. Often, when computing the return, the reward at each subsequent timestep is devalued by some *discount* $\gamma \in [0, 1]$ so as to favor earlier rewards over later rewards and bound expected reward in non-episodic domains:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{2.1}$$

To provide some information about the agent's state, at each timestep the agent receives an *observation* O_t from the environment. An agent's state is a result of the history of its interactions with the environment (observations, actions, rewards). An agent's observation O_t is equivalent to its state S_t if full-observability of the environment is assumed. Furthermore, in a setting where an agent's state encompasses all relevant information about an agent's history, the transitions between states are Markovian [denoted as a *Markov Decision Process* (MDP)]. In this case, the agent's history prior to t contains no additional useful information. However, the information provided to an agent is often incomplete and may not represent long-term dependencies. In such a setting, the observations are not Markovian and it is referred to as a *Partially Observable Markovian Decision Process* (POMDP).

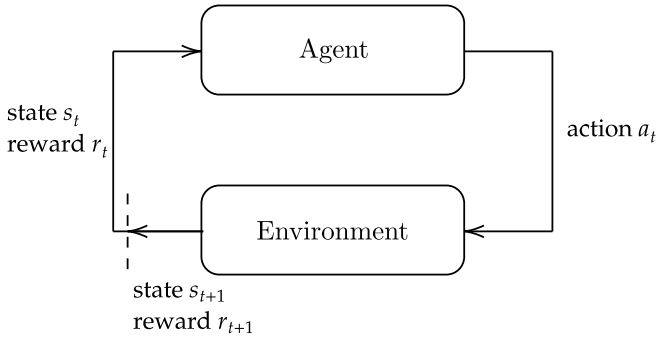


Figure 2.1: The reinforcement learning problem. At each timestep t, an agent receives a state s_t and reward r_t from the environment and produces an action a_t. The environment then produces the next state s_{t+1} and reward r_{t+1} and so on, resulting in a loop of complementary interactions.

For the purposes of the theory behind RL, an MDP is generally assumed to be approximated. The dynamics of the agent and environment can then be defined through probability distributions of transitions between subsequent timesteps:

$$P(R_{t+1} = r, S_{t+1} = s' | S_t, A_t). \tag{2.2}$$

Leveraging the Markov property allows for the computation of other relevant values, such as the *state-transition probabilities*,

$$p(s_{t+1}|s_t, a_t) = P(S_{t+1} = s' | S_t = s, A_t = a) \tag{2.3}$$

$$= \sum_{r \in \mathcal{R}} p(s', r|s, a). \tag{2.4}$$

and *expected reward*,

$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] \tag{2.5}$$

$$= \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a), \tag{2.6}$$

Many RL algorithms attempt to estimate the *value functions* of states, $v(s)$, or of state-action pairs, $q(s, a)$, which allow the agent to evaluate the favorability of being in a particular state or of performing a particular action in a particular state. The value function is dependent on the agent's policy and is defined by the expected return from that state, called the *state-value function* $v_\pi(s)$, or state-action pair, called the *action-value function* $q_\pi(s, a)$. Furthermore, these functions can be defined recursively,

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] \tag{2.7}$$

$$= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a)[r + \gamma v_\pi(s')], \tag{2.8}$$

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \tag{2.9}$$

$$= \sum_{s',r} p(s',r|s,a)[r + \gamma \sum_{a'} \pi(a'|s')q_\pi(s',a')]. \tag{2.10}$$

The recursive representations in Equations (2.8) and (2.10) are referred to as *Bellman equations* for v_π and q_π, respectively.

To maximize its return, an agent must learn a policy through estimating value functions and choosing actions based on those estimations. A policy π is better than or equal to another policy π' if for all states $s \in \mathcal{S}, v_\pi(s) \geq v_{\pi'}(s)$. There always exists one policy that is better than or equal to all other policies, which is the *optimal policy*. The optimal policy will utilize the *optimal state-value function*, $v_*(s) = \max_\pi v_\pi(s)$ and *optimal action-value function* $q_*(s,a) = \max_\pi q_\pi(s,a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$ so as to select actions that result in the agent being in states yielding the highest return. Furthermore, the action-value function can be written in terms of the state-value function,

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}|S_t = s, A_t = a)]. \tag{2.11}$$

Intuitively, this demonstrates an action-state pairing as a function of the expected generated reward and the value of the subsequent state. Thus, both value functions must be complementary and self-consistent. Furthermore, the Bellman equations can be rewritten as *Bellman optimality equations*,

$$v_*(s) = \max_{a \in \mathcal{A}} \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1})|S_t = s, A_t = a] \tag{2.12}$$

$$= \max_{a \in \mathcal{A}} \sum_{s',r} p(s',r|s,a)[r + \gamma v_*(s')] \tag{2.13}$$

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a')|S_t = s, A_t = a] \tag{2.14}$$

$$= \sum_{s',r} p(s',r|s,a)[r + \gamma \max_{a'} q_*(s',a')]. \tag{2.15}$$

With an optimal state-value function v_*, the optimal policy π_* then becomes the assignment of nonzero probabilities to only the actions which yield the maximum in the Bellman optimality equation. Thus, the optimal policy is exploitative (*greedy*) of the optimal state-value function. The optimal action-value function q_* concatenates the evaluation of next-possible states into optimal actions at t, further simplifying the problem.

The growing field of RL continues to find new approaches and solutions to approximating such value functions. One important consideration is the tradeoff between *exploration* and *exploitation*. In order to adequately evaluate and approximate the value of particular states and actions over others, an agent must have interacted sufficiently with its environment in

novel ways (exploration). However, the state and action space may be high-dimensional, or potentially infinite, making the exploration of every setting unfeasible with finite time and compute. Thus, methodical exploration is important so that the agent may exploit the information it has acquired about the environment and select actions that will more efficiently lead to better performance (exploration vs. exploitation).

2.1.3 Proximal Policy Optimization

The field of deep RL, using neural networks as value function approximators, has spawned several different approaches in recent years among which include deep Q-learning [16] and various policy gradient methods [17, 18]. *Proximal policy optimization* (PPO) [19] is one-such policy gradient method which, similarly to *trust region policy optimization* (TRPO) [18], was motivated by the goal to take the largest gradient steps on a policy without overstepping and substantially decreasing performance. PPO has proven to be robust and been successful on a variety of different tasks compared to other methods. The algorithm has two main variants, one using clipping and the other using a KL-constraint. The clipped version of PPO was shown to perform better out of the two in the original work [18] and is generally the preferred variant of PPO; it is also the variant that we use in the following work.

Policy Gradient Methods

Policy gradient methods are *on-policy*, meaning that their policies update directly from interacting with their environment, rather than storing experiences in a replay buffer (as is the case in Q-learning). These methods compute a gradient estimator of the policy and perform stochastic gradient ascent on it. A commonly used estimator is

$$\hat{g} = \hat{\mathbb{E}}_t[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t]. \tag{2.16}$$

The expectation $\hat{\mathbb{E}}_t[\dots]$ computes the average over a batch of samples accumulated at time t with the current policy π_θ and *advantage function* $\hat{A}_t$, which estimates the relative value of the selected action in the current state ($A(s,a) = q(s,a) - v(s)$). The value function $v(s)$ is estimated by a neural network, as described in the definition of actor-critic architectures below. The gradient estimator (2.16) is computed by differentiating the loss,

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t[\log \pi_\theta(a_t|s_t)\hat{A}_t]. \tag{2.17}$$

Importantly, optimization is performed once on the same trajectory so as not to create a policy update that oversteps.

Trust Region Policy Optimization

In contrast to vanilla policy gradient methods, TRPO [18] uses a different objective function and a KL constraint to control the magnitude of the update from its previous policy.

Formally, this is expressed as

$$\underset{\theta}{\text{maximize}}\ \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] \tag{2.18}$$

$$\text{subject to } \hat{\mathbb{E}}_t[D_{KL}[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]] \leq \delta. \tag{2.19}$$

TRPO is an effective algorithm, but its use of a second-order method makes the optimization process more difficult. PPO seeks to improve upon TRPO by using a simpler first-order optimization while maintaining reliable performance.

Clipped Surrogate Objective

Without the KL constraint used in TRPO, maximizing the objective function (2.18) leads to large policy updates. To instead create a first-order method, PPO modifies TRPO's objective function by penalizing policy changes using the ratio between the action probabilities of the proposed policy and the previous policy, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$. By clipping, it limits the update to the new policy, constraining $r_t(\theta)$ from moving further than the interval $[1-\epsilon, 1+\epsilon]$ for some small ϵ $(0.1-0.3)$. Thus, the minimum of the clipped and unclipped objective is taken so as to reach a lower bound on the unclipped objective:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon(\hat{A}_t)] \tag{2.20}$$

Using this function, the change in $r_t(\theta)$ bounded $(1+\epsilon)$ above for a policy that yielded a higher return (positive advantage) and bounded below $(1-\epsilon)$ for a policy that yielded a lower return (negative advantage). Bounding the magnitude of change prevents the optimization from taking too large of a gradient step in either direction and ensures that the probability of unfavorable actions does not decrease to 0.

PPO Algorithm

The PPO algorithm [19] employs an *actor-critic* architecture, which uses two separate neural networks to represent the policy (actor) and the value function (critic) independently of one another. The PPO model performs a T-step trajectory and samples interactions with the environment under the updated policy. It then computes advantage estimates and optimizes the policy using the objective function. For actor-critic networks using shared parameters, the loss function must include both the objective function $L_t^{CLIP}(\theta)$ and the value function squared-error term $L_t^{VF} = (V_\theta(s_t) - V_t^{\text{targ}})^2$. Additionally, an entropy term $S[\pi_\theta](s_t)$ can be included to promote exploration. This yields the loss function

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)], \tag{2.21}$$

where c_1, c_2 are coefficient hyperparameters.

Algorithm 1 PPO, Actor-Critic Style from Ref. [19]

for iteration=1,2,... **do**
  for actor=1,2,...,N **do**
    Run policy $\pi_{\theta_{old}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{old} \leftarrow \theta$
end for

2.2 Animal-AI Environment

This section aims to provide motivating context for and describe the RL environment we will be using in our work.

2.2.1 Context

In recent years, the field of RL has seen impressive achievements, reaching superhuman performance in challenging games, such as Atari [20] and Go [21]. While these successes have been important to the advancement of RL, they still lack the ability to more-broadly generalize. Although such models may be able to perform well in new test settings, these environments generally involve performing very similar types of tasks to those encountered in training and do not generally test for different types of cognition. Even with agents outperforming humans on certain tasks, animals and humans alike are far superior at adaptation.

Animal cognition and intelligence rests on a large bed of literature, having been studied widely in the field of comparative psychology. Over several decades, researchers have developed a range of tasks to test for particular behavioral and cognitive constructs with rigorous minimization of external confounding factors and noise so as to isolate the paradigm of interest. These tasks are designed to ensure that the objective cannot be solved without proper utilization of the cognitive function of interest. They generally require some behavior and exploitation of knowledge/environment in order to receive a positive reward (food), or avoid a negative reward (pain, discomfort). Some abilities tested for include object permanence [22], spatial memory [23], causal reasoning [24], and tool usage [25]. Animal cognition tasks provide an important and relevant framework for further studying and improving-upon human-like cognition in RL. Such tests resemble real-world problems that animals and humans alike face, which require manipulation of knowledge and experience to generalize to new settings.

2.2.2 The Environment

The Animal-AI Environment [14] is an artificial environment inspired by work in animal cognition and provides a platform for designing tasks similar to those used in behavioral-intelligence research. Different experimental configurations can be created to test for various cognitive functions; each configuration is created within an arena with the addition of simple objects and realistic physics (gravity, collision, friction, etc.). The agent is a sphere that can interact with its environment through its motion in eight directions – traveling through the environment, pushing objects, and climbing slopes. Its observation space can be adjusted to either be a monocular first-person view of pixel input or to use object-oriented raycasts which detail the distance to the particular objects the raycasts hit. Additionally, the agent receives information about its health, velocity, and global position. Just as most animal cognition tasks, each setting involves obtaining a positive reward (green/yellow sphere 'food') or avoiding a negative reward (red sphere 'punishment') within a time frame.

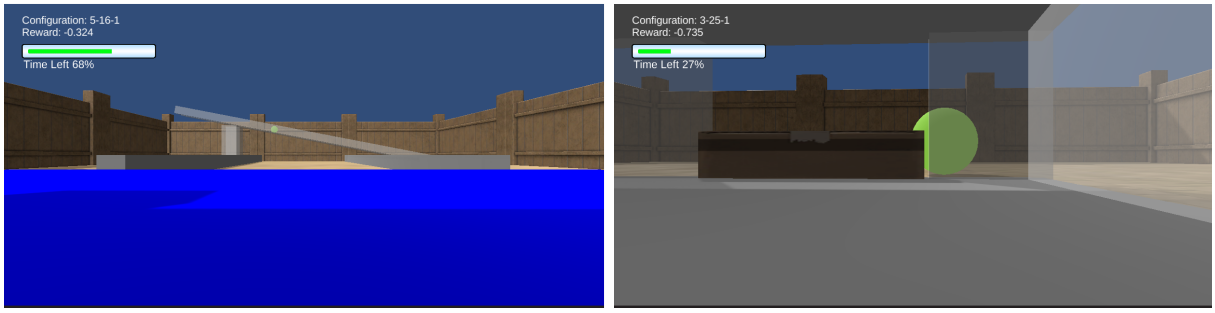


Figure 2.2: Animal-AI example configurations [14]. The left image displays a task involving food supported by other objects and the use of gravity. The right image displays a task recreating Thorndike's puzzle box experiments [26], where the agent must escape its enclosure and retrieve the reward outside.

A notable feature of the Animal-AI Environment is that it can employ different tasks for the same cognitive ability or for different types of cognitive skills, while maintaining the same basic setup so that the same agent can easily be trained or tested on new settings. Each task is loaded into the environment through a configuration file detailing the specific design and various parameters (object location, agent location, configuration-version, etc.) can be easily randomized across episodes. Thus, the environment better avoids over-fitting to one setting, forcing agents to be robust to changes. Additionally, the user has the flexibility to create new experimental designs, rather than being subject to the constraints of the environment creators.

Apart from Animal-AI's utility in developing generalizable RL, it also provides a novel method for studying neuroscience through biologically-inspired models. Previous research has used computer vision to study and compare to the visual system of primates and humans [27, 28, 29, 30]. Extending such style of work and analyzing biologically-inspired models in the context of high-level cognition will be an important next step in understanding in-

telligence, both for the purposes of machine learning and neuroscience. A powerful feature of this line of computational research is the ability to be able to manipulate experimental constructs easily and control for external noise to a higher degree than what is possible in human or animal research. Artificial neural networks also have the utility that they encompass the entire 'intelligent' system of interest and are accessible to full-observation, perturbations, and any other form of analysis, whereas only a limited amount of information can be extracted about a brain's neural system by using imaging/recording technology. Furthermore, isolating cognitive functions of interest in the Animal-AI environment may allow us to better understand the brain in the context of evolution facilitated by the need to perform certain tasks for survival.

2.3 Information Theory

This goal of this section is to provide a theoretical background of the aspects of information theory that are pertinent for the current study and their use. Although information theory also applies to continuous variables, for simplicity we will only consider and describe systems of discrete variables with a finite number of states, as used in our work.

2.3.1 Introduction

Since its formulation by Claude Shannon in 1948 [31], information theory has been used in a range of fields to study data compression and transmission, and the interaction between the two. As statistical inference and information theory have many similar components, information theory has been used to study the theory of deep learning [32, 33, 34, 35]. This application can be motivated by the desire to understand neural networks from the angle of information encoding, compression, and transfer. Furthermore, the framing of the brain as an information processing system has lead to information theory's wide use in the field of neuroscience [36, 37], as these tools may help us better understand the encoding of neural information transmitted in the brain.

2.3.2 Partial Information Decomposition

Much of the work in information theory assumes the simplest case, bivariate data, originating from data-receiver pairs in engineering. However, many difficult scientific inquiries involve three or more variables. One such example is the brain, where the encoding of an external stimulus is often dependent on the joint activity of multiple neurons (population coding) [38]. The brain also performs multisensory integration, combining information from various sensory modalities (vision, audition, olfaction, etc.) to give rise to unified experience [39]. *Partial information decomposition* (PID) [5] is one framework that generalizes a key measure, Shannon's *mutual information*, to the multivariate case through a nonnegative decomposition of information.

Preliminaries

First, to describe PID, some important measures in information theory must be defined, which can be found in more detail in Cover and Thomas' book on information theory [40]. The most central is that of *entropy*, which can be described as the amount of information, or *surprise*, contained in a random variable. If the value of a variable X with possible outcomes $x_1, x_2, \ldots, x_M$ is known with absolute certainty $[p(x_m) = 1, p(x_{\neg m}) = 0]$, it has a low entropy – there is no additional information or uncertainty provided by any particular realization. Alternatively, if the probability distribution of the variable is uniform with equal probability of being in any given state $[p(x_1) = p(x_2) = \cdots = \frac{1}{M}]$, the variable has a high entropy – there is a maximum amount of uncertainty about its state. Formally, the entropy H of a random variable X with possible values $x_1, \ldots, x_M$ and probability mass function $p(X)$ is defined as

$$H(X) = \mathbb{E}[-\log p(X)] \tag{2.22}$$

In the discrete case, this is equivalent to

$$H(X) = -\sum_{i=1}^{M} p(x_i) \log p(x_i). \tag{2.23}$$

Entropy is generally expressed using a logarithm of base 2 and measured in bits. Similarly, the *joint entropy* can be defined as the measure of surprise for a set of variables. For a variable Y with possible outcomes $y_1, y_2, \ldots, y_N$, the joint entropy of X and Y is defined as

$$H(X, Y) = -\sum_{i,j=1}^{M,N} p(x_i, y_j) \log p(x_i, y_j) \tag{2.24}$$

Furthermore, the *conditional entropy*, the amount of surprise of one variable X given another variable Y, can be defined as

$$H(X|Y) = H(X, Y) - H(Y) = -\sum_{i,j=1}^{M,N} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(y_j)} \tag{2.25}$$

in the discrete case.

Finally, the *relative entropy*, more commonly known as the *Kullback-Leibler divergence* (KL divergence), D_{KL} quantifies how different one distribution $p(X)$ is from another $q(X)$ in terms of surprise. The KL divergence is defined as the expectation with respect to p of the logarithmic difference between p and q,

$$D_{KL}(p\|q) = \mathbb{E}_p[\log p(X) - \log q(X)]. \tag{2.26}$$

In the discrete case this is equivalent to

$$D_{KL}(p\|q) = \sum_{i=1}^{M} p(x_i) \log \frac{p(x_i)}{q(x_i)}. \tag{2.27}$$

Entropy can be further used to compute other information theoretic values, one of which is mutual information. Mutual information $I(X;Y)$ is the measure of the average amount of information obtained about one variable X given another variable Y – the difference between the joint distribution $p(X,Y)$ and the product of the marginal distributions $p(X)p(Y)$. If X and Y are independent, their mutual information is zero – knowing one variable discloses no information about the other. Alternatively, if X and Y are deterministic inverse functions of each other such that for some function $f : Y = f(X), X = f^{-1}(Y)$, their mutual information is equal to the entropy of either variable alone as all information about one variable is given by the other. Mutual information can be written in several equivalent terms:

$$\begin{aligned} I(X;Y) &= D_{KL}(p(X,Y)\|p(X)p(Y)) \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X,Y) - H(X|Y) - H(Y|X) \\ &= \sum_{i,j=1}^{M,N} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}. \end{aligned}$$

Non-Negative Decomposition of Multivariate Information

PID separates the mutual information between a set of random variables (*sources*) $\boldsymbol{X} = \{X_1, X_2, \ldots, X_N\}$ and another random variable (*target*) Y into non-negative terms that describe the partial information contributed by subsets of $\boldsymbol{X}$ about Y. This can be performed in the simplest case with two source variables, where $\boldsymbol{X} = \{X_1, X_2\}$. In this setting, the mutual information $I(Y; X_1, X_2)$ describes the total information contributed about Y from $\boldsymbol{X}$. Considering X_1 and X_2 separately allows for further decomposition of the information contributed by either source. PID posits that a target's incoming information from two sources can either be (1) *unique* (information carried by one variable but not the other), (2) *redundant* (mutually-shared information), or (3) *synergistic* (information available only from the presence of all sources). We will refer to these as U, R, and S, respectively. For example, X_1 and X_2 contribute unique information if each source provides information about the target Y that the other source does not. Alternatively, if X_1 and X_2 contribute the same or overlapping information, that information is redundant. Finally, the information provided by X_1 and X_2 is synergistic if it is only present when both sources are considered jointly, rather than separately.

One example of synergistic information is the exclusive-OR (XOR) function $Y = X_1 \oplus X_2$, which can only be predicted by having both X_1 and X_2 (the parity of the set of sources). Notably, XOR is provably the only function that maximizes synergy [41], making it a useful

tool for studying this measure. We exploit this feature to study synergy in our experiments.

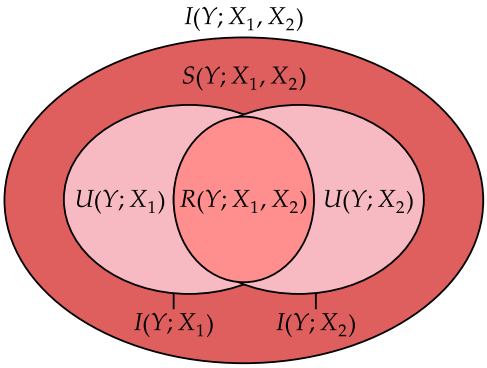


Figure 2.3: Partial information decomposition of two sources. The mutual information between the target Y and sources X_1, X_2 decomposes into unique (U), redundant (R), and synergistic (S) information accordingly.

With the measures of unique, redundant, and synergistic information, mutual information can be decomposed accordingly:

$$I(Y; X_1) = R(Y; X_1, X_2) + U(Y; X_1) \tag{2.28}$$
$$I(Y; X_2) = R(Y; X_1, X_2) + U(Y; X_2) \tag{2.29}$$
$$I(Y; X_1, X_2) = R(Y; X_1, X_2) + U(Y; X_1) + U(Y; X_2) + S(Y; X_1, X_2) \tag{2.30}$$

For an arbitrary number of sources, this decomposition can be structured into a partial information (PI) lattice, ordered with PI atoms that are more redundant towards the bottom and more synergistic towards the top. For example, with three variables, the bottom PI atom becomes $\{1\}\{2\}\{3\}$, representing redundant information shared by all three sources, and the top becomes $\{123\}$, representing synergistic information only present from the joint set of all three sources. Unique information is additionally denoted by $\{1\}$, $\{2\}$, or $\{3\}$. The presence of three or more sources decomposes into different combinations of redundant, synergistic, and unique information along the height of the lattice, such as $\{12\}\{3\}$, or the (redundant) information shared between the synergistic information of X_1, X_2 and the unique information of X_3.

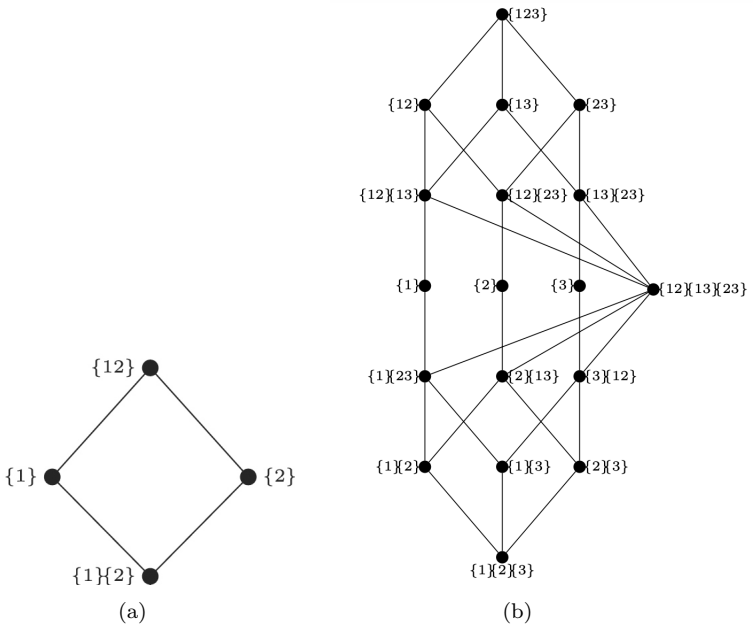


Figure 2.4: Partial information lattice for **(a)** two sources and **(b)** three sources, ordered with more synergistic PI atoms towards the top of the lattice and more redundant PI atoms towards the bottom of the lattice. Figures reproduced from [42, 43] with permission.

Notably, the decomposition does not specify a method to compute U, R, S – it only defines what mutual information of several variables is composed of. Additionally, there is no single widely-accepted method of computing these values, but instead a number of different proposed measures [5, 44, 42]. These methods vary in the values of the measures they yield, but overall are different formalizations of the same concept. Thus, finding consistent results across different measures is important for verifying that observed patterns are a consistent feature of synergistic interactions rather than a feature of the measure employed. Our work will use several of the proposed measures, which we define below. Notably, these measures each formulate a redundancy function. Given a redundancy function, the atoms can be obtained by computing mutual information and solving the linear equation system in Equations (2.28) to (2.30).

$I_{\min}$ Redundancy

The original work by Williams and Beer [5] proposed a quantification of redundancy titled $I_{\min}$. Intuitively, $I_{\min}$ encapsulates the idea that redundancy is the minimum information that any given source provides about the target, although sources may provide different information. $I_{\min}$ is computed using the *specific information* a variable X provides about a particular outcome y of a variable Y. This can be written as

$$I(Y = y; X) = \sum_{i=1}^{M} p(x_i|y) \log \frac{p(y|x_i)}{p(y)}. \tag{2.31}$$

Furthermore, mutual information can be written in terms of specific information,

$$I(Y;X) = \sum_{j}^{N} p(y_j) I(Y = y_j; X). \tag{2.32}$$

$I_{\min}$ redundancy is then formally expressed as

$$R_{\min}(Y; X_1, X_2) = I_{\min}(Y; X_1, X_2) \tag{2.33}$$

$$= \sum_{j=1}^{N} p(y_j) \min\{I(Y = y_j; X_1), I(Y = y_j, X_2)\}. \tag{2.34}$$

A convenient aspect of $I_{\min}$ is that it can theoretically be extended to an arbitrary number of sources, restricted only by computation and memory. However, this scales poorly for more than two sources, as the number of terms increases rapidly and makes the computation of these values intractable for more than approximately twenty sources. For most other measures, computing synergy requires computing all other redundancy atoms in the PID lattice and solving the linear system in Equations (2.28) to (2.30), which can become very computationally expensive. Fortunately, $I_{\min}$ (and I_{MMI}) bypasses this condition by using $I_{\max}$ (defined the same as $I_{\min}$ with min substituted for max). Thus, using $I_{\min}$ and $I_{\max}$, redundancy and synergy can be computed more efficiently:

$$R_{\min}(Y; X_1, \ldots, X_M) = I_{\min}(Y; X_1, \ldots, X_M) \tag{2.35}$$

$$= \sum_{j=1}^{N} p(y_j) \min\{I(Y = y_j; X_1), \ldots, I(Y = y_j, X_M)\} \tag{2.36}$$

$$S_{\min}(Y; X_1, \ldots, X_M) = I(Y; X_1, \ldots, X_M) - I_{\max}(Y; \boldsymbol{A} \in \{X_1, \ldots, X_M\} : |\boldsymbol{A}| = M - 1) \tag{2.37}$$

I_{MMI} Redundancy

Another proposed method of computing PID is that of Barrett (2015)'s I_{MMI} [44]. It defines redundancy as the minimum mutual information provided by any source, as opposed to its expectation as in $I_{\min}$:

$$R_{MMI}(Y; X_1, \ldots, X_M) = I_{MMI}(Y; X_1, \ldots, X_M) \tag{2.38}$$

$$= \min\{I(Y; X_1), \ldots, I(Y; X_M)\}. \tag{2.39}$$

With the I_{MMI} redundancy function, synergy is then derived as

$$S_{MMI}(Y; X_1, \ldots, X_M) = I(Y; X_1, \ldots, X_M) - \max_{\boldsymbol{A} \in \{X_1, \ldots, X_M\} : |\boldsymbol{A}| = M-1} \{I(Y; \boldsymbol{A}_1), \ldots, I(Y; \boldsymbol{A}_K)\} \tag{2.40}$$

Chapter 3

Related Work

In this chapter we review some of the literature that has leveraged information theory and partial information decomposition (PID) to study various systems, including artificial neural networks, emergent behavior, and the human brain.

3.1 Neural Networks

Understanding the way artificial neural networks learn has been a long-standing goal within the field of deep learning. Various attempts have been made at devising theories to explain the success of deep networks, popularly termed "black boxes". One vein of ongoing research has investigated such ideas in the context of information theory. Our study builds on some of the following work and uses it for guidance and inspiration in designing our experiments.

One famous application of information theory in deep learning is that of the information bottleneck (IB) principle. First, Tishby and Zaslavsky [32] reframed neural networks as Markov chains that propagate and transform successive representations about the input layer space into the output space. This formulation allowed for the study of such networks in the proposed *information plane*, or the plane of mutual information values of a given variable T (within the network's hidden layers) with the input variable X ($I(X;T)$) and the corresponding output variable (label) Y ($I(T;Y)$). The authors further asserted that optimal networks approach an IB bound, which trades off the information loss due to compression and the information preserved about the desired output of each layer from the previous layer.

In 2017, Schwartz-Ziv and Tishby [33] extended the use of the proposed information plane to analyze neural networks during training with stochastic gradient descent (SGD). Their primary contribution was the distinction of two subsequent stages of learning: a drift phase, where network layers increase mutual information on the labels $I(T;Y)$, and a diffusion phase, where network layers decrease mutual information on the input $I(X;T)$. Intuitively, the drift phase can be understood as a rapid increase in information about the input being propagated through the network for classification (fitting), as the gradient mean is initially

high and the function changes quickly towards the optimum. Alternatively, the diffusion phase can be interpreted as the minimization of the retainment of unimportant features (compression); the gradient mean is small with high variance as empirical error is saturated and overfitting may be avoided. The authors further claimed that the strong generalization capabilities of neural networks are causally related to the diffusion phase and that this phase is a property of SGD.

Although the results of these studies provided a new perspective for the theory of deep learning, they later revealed points of caution in the use of information theory. Saxe et al. [34] showed that the assumptions made by Refs. [32, 33] to compute finite discrete mutual information values in neural networks do not generalize to all settings. In particular, the information plane dynamics observed in [33] were attributed to the double-sided saturating tanh activation function. Using similar discretization methods (binning), the authors found that linear activation functions and single-sided saturating activation functions (e.g., ReLU) instead did not yield a compression phase. They specified that double-saturating nonlinearities create the observed compression as neurons enter a 'saturation regime', caused by the binning procedure. Furthermore, they observed that compression is not required for generalization and that the compression phase is not a result of the stochasticity of SGD. Finally, the study found that the compression of unimportant information and the increase of information about the input occur simultaneously during the fitting process, rather than in separate phases.

Goldfeld et al. [35] further studied the estimation of mutual information between input layers and hidden layers. They extended the line of IB work by analyzing deterministic and stochastic networks, finding that compression is a result of clustering of internal representations. The study established that regularization suppresses the formation of clusters, eliminating compression and again showing that compression may not be causally related to the generalization of networks. Furthermore, the authors asserted that the measure of binned mutual information does not accurately estimate mutual information and is instead a quantification of clustering.

Related to the aforementioned studies was the work of Tax et al. [45], which used PID to analyze the internal representations of discriminative restricted Boltzmann machines (DRBM), a type of generative neural network model, during training on modified MNIST images. The authors discovered two phases of learning: a first phase where source neurons predominantly contained redundant information about the target label, and a second phase where source neurons predominantly contained unique information about the target label. Intuitively, these patterns suggest that neurons specialize in the second phase, allowing for disentangled representations to emerge. The second phase also yielded an increase in synergy, perhaps indicating neuronal specialization allows for the representation of additional distributed features. Examining models of various sizes at convergence, the mutual information between single neurons and the target label was found to increase with the number of neurons, sug-

gesting that larger networks compressed less information into individual neurons and yielded more distributed representations. This was further supported by an increase in normalized synergy in larger subsets of neurons corresponding to the increase in size: synergistic information became more distributed, integrating larger groups of neurons as the size of the network grew. The order of grouping yielding meaningful synergy values is thus highly dependent on the size of network. This observation may partially explain why larger networks perform better as they use higher-order correlations between neurons, which may provide more complex representations by integrating many sources of information. Instead, smaller networks may be constrained to specialize neurons more (increasing their unique information) in order to more effectively compress information and yield disentangled representations.

Shifting focus from settings pertaining to classification, another line of research has studied networks in embodied agents interacting with an environment. In 2014, Albantakis et al. [46] investigated the behavior of small, adaptive logic gate networks ("animats") in an environment of increasing complexity. Animats mimic RL agents in some sense: they contain sensors, hidden elements, and motors which evolve to better suite their task using a genetic algorithm. In the study, animats learned to catch or avoid falling blocks of varying size in a game similar to Tetris. The authors used a measure conceptually-related to synergy – *integrated information* Φ [11, 13], or the causal information that exists beyond that generated by its parts (irreducible information). In addition to integrated information, the work also studied *concepts*, a formalized measure of the causal repertoire specifying what past/future states are or aren't possible. The authors found that as the task complexity increased, specifically in its requirement for sequential memory, animats evolved to have more concepts and have higher measures of integrated conceptual information within their networks. Additionally, animats with increasingly restricted sensory or motor capacities, which created a stronger reliance on memory, also exhibited an increase in concepts and more integrated conceptual structures. These results indicated that higher levels of integration are advantageous for constrained systems acting in complex environments. This may be explained by the fact that such networks can use higher-order concepts to accommodate limited access to information. This idea also relates to our results in this study, which found that synergistic information increased with the additional complexity of tasks and need for generalization in small networks which have a constrained neuronal information space.

The study of the effects of an agent's environment and 'body' (in the form of task complexity and sensory/motor capacities) on the agent's information dynamics treated the animat's 'brain' (neural network) as a separate system than its embodiment in Ref. [46]. A natural following question to ask might be: how is information transmitted when the body, brain, and environment of an agent are treated as a single system? Langer and Ay [47] sought to formalize this inquiry and particularly analyze how the complexity of solving a task is distributed among different parts (body, brain, environment) of the system. These ideas are related to the concept of morphological computation, which refers to processes performed by the body and environment that result in in a reduction of computational complexity for

the brain [48]. To study these ideas, Langer and Ay [47] used agents with two protruding sensors and a separate body, tasked with moving around a racetrack; the agents were required to move at all times and died if their body touched a wall. The entire system (brain, body, environment) was created as a graphical model consisting of a single 'world' node, two sensor nodes, two actuator nodes (corresponding to motor action), and two controller nodes (corresponding to the 'brain'). This framework allowed for the system interactions to be studied as a Markov Process and to use Planning as Inference [49] for training. The authors found that the amount of integrated information of the controller was dependent on the information flow to (sensors) and from (actuators) the controller. Additionally, they showed that as an agent can increasingly rely on the interaction between the environment and its sensors, the less integrated information the controller contains, and similarly, as the amount of information to the sensors decreases, the amount of integrated information in the controller increases. Both of these studies [46, 47] support the idea that the complexity of an environment and an agent's access to information about the environment are correlated to the information dynamics of the agent.

3.2 Emergence

The motivation to describe complex systems and behavior, specifically those which have interacting hierarchical levels (e.g., microscopic vs. macroscopic), has lead to several schools of thought. One such concept is that of *reductionism*, which claims that all layers of a system's hierarchy can be sufficiently explained by its components considered at the smallest scale. Alternatively, the construct of *emergentism* asserts that there are additional properties that arise when a system is considered at a larger scale. Although emergence does appear to be a commonly-observed phenomena, ranging from schools of fish to the fractal patterns of snowflakes to economic market behavior, there does not exist a single widely-accepted mathematical definition of it. To this end, research in systems theory has sought to formalize emergence. In addition to other domains studying various system behaviors, a quantitative measure of emergence could potentially provide a basis for better understanding both biological and artificial neural networks. Natural questions that arise in this area might be: how do macroscopic features exert causality over microscopic elements? How may higher-order concepts or features emerge in networks? How does mind emerge from neural patterns? Accordingly, recent work has leveraged PID to form a theoretical framework of emergence, as the measure of synergy has an intuitive relation to the concept of emergence: information emerging from the presence of the entire system.

One existing theory is that of *causal emergence* put forth by Rosas et al. [43], which builds on the idea that macroscopic observables can exert causal influence which is not observed at the microscopic level. To formalize this notion, the work considers a system's evolution over time with system measurements at time t denoted as $\boldsymbol{X}_t$. In this setting, a supervenient feature V_t is considered to be fully determined by the state of the system $\boldsymbol{X}_t$ at time t and provide no predictive power about $\boldsymbol{X}_{t+1}$, given $\boldsymbol{X}_t$, forming a Markov chain. The authors

define causal emergence as occurring when a supervenient feature V_t has irreducible causal power, such that it only has causal influence when all parts of the system are present. Thus, V_t is an emergent property if it contains causal information about the system and if this information is additional to that given by subsets of the system when considered separately. In PID terminology, this translates to a supervenient feature V_t exhibiting causal emergence if $0 < U(V_t; \boldsymbol{X}_{t+1}|\boldsymbol{X}_t) \leq S(\boldsymbol{X}_t; \boldsymbol{X}_{t+1})$. Using integrated information decomposition [50], which extends PID to settings with multiple targets, the study further characterized two forms of causal emergence, *downward causation* and *causal decoupling*. Downward causation refers to the causal influence of the irreducible supervenient feature on individual parts of the system, whereas causal decoupling refers to the influence on other collective properties (i.e., other supervenient features). Using its formalized definitions, the study analyzed two systems considered to be primary examples of emergent behavior, namely Conway's Game of Life (GOL) [51] and Reynolds' flocking boids model [52]. Indeed, emergent features were found in both settings: as particle collisions in GOL and as flock dynamics in simulated birds. Furthermore, the authors examined electrocorticogram (ECoG) and motion capture (MoCaP) data from Japanese macaque monkeys performing a reaching task [53], finding that the representation of motor behavior is an emergent feature of cortical activity.

Varley and Hoel [54] proposed a framework for measuring causal emergence by comparing different dimensions (macroscale versus microscale). They introduced a *partial information spectrum* to compute the proportion of total mutual information in all PI atoms at a given height in the PI lattice, allowing for the creation of a distribution across the hierarchy of the lattice. Intuitively, a 'top-heavy' distribution would yield a higher degree of *synergy bias* (lower degree of *redundancy bias*), as the majority of information about the target is present synergistically and vice versa. To assert the claim that information conversion occurs at different scales, the study analyzed three logic gates (AND, OR, XOR) at a macroscale (the gates as defined) and microscales (the gates divided into collections of gates with simpler mechanisms) in terms of their respective synergy bias, observing that synergy bias increased with dimension reduction. The authors also revealed the same phenomena in Boolean network systems where mutual information was held constant at different scales, providing evidence for a conversion of redundant to synergistic information, rather than a removal of redundant information due to compression. The work related these findings to *effective information* (EI) [55], a measure of causal emergence which was found to increase with dimension reduction in previous research [56]. Finally, the authors concluded that causal emergence can be seen as a conversion from redundant to synergistic information, with the entropy of transitions being translated to causally-relevant EI through compression.

3.3 Neuroscience

Information theory provides a strong framework for studying the brain and has been leveraged widely in the field of theoretical neuroscience. Its general applicability has allowed developing research in information theory to compliment and improve upon our understand-

ing of the human brain. One area of interest is that of complex high-level cognition, such as numerical cognition, reasoning, memory, and decision-making. Although it is accepted that such processes occur in the frontal and parietal regions of the brain, it is still unknown exactly how they proceed. In machine learning, answering this question could potentially have wide applicability for improving upon current systems to create algorithms that are better suited for performing complex tasks and being able to generalize as humans do.

Applying PID to data from several methodologies including fMRI, PET, cytoarchitectonics, and genetics, Luppi et al. [6] found that synergistic information is central to complex human cognition. In the study, the authors revealed that synergy is higher in regions of the brain [default mode network (DMN), fronto-parietal executive control network (FPN)] that are partially responsible for high-level cognition. Correspondingly, redundancy was found to be higher in areas (primary motor, sensory, and insular cortices) responsible for perception and low-level cognition. They also found that the human brain contains more synergistic information than non-human primates and that brain regions with high synergy underwent the largest cortical expansion during evolution. Furthermore, human genes were seen to enhance synaptic transmission, which facilitates synergistic interactions in the brain.

Another cognitive process that has been studied using information theory is that of consciousness. Currently, there does not exist one single widely-accepted theory of consciousness. Thus, a central goal of the field is to work towards a theory that can provide a strong explanation for experimental observations. In the realm of consciousness science, there exist several proposed theories of which are part of ongoing research [57]. *Global Neuronal Workspace Theory* (GNWT) [8, 58, 9, 10] is one influential theory, which explains consciousness as being a 'global workspace' for the brain that focuses its attention on salient neural information such that it becomes available for conscious access and broadcasts that information back to the rest of the brain. Another prominent theory is that of *Integrated Information Theory* (IIT), first detailed by Giulio Tononi in 2004 [11, 12, 13]. A notable aspect of IIT is that it provides a formal mathematical explanation and quantification for a system's level of consciousness and its content. It builds upon the premise that conscious experience is a result of the integration of information from multiple elements (the whole system being more than the sum of its parts) using information theory. Importantly, both theories address different aspects of the integration of information into the emergent feature of consciousness, and may in fact be complementary [7].

Central to IIT is the quantification of *integrated information* (Φ), or the causal information that exists in a system beyond that generated by its parts (irreducible information). Broadly, IIT postulates that the level of an experience (how 'conscious' a system is) can be described by its maximum irreducibility Φ. Although IIT is a useful starting point for developing a theory of consciousness, it has it limitations and critiques [59, 60]. In parallel streams of work, there has been effort to address some of these issues using PID. In particular, previous versions of IIT's Φ [12] have been connected to synergy, as both measures serve

to quantify collective information about a system that can not be captured by its parts alone.

Since IIT's formation, several measures of integrated information have been introduced. To provide a unified explanation of the behavior of some of these different values, Mediano, Seth, and Barrett [61] compared six candidate measures in complex networks. Notably, none of the measures used were found to consistently agree across all analyses performed. Evidently, this lack of agreement makes empirical work difficult to evaluate and interpret across different measures. To address these problems, Mediano et al. [62] proposed *integrated information decomposition* (ΦID), merging principles from both IIT and PID in an effort to overcome limitations of both frameworks. This was motivated by the argument that IIT only considers information transferred between parts of a system, ignoring high-order interactions (as provided by PID's synergy).

Related to Ref. [6], Luppi et al. [7] used PID in the context of human consciousness and its dependence on the global integration of information as suggested by GNWT and IIT. Using results from fMRI data, the authors identified a 'synergistic workspace' (comprising of the DMN and FPN) characterized by synergistic interactions in the brain, effectively linking GNWT and IIT. Furthermore, their study showed that synergy is reduced in this workspace during states of unconsciousness (anaesthesia, severe brain injury) and is restored upon recovery. Studying how and why synergy emerges, as we do in this work, could also provide better insight as to how and why consciousness emerges.

Chapter 4

Methodology

4.1 Logic Gate Network Experiments

To preface our work in the Animal-AI Environment, we first sought to observe synergistic and redundant interactions in feedforward networks with the same architecture as those used in our RL models. The primary goal of this experiment was to observe how information-theoretic measures behave depending on the nature of the task and flow of information (subject to dropout as described below), and support our method of computing redundancy and synergy (discretization method, using 2^{nd}-order measures). These experiments also provided baseline measures before the addition of complexity provided by a RL setting; the experiments were easier and faster to run so that additional analysis could be performed if needed and provided a starting point for interpreting later results.

4.1.1 Task and Architecture

We trained small feedforward networks with 20 neurons (2 layers with 10 neurons each) on learning a unique (UNQ) or exclusive-or (XOR) logic gate with two sources. For the two types of logic gates learned, the data consisted of a two-dimensional binary input and a binary output corresponding to the appropriate logic gate result, as displayed in Tables 4.1 and 4.2.

Table 4.1: UNQ Logic Gate.

Data	Label
00	0
01	0
10	1
11	1

Table 4.2: XOR Logic Gate.

Data	Label
00	0
01	1
10	1
11	0

We used networks with stochastic training methods, which could interrupt the transfer of

information, to study the resulting patterns of information decomposition. In particular, we used *dropout*, a form of regularization [63]. Dropout is used to prevent overfitting in neural networks by randomly omitting neurons (with some given probability) during training. This technique has the effect of approximating ensemble methods, which train many different neural networks in parallel. Dropout creates a 'new' model with each successive removal of a neuron, as the passage of information is disrupted, allowing for gradient updates to be performed according to many different models within the network. This also adds noise to the training process, which forces the network to be robust rather than individual neurons becoming strongly dependent on each other. Interestingly, dropout has also been shown act as a form of Bayesian approximation which can be used to evaluate model uncertainty [64]. In our work, dropout provides a convenient method to study the way stochastically disrupting information transfer may affect information representations and provide more intuition about the role of different forms of information.

We specifically chose networks with small layer sizes due to the computational constraints of calculating PID values on a large number of sources (i.e., greater than 10). Additionally, we used networks with the same architecture as the networks used in our RL models (section 4.2.1) for the purposes of consistency. Each group of networks had a varying level of dropout applied ($p = \{0.0, 0.1, 0.3, 0.5\}$) after each linear layer. We additionally used a rectified linear unit (ReLU) activation function, which constrains its output from $[0, \infty]$, to simplify the discretization of the network activations as required by our synergy and redundancy measurement method. Additionally, the use of a ReLU function avoided the compression of mutual information associated with a double-saturating non-linearity (such as tanh) as addressed in Ref. [34] (discussed in Section 3.1).

4.1.2 Information Decomposition Calculations

After training, we sampled the activations of the networks during testing, with network weights frozen. Each activation value was discretized using 3 bins from $[0, 5]$ to ensure a sufficient number of samples in each sources-target pair. The binning range was chosen from empirical observations of the network activations being heavily concentrated within the range of $[0, 5]$.

We used two different measures from the literature to compute synergy and redundancy, namely $I_{\min}$ [5] and I_{MMI} [44], as described in Section 2.3.2. These methods were favorable due to their relative efficiency and tractability in larger systems compared to others, being closed-form expressions for synergy which do not require the computation of the entire PID lattice.

First, the discretized sampled data was used to calculate a probability distribution over the set of sources and target by simply counting the number of occurrences of each joint sources-target state and using the so-called plug-in estimator [65]. We then used Ref. [66],

a package made for computing discrete information theory measures, to create the distribution used for our calculations. In this setting, each source may correspond to a neuron in a layer of the network or to a set of dimensions/single-dimension of the input. Alternatively, the target can correspond to either a subsequent layer in the network or the output of the network. With the distribution created, redundancy and synergy can be computed using equations Equations (2.36), (2.37), (2.39) and (2.40) for both measures.

PID values of a system can also be computed over subsets of k cardinality, denoted k-order values. For example, a k-order synergy measure can be calculated by using only k elements of a system as sources, rather than the entire set of elements. Performing this operation over each combination of k elements in the system and taking the mean then gives the average k-order synergy. This is useful because studying subsets of k sources provides the ability to observe information representations at different scales.

Using the same methodology as the full-order case, information decomposition values were also computed at the k-order. In this setting, we computed the same values over each combination of 2 neurons in the source layer with the same target and took the mean to give the average 2-order synergy for the given source layer and target. These values were computed in parallel with the values computed for full-order.

4.2 Animal-AI Experiments

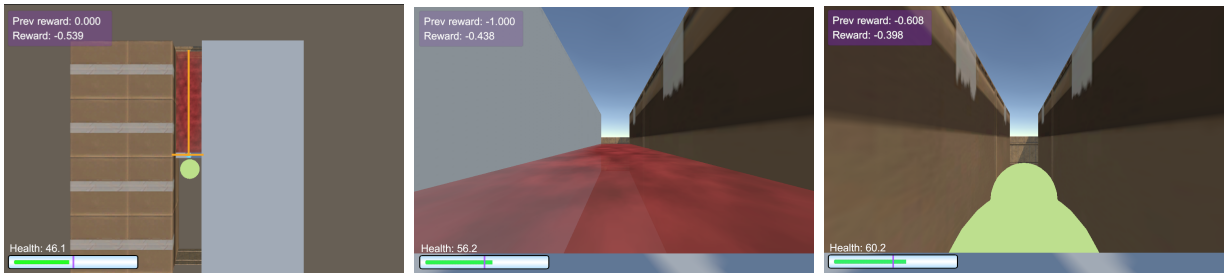
We next extended the idea of logic gates to the context of RL agents in the Animal-AI Environment. Of particular interest were the questions of whether tasks which require integration of multiple sources of information (synergistic) are more difficult to perform than tasks which do not, and whether the amount of integration of information required for the task is correlated with the amount of synergy in an agent's network. Another point of interest was the study of synergy in task-transfer and in the addition of complexity through compound tasks. We also were interested in observing the synergy dynamics within the RL networks as compared to the networks from our first experiment.

4.2.1 Task and Architecture

We created several tasks to mimic 2-bit logic gates and made various extensions. The basic design of the configurations consists of an agent placed on a platform with a pit in front of and behind it, and left and right barriers enclosing the space (fig. 4.1). The barriers' object type corresponds to the logic gate input it represented (a wall object representing a '0' and a cardbox object representing a '1'). Additionally, the output of the logic gate corresponds to each pit: in front of the agent being '0', behind the agent being '1'. Within the pit corresponding to the correct output of the logic gate lies an occluded positive goal with a reward of 4, and within the pit corresponding to the incorrect output lies an occluded 'death zone' with a reward of -1 which immediately terminates the episode. Thus, the agent

is constrained to movement on the platform and can only successfully complete the task if it uses information relayed by the bit-representing objects. Due to the larger action and state space of the agent, this initial task still differs from the prior logic gate tasks in experiment 1; for example, as an alternate solution, the agent could potentially learn to rotate 90 degrees on the platform and move forward and backward, without reaching either the positive goal or the negative death zone. Using this setup, we designed a set of four configurations (combinations of 2 bits) for each gate, which were iterated through for each episode during training.

Notably, the observation space was constrained to three object-oriented raycasts, each detailing the type of object hit by the ray projected from the agent and its normalized distance, and a vector relaying information about the agent's health, velocity, and global position. The rays were projected 90 degrees apart – directly in front of the agent, directly to its left, and directly to its right – ensuring that the agent had all the information required to solve the task at initialization (fig. 4.1a). Additionally, the raycast observation space occluded the positive/negative rewards in either pit which would have otherwise been visible to an agent receiving the full pixel space. The small observation space also facilitated the ability for our models with small network parameter spaces to solve the task and represent the input information adequately. Additionally, the small networks being used could more reliably perform the task (as opposed to receiving pixel image inputs), without the need for additional convolutional layers.



(a) (b) (c)

Figure 4.1: Example Animal-AI configurations for the 2-bit XOR task. The observation space is constrained to raycasts directly in front and to the left and right of the agent, occluding pixel information about the contents of the pit (i.e., the reward and the death zone are not visible to the agent on the platform). **(a)** displays an aerial view of the '10'-input XOR configuration, where the left cardbox barrier represents a '1', the right wall barrier represents a '0' , and the backward-relative-to-agent position of the green reward represents a '0' output of the gate. The orange lines represent the orientation of the raycasts projected from the agent. Similarly, **(b)** displays the agent's perspective for the '01'-input XOR configuration, whereby the '0' gate output is in the pit behind the agent on the platform. Finally, **(c)** displays the agent's perspective for the '11'-input XOR configuration, whereby the '1' gate output is in the pit in front of the agent on the platform.

We trained PPO models [67] with small feedforward actor-critic policy networks (2 layers with 10 neurons each) with ReLU activation functions between each linear layer, identical

to those used in experiment 1 for dropout $p = 0$.

2-Bit XOR and 2-Bit UNQ Task

Using the aforementioned logic gate base design, we created 2-bit XOR and 2-bit UNQ tasks in the Animal AI-Environment as a baseline extension from the logic gate networks used in experiment 1. As shown in Figure 4.1a, agents were placed on a short platform in an effort to restrict the state space and simplify the task as much as possible to measure the construct of interest – the minimal baseline of solving of the UNQ and XOR logic gates in a RL setting.

3-Bit XOR Task

Interested in the effect of integrating an increasing number of sources, we additionally designed a 3-bit XOR task. In particular, we created a set of eight configurations for a 3-bit XOR gate by placing a barrier directly in front of the agent at the end of the pit, treating it as an additional input source. Our initial use of three raycasts in the baseline 2-bit tasks allowed for a simple extension to 3-bit by ensuring each input bit could be observed from initialization in an effort to prevent additional bias/exploration that could be introduced by one input bit being occluded (which would also require integrating input information over time).

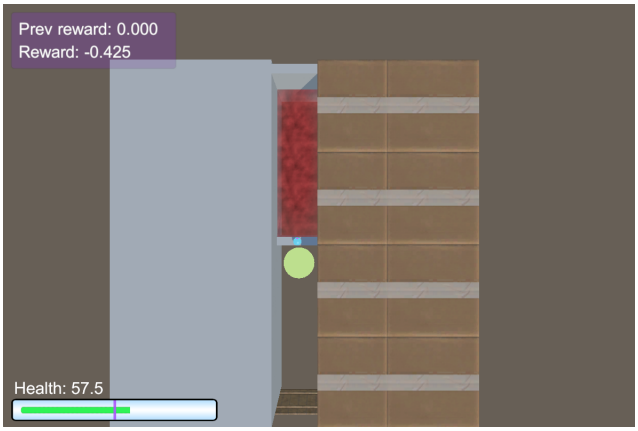


Figure 4.2: 3-bit XOR task configuration for an input of 001 and an output of 1. See caption of Fig. 4.1 for details.

2-Bit Distance XOR Task

Another point of interest was that of creating a compound task by pairing the synergistic XOR logic gate task with an additional non-synergistic task. One of the simplest ways to explore this idea was to increase the distance between the agent and the reward. While such a modification of the task may seem trivial, it significantly increases the state space of the agent and potentially allows for the off-loading of synergistic information about the input

into a particular trajectory based on global position. To create the task, we elongated the platform from length 1 to length 10 such that the agent would have to travel further on the platform before falling into the pit yielding either the reward or the death zone.

Curriculum Tasks

After observing each aforementioned task trained on individually for each agent, we sought to analyze the ability of agents to generalize to more complex tasks and the effects of task-transfer on the amount of synergistic information at different points during training. As such, we employed several curriculum-style experiments. In each, agents were trained on the baseline 2-bit XOR task until first reaching either a maximum reward threshold during intermittent testing or a maximum number steps of training. Agents were then additionally trained on another task (3-bit XOR or distance XOR) for a particular number of steps. We then tested agents and computed synergy measures at several different points based on the observed training curves, as shown in Figure 4.3a. In particular, we defined these checkpoints as *configuration 1 initialization* (1I; the random initialization of the network prior to any training), *configuration 1 threshold* (1T; the point at which the model successfully reached the maximum reward threshold during evaluation), *configuration 2 adaptation* (2A; 10,000 training steps into in the second configuration), *configuration 2 recovery* (2R; the estimated empirical point in which the mean training reward stabilizes and 'recovers' from its initial decrease at the point of configuration change), and *configuration 2 end* (2E; the final model at the end of training on the second configuration). Each checkpoint was tested using the same full-set of configurations part of the training curriculum in order to standardize synergy measures and ensure that they were comparable, rather than an artifact of the configuration tested on (i.e., agents tested at each checkpoint were tested on the entire curriculum regardless of whether or not they had been trained on the configuration yet).

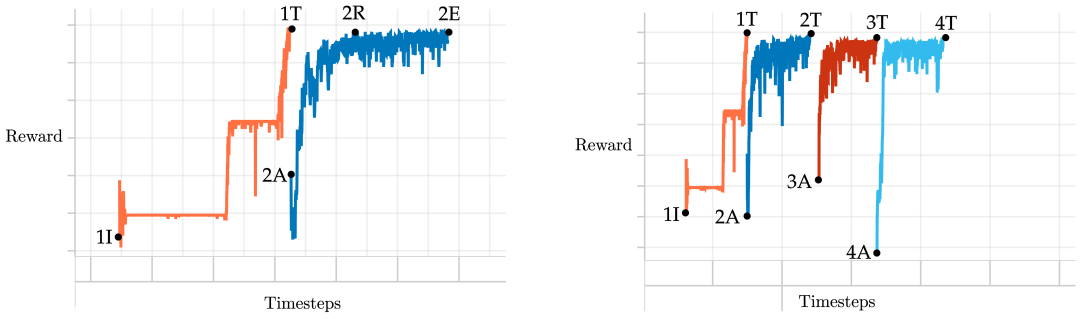


Figure 4.3: Curriculum training synergy checkpoints for **(a)** curricula with two configurations (2-bit to 3-bit XOR curriculum and 2-bit to 2-bit distance XOR curriculum), and **(b)** curricula with four configurations (increasing distance curriculum).

Finally, to study transfer over several consecutive increases in complexity, we extended the curriculum method over several configurations, increasing the platform distance to the

reward in each subsequent configuration (lengths 1, 10, 20, 30). This was done in an effort to observe the training and synergy dynamics with the repeated requirement to generalize and either reuse or relearn information representations. Again, agents were trained on each configuration until first reaching either the maximum configuration reward or the maximum number of steps, before moving to the next task in the curriculum. The models were then evaluated and, as shown in Figure 4.3b, synergy was computed at random initialization prior to training (1I), the threshold point for each configuration (_T), and the initial training points 10,000 training steps into each configuration excluding the first (_A).

4.2.2 Information Decomposition Calculations

We used the same method for computing synergy as for experiment 1. After training we sampled the activations of the actor networks during episodes of testing on each configuration trained on and discretized their values using 3 bins from $[0, 5]$. We used the same measures from experiment 1 of $I_{\min}$ and I_{MMI}, discretized via binning, for computing synergy over the actor networks. Similarly to experiment 1, these values were computed between each layer and the subsequent layer. Because the observation space exceeded 20 dimensions, the proposed synergy measures could not be efficiently computed over the entire input and instead had to be approximated by grouping larger subsets of the input space into separate sources and computing average 2-order synergy measures. Additionally, the modularization of the input made it possible to treat rays containing information about individual objects as single sources and allowed for the isolation of information relevant to the task. Thus, each object-oriented raycast was treated as a single source with its dimension distance discretized using 3 bins from $[0, 1]$; the global position was also used as a single source and discretized using 5 bins from $[0, 40]$ (40 being the length of the arena). The 2-order synergy was then computed over each pair of raycast sources and each pair of single raycast source and the position source, with the target being either the first layer of neurons. We additionally computed the mean 2-order synergy for the other layers in an effort to keep our comparisons across layers consistent (both input and layer sources being 2-order values). This was also supported by our observations in experiment 1 and previous findings that smaller networks concentrate synergy at smaller scales (synergy is less-widely distributed across the network), as compared to larger networks [45].

Chapter 5

Logic Gate Network Results

5.1 Introduction

For both UNQ and XOR logic gates, we trained an ensemble of 10 networks per dropout probability, with each network initialized separately using a different random number generator seed. We then quantified the redundant and synergistic information between each layer source, including the input, and its subsequent layer target, as well as between each layer source and the output target of the model. We also repeated the same calculations with averaged 2-order measures. Synergy and redundancy were computed using both $I_{\min}$ and I_{MMI} measures and found to agree in ordering and pattern across the network. Thus, for display purposes, only $I_{\min}$ values are shown and discussed in the remainder of this section; figures with I_{MMI} measures can be found in the appendix. Each measure is reported in bits. Using varying levels of dropout during training allowed us to observe how the networks represent information about the input and how they adapt when these representations are stochastically interrupted by the removal of a set of neurons, disrupting the flow of information. The addition of dropout in this context can be seen as a method of regularization, forcing the network to become more robust in its representations by encouraging redundancy of relevant information and by pruning irrelevant information.

To preface our analyses of results, we first provide a guide for reading the figures in this section. Each plot shows the mean [$\pm$ the standard error of the mean (SEM)] measurement (2-order redundancy, full-order redundancy, 2-order synergy, or full-order synergy) of the ten networks trained for the specific logic gate task (UNQ or XOR) for each dropout probability ($p = 0.0, 0.1, 0.3, 0.5$). Furthermore, plots are also distinguished by the target used in computing the measurements: either the subsequent layer (labeled 'Layer Target') or the output (labeled 'Output Target'). For example, in plots termed 'Layer Target', the measurement is computed between (1) the input sources and first layer target; (2) the first layer sources and the second layer target; and (3) the second layer sources and the output target. Instead, in plots termed 'Output Target', the measurement is computed between (1) the input sources and the output target; (2) the first layer sources and the output target; and (3) the second

layer sources and the output target. In all plots, the line between discrete data points is not representative of any additional information and is only used as a visual aid.

5.2 Analysis of Redundant Interactions

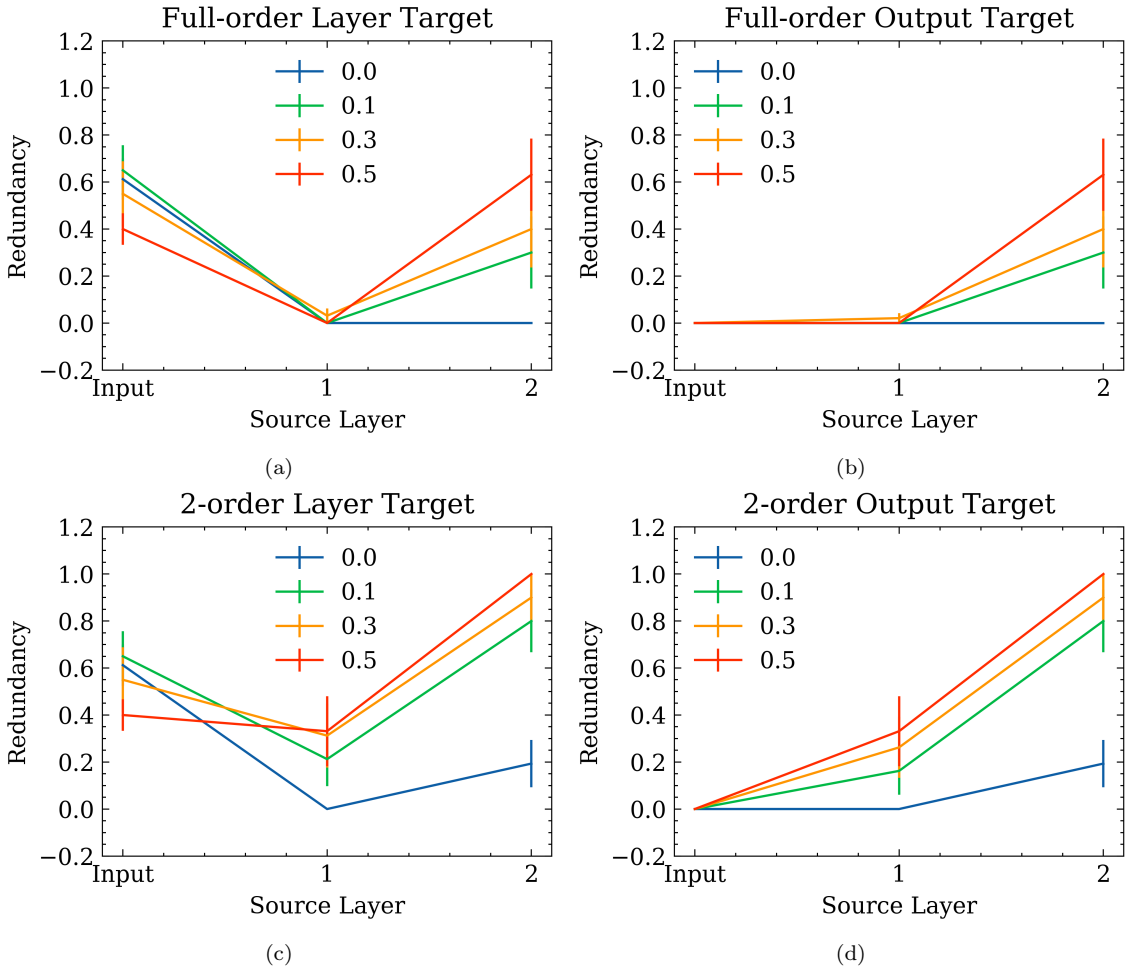


Figure 5.1: UNQ network redundancy for varying levels of dropout. **(a)** Redundant information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order redundant information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

In Figure 5.1, we show the redundancy measurements for the UNQ logic gate networks for varying levels of dropout. We note several observations and interpretations. First, as dropout probability increases, redundant information from the input source to the first layer target decreases in both the full-order (fig. 5.1a) and 2-order (fig. 5.1c) case. This can be explained by dropout effectively pruning unimportant redundant information about the input (i.e.,

the input bit not corresponding to the gate output), as its representation is not needed to successfully solve the task.

Then, in the first layer sources to the second layer target, redundancy drops to approximately 0 in the full-order case (fig. 5.1a), suggesting a transformation of full-order redundant information into another form. We argue that this redundant information is compressed into unique information at the full-order scale, due to the relative inefficiency of representing information redundantly across many neurons instead of a single neuron or a subset of neurons. We support this finding with additional results later in this section.

Rather than collectively decreasing as in the full-order case, in the 2-order case (fig. 5.1c), redundancy from the first layer sources to the second layer target increases as dropout increases. In this setting, dropout encourages the pruning of redundant information about the input that is unimportant for the task and later increases low-order redundant representations about the information that is important to resist the effects of dropout. This is sensible, as dropout can turn off any set of neurons, resulting in important information, which could otherwise be potentially lost, to be over-represented redundantly. The higher presence of redundancy at a smaller-scale (2-order rather than full-order) also supports our assertion that networks favor the compression of redundant information about the input (predominantly to unique forms); representing redundant information across two neurons is more efficient (and compressed) than across ten neurons.

Finally, in both the full-order (fig. 5.1b) and 2-order (fig. 5.1d) settings, the redundancy from the second layer sources to the output target increases as dropout increases; additionally, the 2-order case exhibits the same pattern in the first layer sources. This is expected, as a redundant representation for the output is favorable due to the perturbations caused by dropout. Again, the same pattern of higher levels of 2-order redundancy compared to full-order redundancy is replicated, supporting our hypothesis of compression into unique information/lower-order representations. We further note that the observed patterns of redundancy in terms of relative-ranking and pattern across the network tend to mostly agree across full-order and 2-order measures, supporting the use of smaller-order measures for approximating higher-order information decomposition.

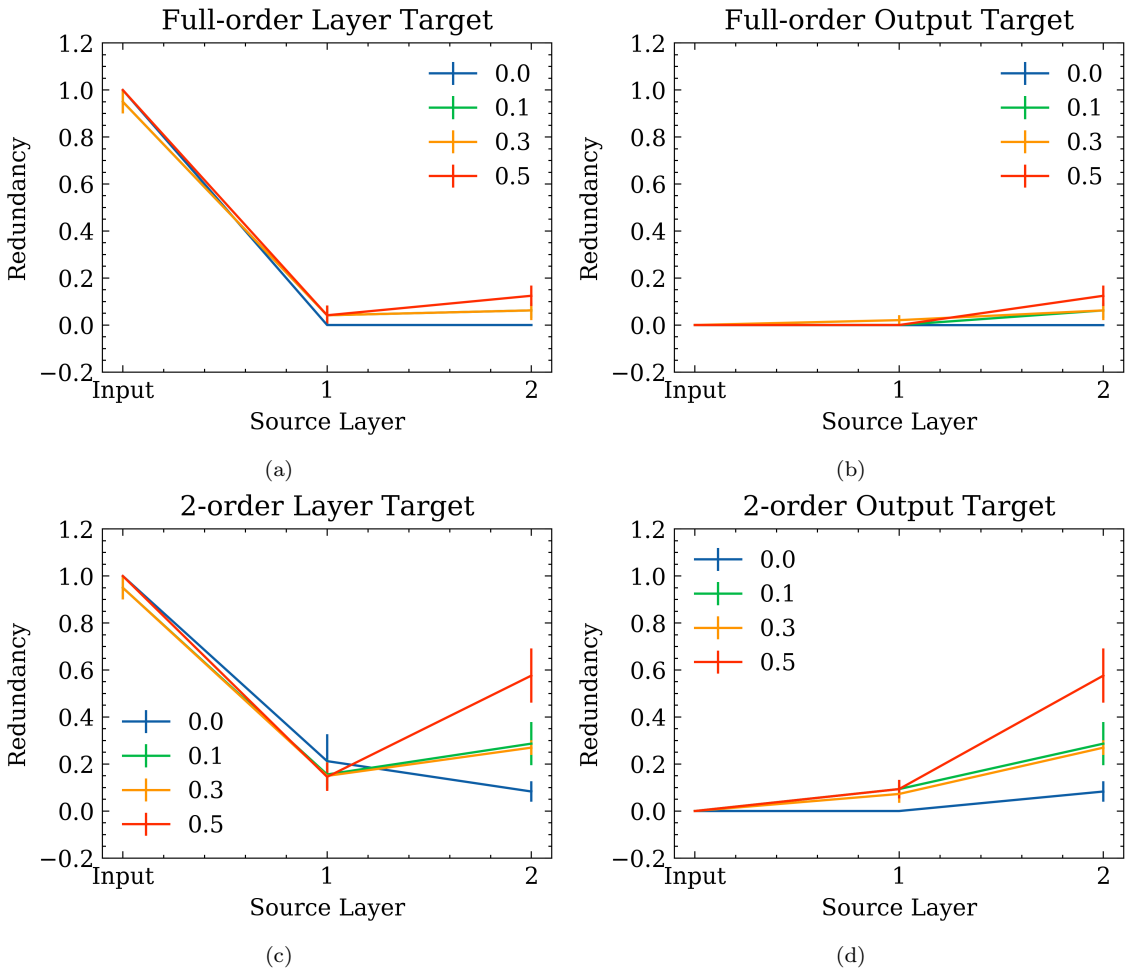


Figure 5.2: XOR network redundancy for varying levels of dropout. Redundant information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order redundant information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

Figure 5.2 displays redundancy measures for the XOR networks with varying levels of dropout. We see that redundant information from the input sources to the first layer target are high across all dropout probabilities for both the full-order (fig. 5.2a) and 2-order case (fig. 5.2c). Unlike the UNQ networks, dropout does not lead to pruning of redundant information about the input because all of it is necessary for solving the XOR gate. Thus, redundancy is preserved in cases where it important for learning the task.

The same drop in redundancy from the first layer sources to second layer target observed in the UNQ gate occurs in the XOR gate for the full-order case (fig. 5.2a), again supporting our interpretation of the compression of redundant input information occurring in the network. Interestingly, unlike the UNQ gate networks, there does not appear to be a strongly distin-

guishable pattern of redundancy related to dropout at the 2-order level (fig. 5.2c). Instead, 2-order redundancy appears to be reduced to the same approximate magnitude ($\sim$0.2 bits) across networks. One speculation for why this occurs is that all networks learn to represent information about the input similarly, despite the amount of dropout applied, due to the need to retain and process all information about the input. Thus, all networks retain information about the input redundantly and then compress information to 2-order and unique information in similar ways.

As in the UNQ networks, increasing levels of dropout also increase the redundancy from the second layer sources to the output target in both full-order and 2-order cases in the XOR networks (figs. 5.2a and 5.2c). We would expect the magnitude of redundancy in the XOR networks to be similar that in the UNQ networks – in this setting, redundant information appears to be used to resist the loss of important information due to dropout. Interestingly, in the XOR networks, both full-order and 2-order redundancy is significantly smaller in magnitude than the UNQ networks. We do not have a strong understanding as to why this occurs and further investigation is needed to better understand these dynamics.

This observation also parallels the fact that redundant information about the output target is also significantly reduced in magnitude for the XOR networks (Figures 5.2b,d) compared to the UNQ networks (figs. 5.1b and 5.1d). We finally note that, as in the UNQ networks, full-order and 2-order redundancy measures in the XOR networks are relatively consistent in terms of relative-ranking and pattern across the network, again supporting the use of smaller-order measures to approximate higher-order information decomposition.

5.3 Analysis of Synergistic Interactions

Similar to our measures of redundancy, we also study the patterns of synergy across the UNQ and XOR networks in Figures 5.3 and 5.4. Consistent with our finding that dropout facilitates the pruning of unimportant redundant information about the input in UNQ networks (figs. 5.1a and 5.1c), we see that irrelevant synergistic input information is also removed (figs. 5.3a and 5.3c). Intuitively, synergistic information is less favorable with high levels of dropout, as the removal of any subset of neurons results in a loss of distributed information. Thus, in the case of UNQ networks where synergistic information about the input is not needed for the completion of the task, this information is selected against.

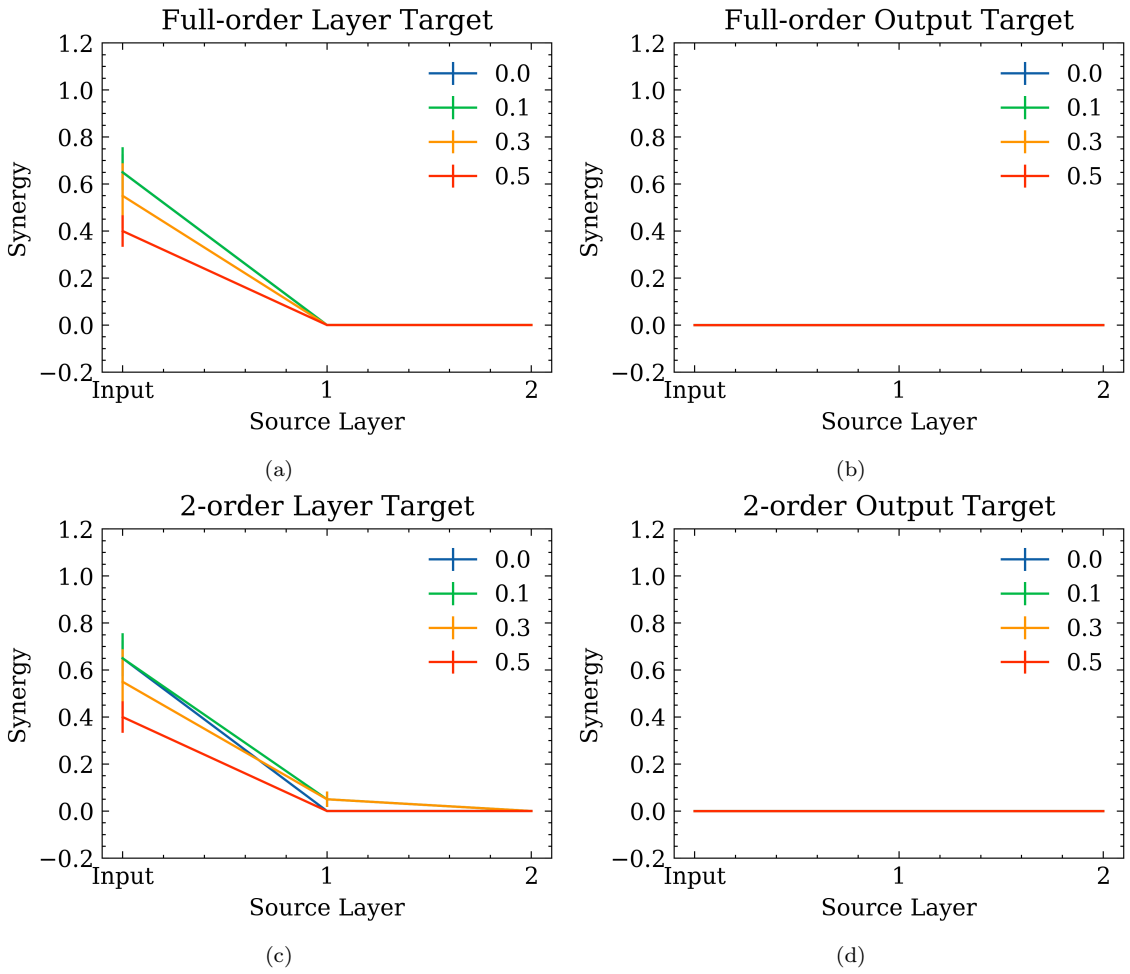


Figure 5.3: UNQ network synergy for varying levels of dropout. **(a)** Synergistic information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order synergistic information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

In Figure 5.3, we observe that 2-order and full-order synergy drop from the first layer sources to second layer target in UNQ networks. One may attribute this decrease in synergy to the setting of dropout, as it clearly would not be a favorable representation within such networks. However, this pattern is present in networks without dropout applied ($p = 0.0$), meaning that all of the networks actually compress synergistic information into other forms, regardless of whether dropout is applied. This parallels with the observation from (fig. 5.1), whereby full-order redundant information about the input drops in the first layer sources. Thus, the mutual drop in full-order redundant and synergistic information in UNQ networks implies an increase in full-order unique information. Furthermore, the drop in 2-order synergistic information (fig. 5.3c) also implies an increase in 2-order unique information, as 2-order redundant information (fig. 5.1a) does not increase. These observations further support our

finding that synergistic and redundant information is compressed into unique information when possible and favorable, due to its increased efficiency as a representation.

Finally, there is no synergy in the second layer sources to the output target in UNQ networks, as shown in Figure 5.3. This is again consistent with the assertion that synergy is an inefficient representation, which the network compresses into other forms.

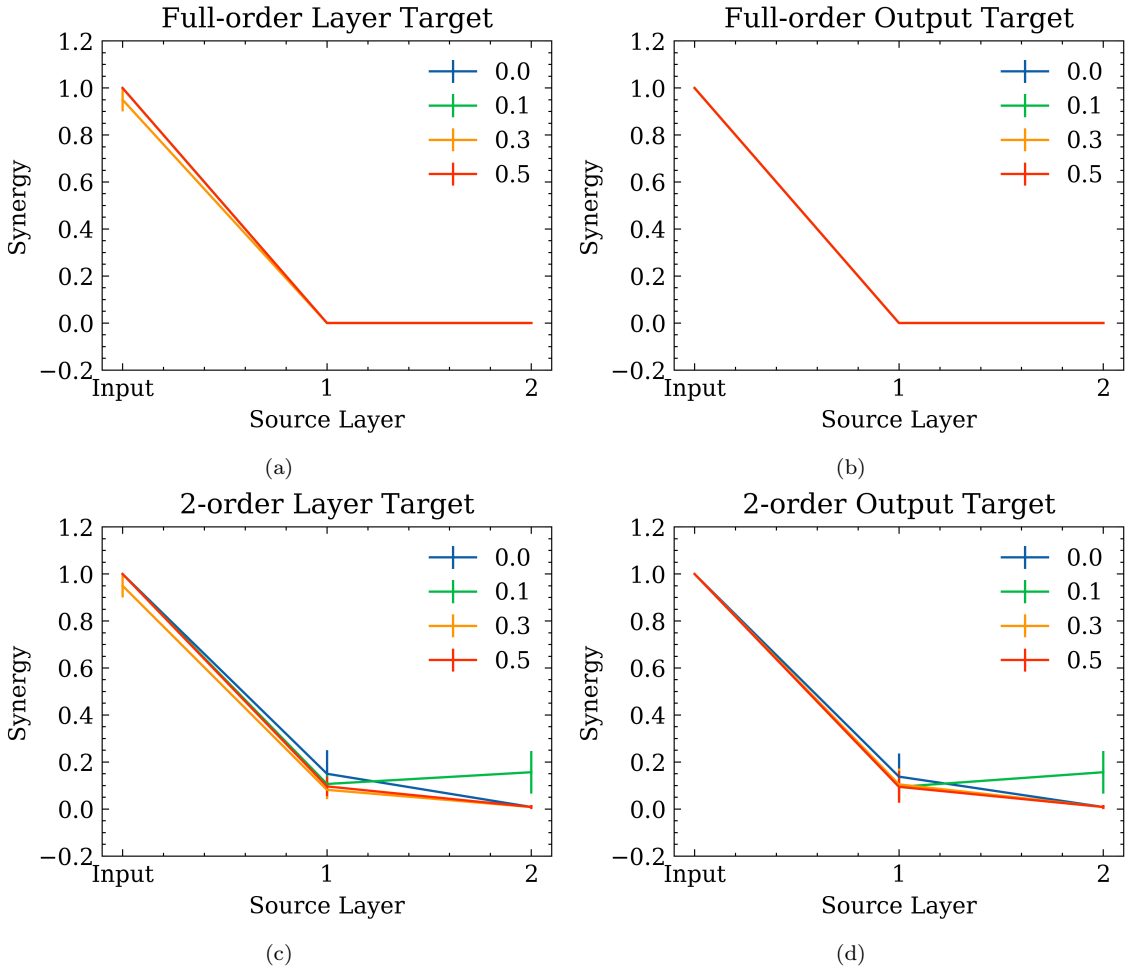


Figure 5.4: XOR network synergy for varying levels of dropout. **(a)** Synergistic information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order synergistic information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

Just as there is a high level of redundant information from the input sources to first layer target for the XOR gate (fig. 5.2), there is also a high level of synergistic information (fig. 5.4). This is due to the fact that the XOR gate requires complete information about each input in order to be successfully predicted, yielding maximum synergy. Thus, synergistic in-

formation about the input is crucial. We see that dropout does not prune this information because of its importance. We see that, compared to the UNQ gate, the XOR gate elicits more synergistic information in the network. This supports our hypothesis that the level of synergy in neural networks is influenced by the task performed by the model; thus, tasks can be compared by the synergy they elicit, resulting in ones that are more synergistic (i.e., XOR gate) or less synergistic (i.e., UNQ gate).

The same trend of synergy compression previously observed is shown in Figure 5.4 in source layers 1 and 2, as synergy drops to minimal values. At the full-order level, this also corresponds with drops in redundancy (fig. 5.2), meaning there is an increase in full-order unique information.

Finally, both Figures 5.3 and 5.4 show a consistent correlation between full-order and 2-order synergy, in terms of magnitude, relative ranking, and cross-layer dynamics. We use this as support for our use of 2-order synergy in experiment 2. Importantly, this finding shows that studying networks of our architecture (2 feedforward layers of 10 neurons each) at the 2-order level can provide relevant and important insights about the over-arching dynamics and that synergy is concentrated at the 2-order level. This is also consistent with the findings of Ref. [45] which found that the maximum amount of synergy exists in groups of two (2-order) in small networks of 20 parameters, and becomes more distributed (larger-order groups) as network size increases.

5.4 Summarizing Remarks

This first study was successful in revealing a number of information patterns for which we provided an interpretation for. Our experiment suggests the following claims.

First, tasks requiring the integration of multiple sources of information (XOR gate) yield higher levels of both synergy and redundancy in the input to the first layer compared to tasks which do not require integration of multiple sources (UNQ gate), as a result of the network's pruning during training for the purposes of compression. Thus, the information decomposition of a network is highly influenced by the task it performs and synergistic tasks (XOR gate) elicit more synergy than non-synergistic tasks (UNQ gate). This finding is crucial for the purposes of our study, as we base the remainder of our experiments on the fact that certain tasks elicit more synergy than other tasks (synergy is a function of task).

Second, if a network has the capacity and need to fully represent synergistic or redundant information about the input within the neuronal space, it transforms that information to unique information; thus, both synergy and redundancy are concentrated in the input to the first layer and are substantially reduced from the first layer to the second layer. However, if a network requires a more distributed representation of information, as is the case for dropout, it still maintains that information at a smaller scale (i.e., 2-order redundancy). This is also

a significant finding, as it could provide an explanation for why redundant and synergistic measures appear to be generally concentrated in smaller groups rather than the full set of sources (2-order rather than full-order); this provides additional insight into the results of Ref. [45].

Third, in this setting, 2-order synergy and redundancy appear to be somewhat consistently correlated with full-order measures in terms of ordering and dynamics across the network, supporting the use of k-order approximations for systems where the number sources makes such computations unfeasible. Additionally, 2-order PID measures consistently yield the same or higher magnitude of redundant/synergistic values, suggesting that for the given size of the network, redundant and synergistic information are less-distributed across the full layer space and instead are more concentrated into subsets of neurons. This may also be explained by our re-occuring idea of compression: in addition to synergistic/redundant information being compressed into unique information, it is also more efficient for it to be compressed into lower-order redundant/synergistic information. We use this finding to support our use of 2-order synergy as an approximation for full-order synergy in experiment 2.

Fourth, increasing dropout leads to pruning of unimportant information about a network's input, regardless of whether it is synergistic or redundant (UNQ gate), but preserves relevant information when the network has the capacity to (XOR gate).

Finally, the potential loss of neuronal information due to dropout results in higher levels of redundancy in hidden layers of the network, varying in order scale. This increase in redundancy allows for information to be over-represented such that it resists the loss of information due to the removal of a subset of neurons.

We note that our observations are limited and do not necessarily generalize to other systems or settings: they may be a feature of the complexity of the task, network architecture, the neuronal information space required to represent the input information space, and potentially other confounding factors (training method, discretization method, activation function, etc.). Further analysis is required to make any definitive claims about the nature of such information interactions within neural networks. The purpose of this study was to provide a basis for designing and interpreting the results of our next experiment.

Chapter 6

Animal-AI Results

6.1 Introduction

For each task, an ensemble of 10 PPO agents was trained, with each agent's actor-critic network being initialized separately using a different random number generator seed. We then quantified synergy measures of each model at various points during training based on their relative performance or at thresholding points. As in experiment 1, synergy was computed using both $I_{\min}$ and I_{MMI} measures and were found to agree in ordering and pattern across the network. Thus, for display purposes, only $I_{\min}$ is shown in the remainder of this section with each measure being reported in bits; figures using I_{MMI} can be found in the appendix.

To preface this section, we provide a guide for reading the following figures. Each plot shows the mean (± SEM) 2-order synergy of the 10 models trained for a particular task across different time points (as described in section 4.2.1). Furthermore, each measure uses the subsequent layer as a target ('Layer Target' as described in Section 5.1). In each plot, input sources refer to the averaged synergy of either pairs of raycasts (titled 'Pairwise Rays') or pairs of a single raycast and position (titled 'Pairwise Ray/Pos'). Finally, Figures 6.5, 6.8 and 6.10 plot the mean pairwise difference of individual agent's level of synergy at different subsequent time points. Thus, positive values indicate a mean increase in agent's synergy between two subsequent time points and negative values indicate a mean decrease in agent's synergy between two subsequent time points; x-axis labels correspond to the two time points compared (i.e., 1T-1I refers to the difference between time points 1T and 1I).

6.2 2-Bit XOR and 2-Bit UNQ Tasks

Our first task in the Animal-AI Environment extended the idea of simple logic gates in experiment 1 to RL models. We were particularly interested in observing whether our results were consistent with those of our first experiment and in creating a solvable task that could be further expanded upon. We tested the models in this task at different points during training

corresponding to their level of performance. In particular, for the 2-bit UNQ task, we tested and computed synergy values at random initialization prior to training, after successfully solving two of the four inputs to the logic gate (CKPT 2), and after successfully solving all four inputs (CKPT 4). For the 2-bit XOR task, we additionally computed synergy on the networks when 3 inputs to the gate were successfully solved (CKPT 3). We note that not all models trained necessarily reached each checkpoint – some solved several inputs in tandem, while others did not reach past a certain level of performance. As a result, Figures 6.1 and 6.2 do not display the values of all 10 models for certain time points during training, instead showing only the subset that reached the labeled performance threshold.

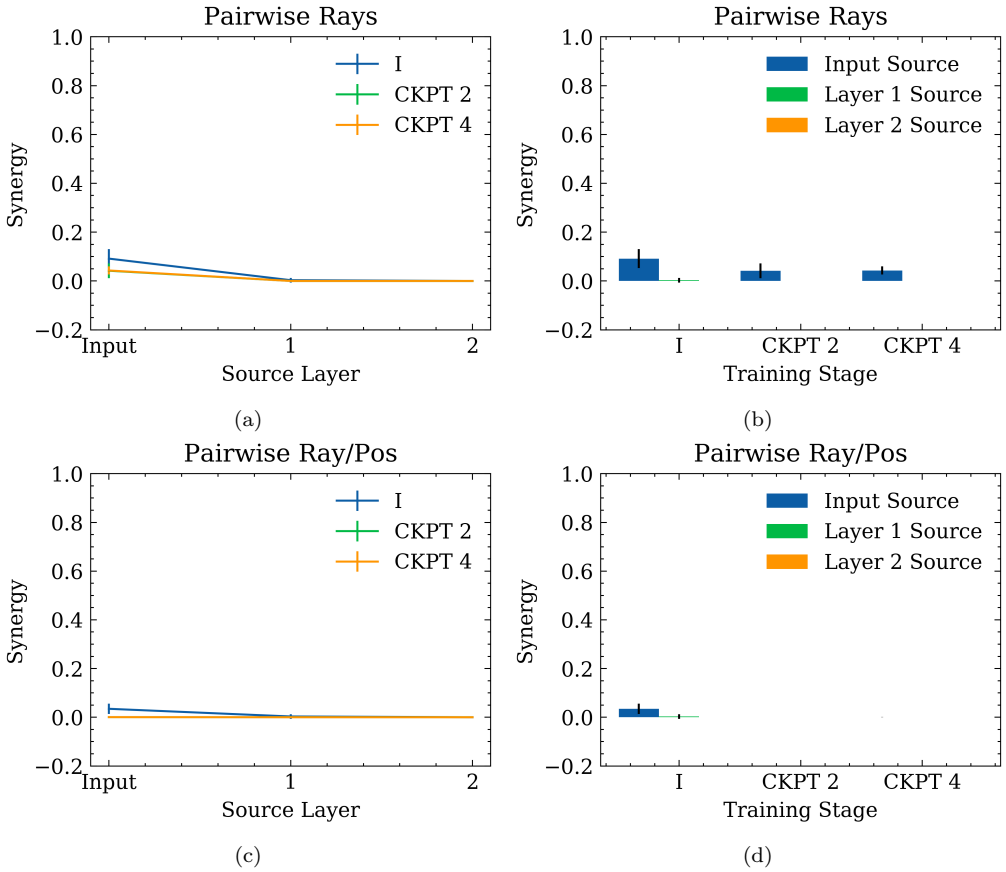


Figure 6.1: 2-bit UNQ synergy for number of inputs to gate solved. **(a, b)** 2-bit UNQ 2-order synergy between each layer source and the subsequent layer target with pairwise ray input sources, **(c, d)** with pairwise ray and position input sources. Values of layer 1 and 2 sources were too small to display on the graph.

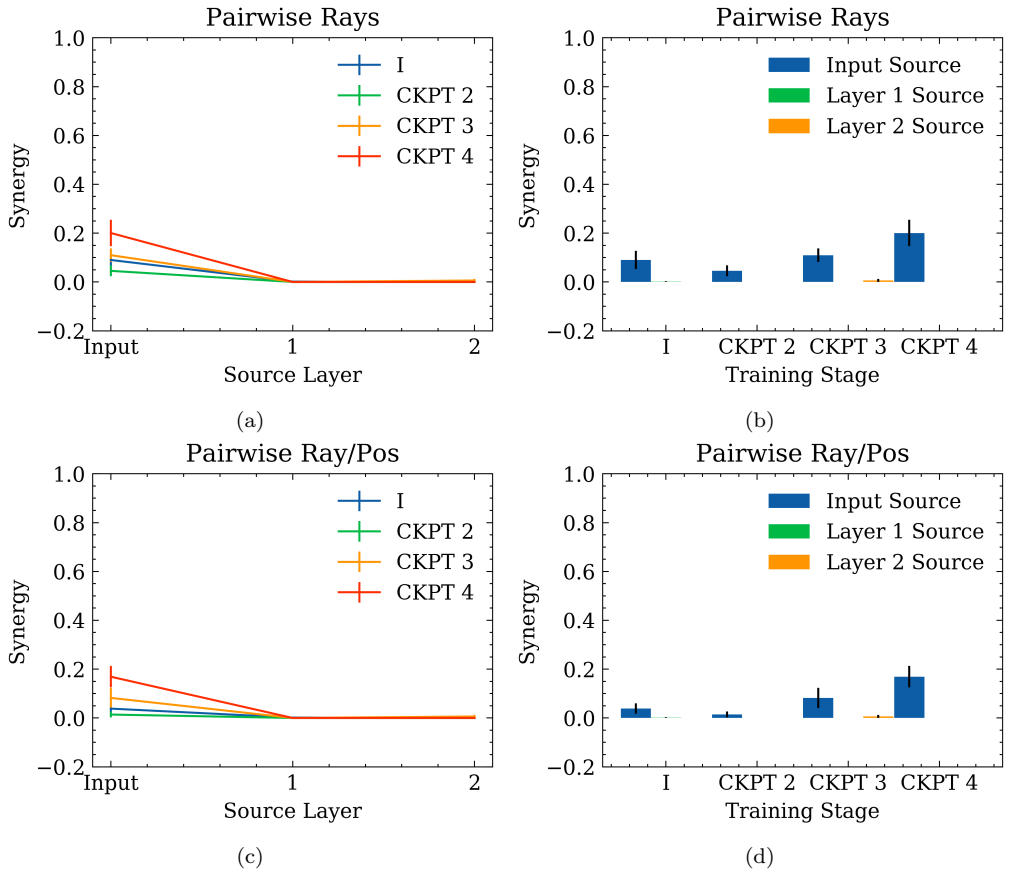


Figure 6.2: 2-bit XOR synergy for number of inputs to gate solved. **(a, b)** 2-bit XOR 2-order synergy between each layer source and the subsequent layer target with pairwise ray input sources, **(c, d)** with pairwise ray and position input sources.

As shown in Figures 6.1 and 6.2, the synergy values computed in this first set of tasks are relatively low, especially compared to those observed in experiment 1. This could be due the presence of other forms of information in the richer environment which dominate over synergistic information. Still, the results are consistent with the patterns we expect. In particular, synergistic information is concentrated from the input sources to the first hidden layer target and is then converted to other forms of information (i.e., unique). Additionally, at the point of perfect accuracy when all four gates are solved, the UNQ task yields minimal synergy and is in fact lower than the level of synergy at initialization, suggesting pruning of irrelevant synergistic information. This is consistent with the results of experiment 1, where UNQ logic gate networks, especially those with high levels of dropout, pruned unimportant synergistic information, yielding lower values of synergy compared to XOR logic gate networks. Additionally, the XOR task shows a consistent pattern whereby the synergy decreases from initialization at the time point when two gates are solved, as the model has learned the non-synergistic UNQ gate; it then increases in synergy gradually as an additional input is integrated with 3 and 4 configurations successfully being solved. Thus, synergy increases

as agents learn to integrate information from multiple sources.

6.3 3-Bit XOR Task

Following our initial experiment, we were interested in observing whether the number of sources being integrated would have an influence on the level of synergy. Furthermore, we were also curious to observe what effect, if any, curriculum training requiring the integration of an additional source of information would have on performance and synergistic information in the network. Notably, a 3-bit XOR gate can be computed using two 2-bit XOR gates [i.e., for 3 bits X, Y, Z, xor(X, Y, Z)=xor(xor(X, Y),Z). Thus, we predicted that a curriculum transferring from a 2-bit XOR gate to a 3-bit XOR gate could potentially be solved by re-using the learned representations in the input sources to the first layer target (presumably, where synergistic information about the 2-bit XOR gate is compressed into unique information to solve the gate) and learning an additional 2-bit XOR gate in the first layer sources to second layer target, yielding more synergy in later layers of the network.

6.3.1 Single Task

For our 3-bit XOR task, we used the same general method as in the 2-bit tasks, whereby we evaluated models and computed synergy based on which points they were able to successfully solve a given number of configurations. As before, not all models solved each of the shown number of gates, yielding plots which do not necessarily encompass the batch of all 10 models for certain performance thresholds in Figure 6.3. We found that the synergy values yielded by the 3-bit XOR task (fig. 6.3) were very similar to those yielded by the 2-bit task (fig. 6.2), exhibiting a similar pattern of pruned synergistic information for the solving of half of the configurations (corresponding to the non-synergistic UNQ gate), and later an increased level of synergy as one or more sources became integrated (with 5 and 8 configurations being solved).

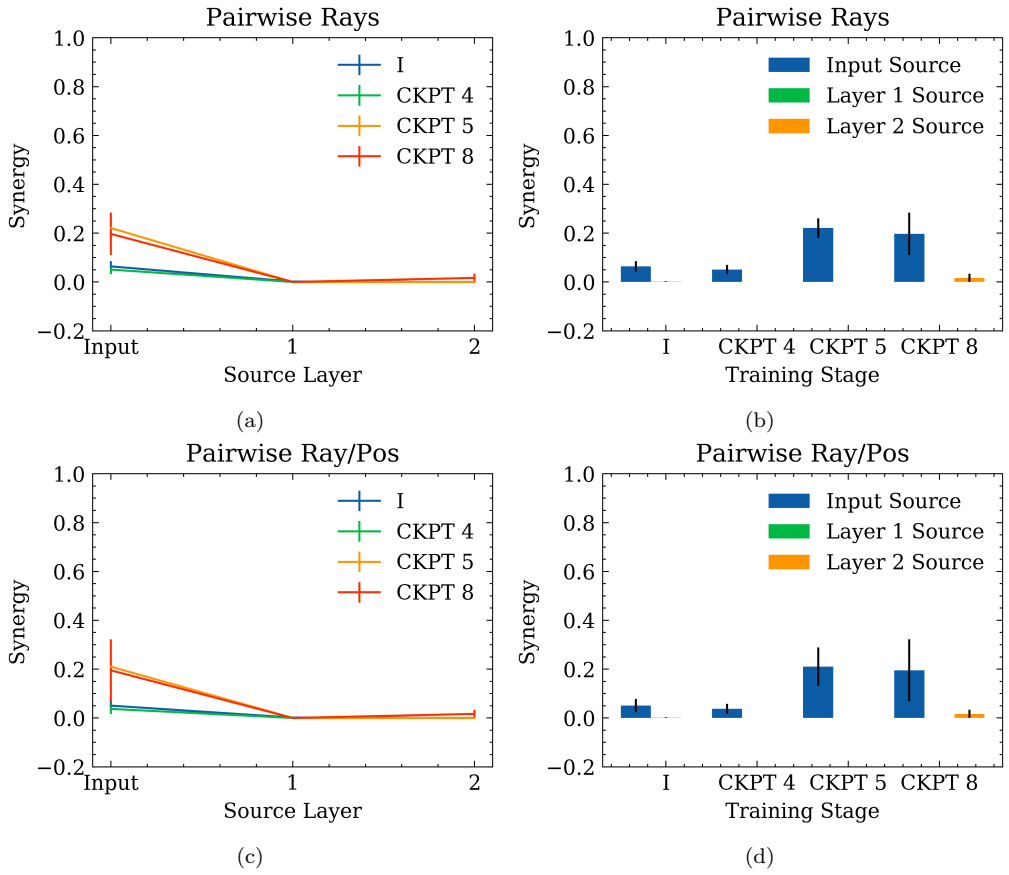


Figure 6.3: 3-bit XOR synergy for number of inputs to gate solved. 3-bit XOR 2-order synergy between each layer source and the subsequent layer target **(a, b)** with pairwise rays input sources, and **(c, d)** with pairwise ray and position input sources.

6.3.2 Curriculum Task

How do information dynamics change when an agent must adapt to integrate an additional source of information and change its input mapping? We studied this question through curriculum training, transitioning from the 2-bit XOR to 3-bit XOR task. During curriculum training, models were tested and synergy values were computed at the training points as defined in Section 4.2.1, namely configuration 1 initialization (1I), configuration 1 threshold (1T), configuration 2 adaptation (2A), configuration 2 recovery (2R), and configuration 2 end (2E).

As expected, we found that synergy values increased (fig. 6.4) from the initialization of the network as each agent reached the threshold of maximum reward or maximum number of steps for the configuration, corresponding with the integration of both sources of information to solve the XOR gate (fig. 6.3). Interestingly, after changing configurations to instead train on the 3-bit XOR task, synergy values dropped (2A) from the threshold simultane-

ously with performance significantly decreasing as agents were not able to generalize to the new task initially. As performance gradually improved on the new task, synergy values also increased (2R, 3E), with the final time step exhibiting the highest amount of synergy.

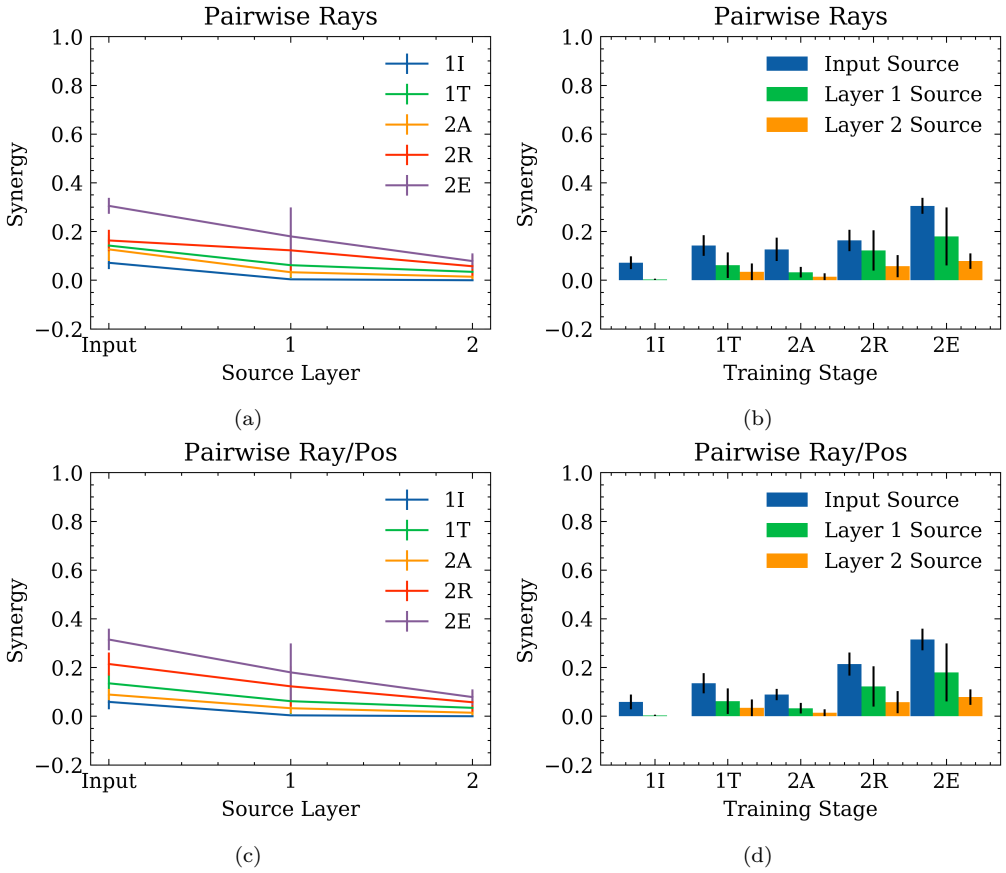


Figure 6.4: 2-bit to 3-bit XOR curriculum synergy. 2-bit to 3-bit curriculum synergy between each layer source and the subsequent layer target **(a, b)** with pairwise rays input sources, and **(c, d)** with pairwise ray and position input sources.

Notably, the level of synergy at the final time step of the second configuration exceeded the level of synergy yielded at successful learning of the 3-bit XOR gate trained alone, although most agents in the curriculum experiment were not able to reach perfect accuracy on the task. Another observation to note is that synergy in the first and second layer sources increased during the learning phase of the second task (2R, 2E), whereas the 3-bit XOR task trained alone yielded minimal to no synergy in the first and second layers. One possible explanation for these behaviors could be that the networks adapt to learning new tasks by increasing the amount synergistic information within the network rather than using other forms of information (redundant, unique); perhaps synergy is more efficient for learning and extracting new information from the input than re-learning (and re-mapping) unique or redundant representations is. This could explain the initial drop in synergy upon task change as being a result of weight-adaptation to accommodate new information, which later

leads to higher levels of synergy. It also supports the presence of synergistic information in the first and second layer sources as having emerged to adapt to the new task, whereas initial training of a first task instead learns compressed unique representations in these layers for efficiency (synergy being an inefficient form of representation). One possibility is that the network could reuse the learned 2-bit XOR 'solver' presumed to be in the input source to the first layer target and then learn an additional 2-bit XOR in the first layer source to second layer target to solve the 3-bit XOR gate. Synergistic information may be seen as acting as an extractor of additional information from a learned initial mapping. Furthermore, these results suggest a relationship between hierarchical representations, transfer learning, and synergy.

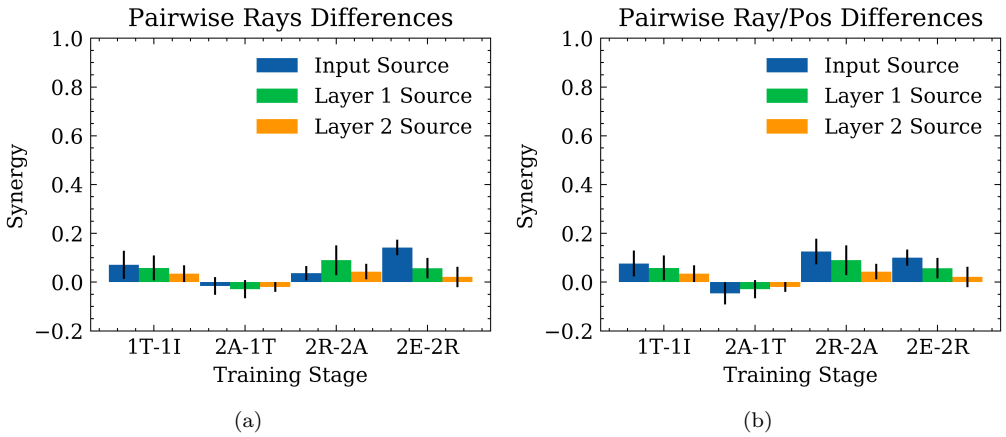


Figure 6.5: Synergy differences between subsequent training points in the 2-bit to 3-bit XOR curriculum **(a)** for pairwise rays input sources, and **(b)** for pairwise ray and position input sources.

One potential interpretation of these results could be that although synergy may generally improve a system's ability to create higher-level representations and thus correspond to better performance, the presence of increased synergy does not necessarily indicate the performance of the network. Indeed, in this experiment, the levels of synergy are high for the 3-bit configuration, but the majority of agents do not successfully solve the task although their networks have higher levels of synergy than agents that are capable of solving the task. However, this idea does not consider the performance of both sets of agents (trained only on 3-bit versus 2-bit to 3-bit) across both the 2-bit and 3-bit tasks. Perhaps it is the case that agents with higher levels of synergy have better overall performance on multiple tasks, and thus have improved generalization, than agents with less synergy. This can be tested in future work.

6.4 2-Bit Distance XOR Task

We next investigated compound tasks by increasing the length of the platform the agent was placed on in order to delay reward, requiring agents to move forward or backward for an extended period of time. We predicted that such a task could be solved in several different ways. The agent could offload synergistic information about the logic gate into a trajectory to the reward after its first action by instead relying on information about the global position rather than the raycast information about the barriers. Alternatively, the agent could continue processing relevant information about the logic gate input at each time step and use it to inform each subsequent action towards the reward. We were particularly interested in studying whether a simple addition of complexity to the XOR task would yield any differences in performance or measures of synergy.

6.4.1 Single Task

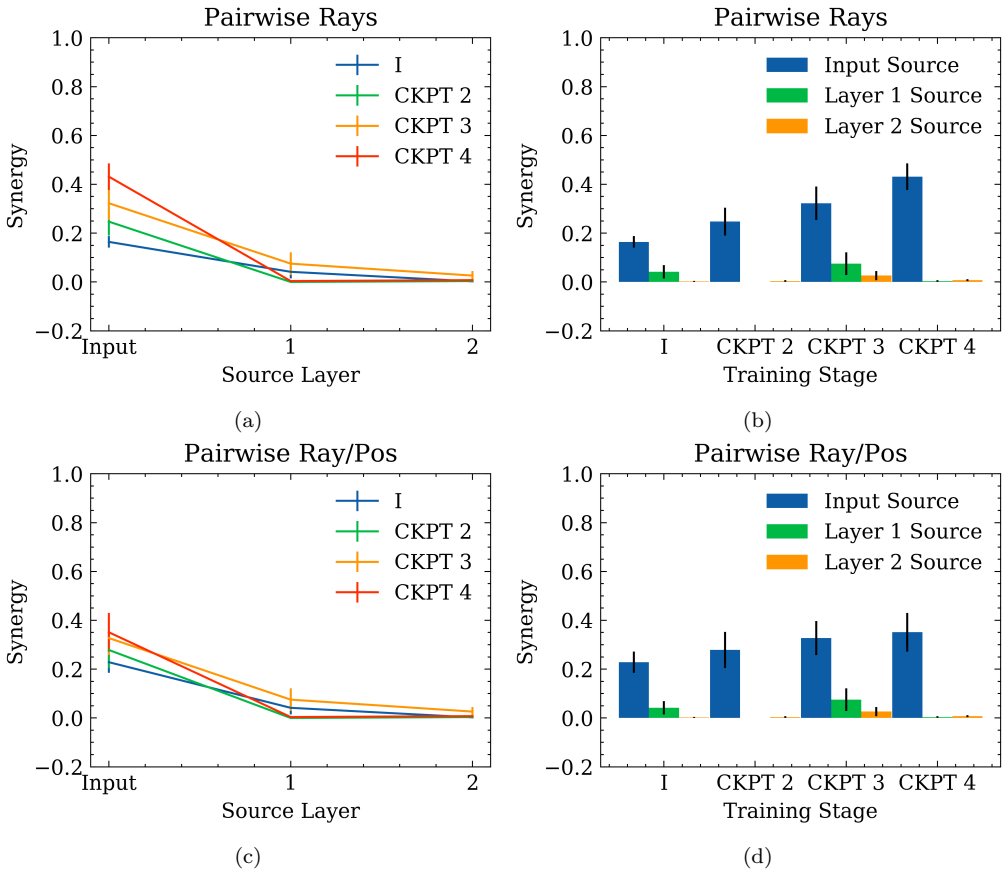


Figure 6.6: 2-bit distance XOR synergy for number of inputs to gate solved. 2-bit distance XOR 2-order synergy between each layer source and the subsequent layer target **(a,b)** with pairwise rays input sources, and **(c,d)** with pairwise ray and position input sources.

As shown in Figure 6.6, the 2-bit XOR task using a platform of length 10 had substantially higher levels of input source synergy compared to the previously observed tasks which only used a platform of length 1. Furthermore, these synergy values remained high for both pairwise raycasts and position raycasts, potentially supporting the idea that agents continuously integrate task-relevant information while solving the task, rather than offloading synergistic information onto the trajectory.

6.4.2 Curriculum Task

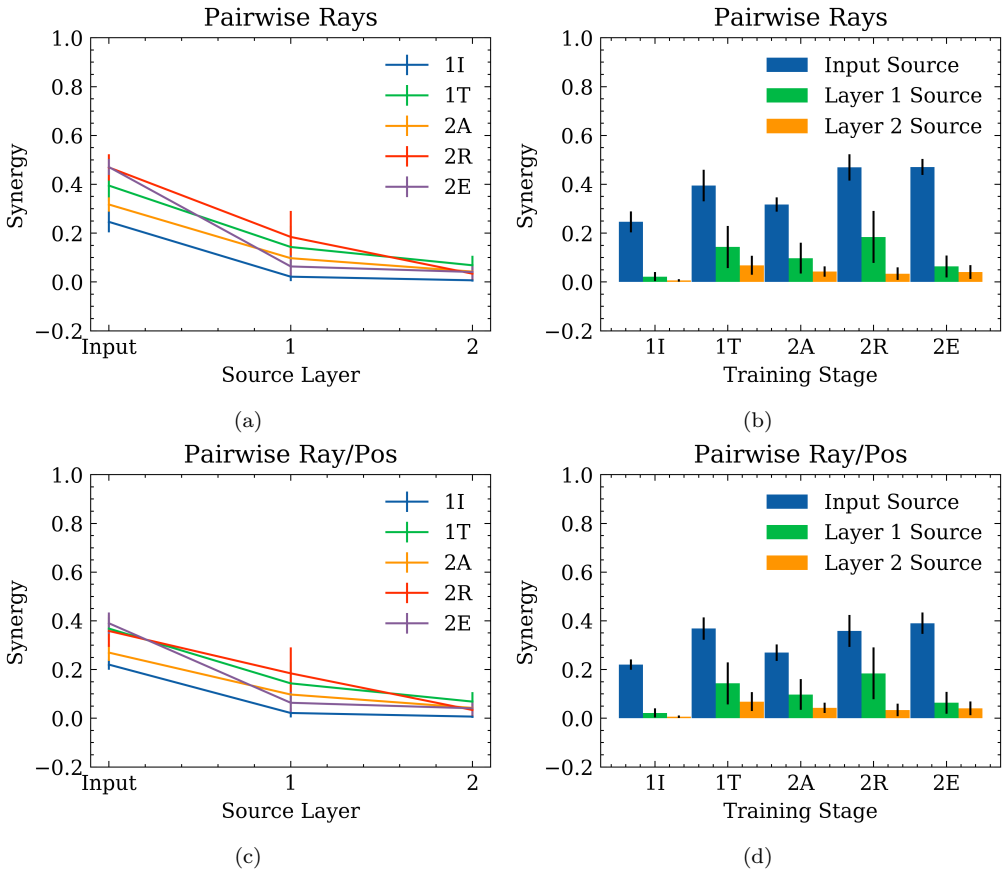


Figure 6.7: 2-bit distance XOR curriculum synergy. 2-bit distance curriculum synergy between each layer source and the subsequent layer target **(a,b)** with pairwise rays input sources, and **(c,d)** with pairwise ray and position input sources.

Interested in expanding our results, we repeated the same experimental design of curriculum training and evaluation, transitioning from the short distance (platform length of 1) 2-bit XOR task to the long distance (platform length of 10) 2-bit XOR task. Again, models were tested and synergy values were computed at training points as previously defined (1I, 1T, 2A, 2R, 2E). We observed results that were consistent with the 3-bit version of curriculum training: after changing tasks, both performance and synergy initially dropped, as agents

were not able to generalize to the new task. After training continued and performance improved on the new task, the level of synergy also increased to exceed the amount of synergy yielded by training the second configuration alone (fig. 6.7). As also seen in the 3-bit XOR curriculum experiment, there was more synergy present in the first and second layer sources after task-transfer compared to training of individual distance XOR tasks which yielded little to none.

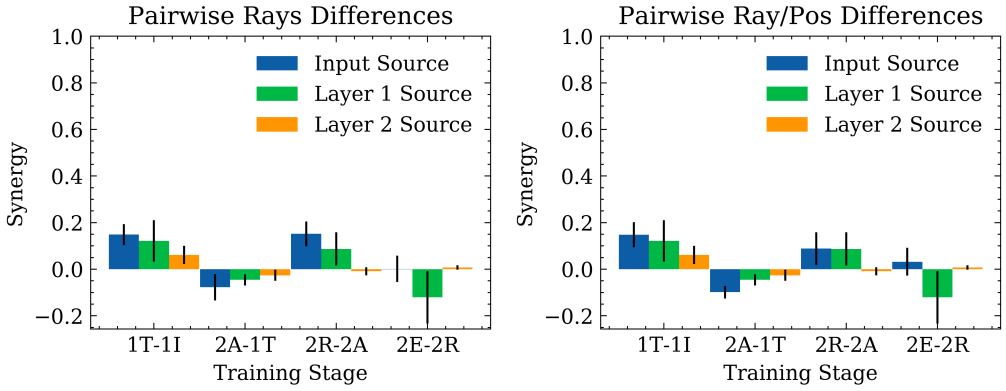


Figure 6.8: Synergy differences between subsequent training points in the 2-bit distance XOR curriculum **(a)** for pairwise rays input sources, and **(b)** for pairwise ray and position input sources.

These observations further support the idea that the networks adapted to learning new tasks by increasing synergistic information rather than predominantly relearning unique representations of information, even in simple extensions of tasks (supersets). As shown in Figure 6.8, the mean difference between the amount of synergy at the end point and recovery point of the second configuration (2E-2R) is negative for the first layer sources. This is different from the pattern seen in the 3-bit XOR curriculum experiment, which shows a consistent trend of increasing synergy as performance improves. As opposed to the 3-bit curriculum, in which the majority of models trained were not able to solve the second task, most models in this curriculum did solve the task which may have led to a pruning of synergistic information unrelated to the solving of the XOR gate itself.

6.5 Increasing 2-Bit Distance XOR Task

The observed pattern of subsequent phases of increasing and decreasing synergy with task-transfer led us to explore whether the phenomena would continue across several configurations. To do this, we trained agents on curricula consisting of four configurations with progressively increasing platform distances (lengths 1, 10, 20, and 30). The agents were then evaluated at initialization (1I), each threshold point (maximum reward or number of steps; _T), and adaption in each task following the first (10,000 steps in the new configuration;

_A). The results are shown in Figures 6.9 and 6.10. Again, the same dynamic of modulating synergy values (synergy increasing from initialization to threshold, decreasing from threshold to initialization of new configuration, and increasing again to threshold, surpassing the previous threshold level of synergy) was present in the input sources to first layer target for the first three configurations. This further supports our hypothesis that synergy improves the ability for models to generalize to new tasks, with less reliance on relearning the mapping of unique information, as it continues across the changing of multiple tasks.

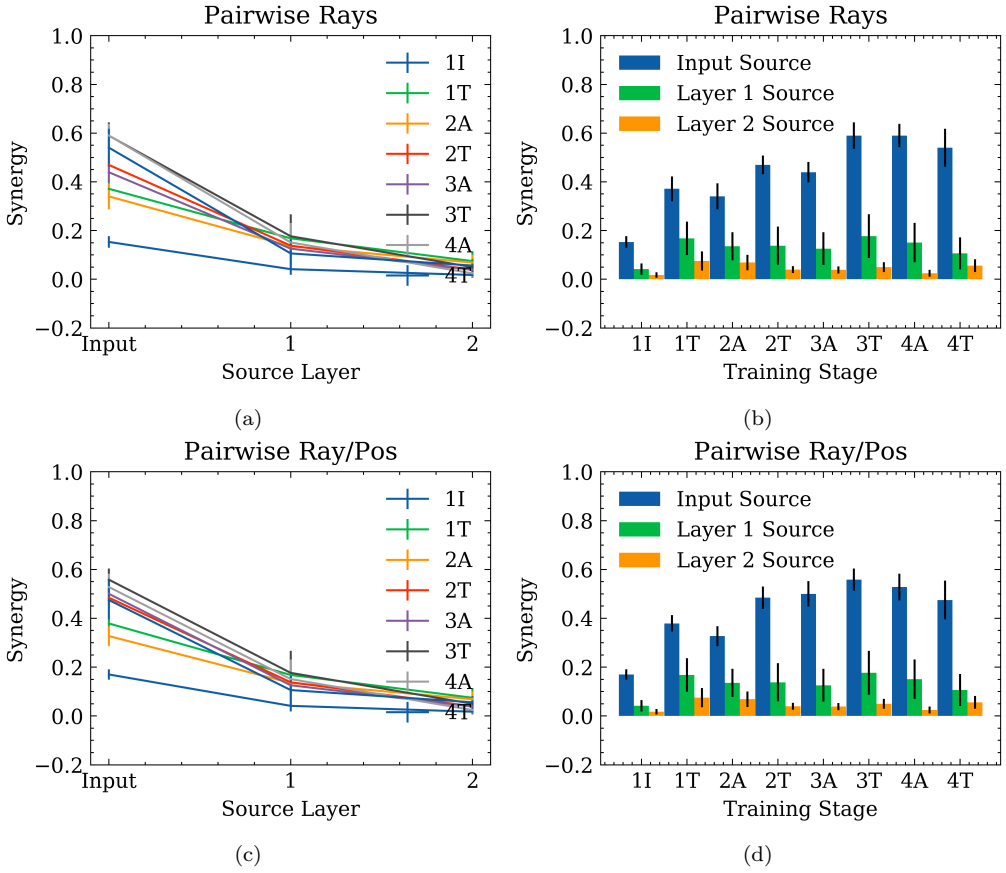


Figure 6.9: 2-bit increasing distance XOR curriculum synergy between each layer source and the subsequent layer target **(a,b)** with pairwise rays input sources, and **(c,d)** with pairwise ray and position input sources.

Interestingly, the fourth configuration exhibited the opposite pattern, with the amount of input synergy at initialization remaining at approximately the same magnitude as that in the third configuration threshold, and then dropping upon reaching the configuration four threshold. This also corresponds with the difference between subsequent steps appearing to decrease in magnitude, effectively 'leveling off.' One possible explanation for this behavior is that the information space available for partitioning into synergistic information about the input is reached. Thus, synergy is eventually bottle-necked due to the need for other forms of information. Due to the network's inability to increase synergistic information

without losing valuable unique information, synergy is subsequently pruned and other forms of representation must be re-learned.

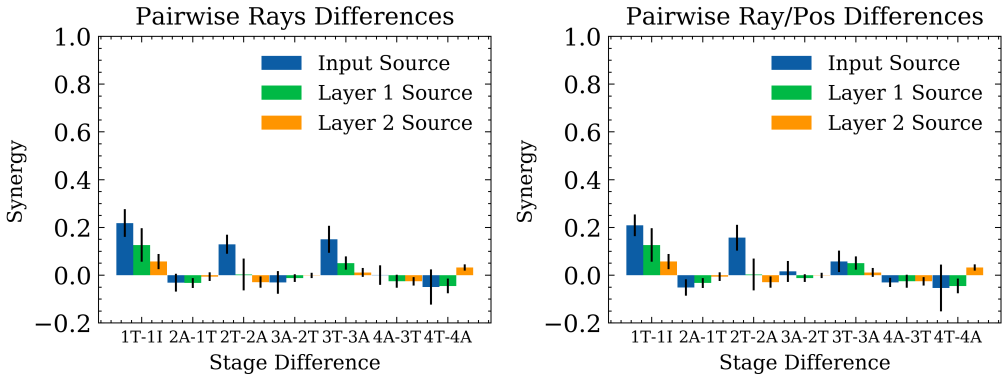


Figure 6.10: Synergy differences between subsequent training points in the 2-bit increasing distance XOR curriculum for **(a)** for pairwise rays input sources, and **(b)** for pairwise ray and position input sources.

Another possible explanation could be that, in this setting, the models have learned a sufficient mapping by the fourth configuration such that they do not require additional synergistic information for 're-learning.' After the changing of task using a platform of length one to length ten, much of the novelty of the transfer has been revealed. Thus, agents may not be learning much additional information with each following configuration, creating the 'leveling-off' of synergy. Perhaps the change to the fourth configuration does not require any synergy to generalize to, as the agents have learned to continue moving in one direction indefinitely based on the barrier types (bits), and thus yield an approximately equal amount of synergy from the third configuration threshold to 10,000 steps into the fourth configuration. If this occurs, the decrease in synergy at the fourth configuration threshold could be seen as a pruning of additional information that is unnecessary for solving the task. Future work should investigate this observed pattern of synergy 'leveling-off' further.

Chapter 7

Discussion and Future Directions

Humans' and animals' capability to perform and generalize over complex tasks distinguishes us from current machine learning models. Understanding how biological brains are able to perform such computations is a question that has been widely studied, but is still not fully-understood. Of the many different approaches to studying both biological and artificial neural networks, one point of convergence is that of information theory. Leveraging the tools of PID, there has been evidence shown for information synergy's correlation to complex cognition and consciousness in the brain. In this study, we use the same measures to analyze information dynamics in artificial models performing various tasks, both in a classification and a reinforcement learning setting.

Our results show that synergy is a function of task. In both supervised networks and RL models learning logic gates, the amount of synergy is partially-dependent on the amount of input information required for successfully solving a task. Hence, tasks which do not necessitate the use of multiple sources of information yield lower values of synergy. This observation is further supported by the fact that dropout prunes synergy in tasks which do not require integration of information, but maintains high levels of synergy when required for successful completion of the task. Our study shows that dropout can be seen as a form of regularization which increases redundancy of important information and reduces synergy of irrelevant input, but preserves essential synergistic information. In RL models, we also see that synergy increases as additional information is successfully integrated for solving a given task. This is expected since, by definition, synergistic information is present when additional information is yielded by the presence of multiple sources meaning that systems which successfully perform tasks which require such integration will have some level of synergy.

One may be inclined to ask why synergistic information is more prevalent from input sources to first layer target compared to the rest of the network, and why networks performing non-synergistic tasks still do not also exhibit high levels of synergy. We argue that due to its distribution across neuronal space, synergistic information is a less efficient form of representation. Rather than compressing information into one neuron (i.e., unique information),

synergy requires the use of several neurons. This is supported by the fact that across all models trained in this study, there was a consistent trend of the highest level of synergy existing from the input sources to the first hidden layer target, whereas all layers afterwards compressed the information into other forms (2-order redundant, unique). Notably, this does not exclude synergy from ever being present in other layers or networks – if the extraction of synergistic information is important for learning a task and creating higher-level representations or for generalization, synergy may still exist. This presents a trade-off in the networks between representation efficiency and potentially richer representations/ability to perform several different tasks.

The present study also suggests that the level of synergy is influenced by the training method. This also serves as an extension to the previous idea of synergy being a function of task; in this setting, the task is learning subsequent sub-tasks. In particular, curriculum-style training requiring the transfer of tasks yields higher levels of synergy upon successful completion of each subsequent new task than in the previous task. One possible explanation for this behavior could be that networks which have already learned a given mapping of input adapt to new tasks by extracting synergistic information rather than relearning other unique representations. This could be due to the accommodation of additional synergistic information in a trained model being potentially more efficient than the relearning of unique information about the input. Thus, networks could be seen as remapping unique information to synergistic information in the new task (in cases where the new task is relevantly related to the previous task), causing an increase in synergy compared to networks which specialize for an individual task and learn to most effectively compress synergistic information pertaining to the task. If true, this phenomena could also explain the slight drop in synergy from the last timestep in one configuration to 10,000 timesteps into the following configuration. The slight decrease could be due to the need to redirect and relearn some of the previous synergistic mappings. The progressive leveling-off of synergy differences with each additional task could be a result of agents learning minimal additional information when transitioning to the fourth configuration due to sufficient prior training on increasing distance tasks; thus, no additional synergy is required for the 'new' task as the agent has already learned sufficient mappings for generalizing to it. An alternative explanation for this behavior could be that networks reach a threshold point of maximum synergy contained in the network without losing other important unique or redundant information.

Our results also showed 2-order synergy to be partially-correlated with full-order synergy in small networks (20 neurons) in terms of relative-rankings and cross-layer dynamics, although larger in magnitude. This observation is consistent with Ref. [45], which found the distribution of synergy across neuronal space to be related to layer size (smaller networks yield maximum levels of synergy in smaller groups of neurons).

In summary, our study provides a basis for future exploration into the following claims: 1. Synergy is a function of task.; 2. There is a trade-off between the higher-order represen-

tations yielded by synergistic information and the efficiency provided by compression into unique information. Networks try to compress synergistic information about the input into unique information when possible.; 3. In task-transfer, the accommodation of synergistic information is more efficient than relearning other forms of information and results in higher levels of synergy.; 4. Lower-order synergy is partially-correlated with higher-order synergy in relative-rankings and cross-layer dynamics in small networks.

What do these observations mean for neuroscience? To preface, these speculations provide suggestions and questions for further study, rather than definitive claims about the brain. First, the evidence that synergistic information is leveraged by the human brain for complex cognition [6, 7] is replicated in our simplified models of the brain (artificial neural networks), supporting their use for the study of information dynamics in learning systems and the idea that previous findings of synergy's relation to human intelligence may also be applicable to machine learning.

This work also highlights a difference between machine learning and the brain. Machine learning generally begins with models that are 'blank' and become extremely specialized for a specific task, whereas the brain is much better at generalization across many different tasks, although it may trade off performance in one specific task. We see the exact same analogy in our experiments – RL models trained on only one relatively difficult task (i.e., 3-bit XOR) perform better than agents forced to generalize from a simpler task to the difficult task (i.e., 2-bit to 3-bit XOR), but have significantly less synergistic information. Comparing the representations between our RL agents and brains, this reveals an important feature of synergy which depends on the learning trajectory of the system. Perhaps the human brain is highly-synergistic because of our 'learning trajectory', being forced to generalize across increasingly many settings, such that our brain develops to have more synergistic representations throughout life.

Our study also provides explanations for the human brain having evolved to become more synergistic [6]. Perhaps the brain evolved to integrate information synergistically in order to perform tasks using multiple sources of information and to generalize to increasingly new settings. For instance, non-human primates may have developed brains which were conducive to tasks innately common to their species (finding food, reproducing, taking care of offspring, etc.). Maybe humans were using similar 'hardware' to learn new tasks, such as using tools. Conceivably, our study suggests this could lead to more synergy in individuals' brains. Perhaps brains which could represent more information synergistically were better suited to adapt to their environment and perform new tasks 'intelligently'; perhaps higher levels of synergy provided greater flexibility in the learning process and acted a form of buffering for changing between tasks rather than specializing on one particular task. Furthermore, if synergy and other information-theoretic values (integrated information) are indeed related to consciousness, as suggested by [7, 11, 12, 13], our results could also suggest that consciousness is an emergent feature of complex and generalizable intelligence. This relates to

Feinberg and Mallat's ideas on global operant conditioning, which argued that animals capable of learning complex behaviors from experience based on rewards (and punishments) must have a conscious perception of pleasure (and pain) in order to learn the behavior related to these stimuli [68].

Another interpretation could be that the previously observed concentration of synergy in the prefrontal cortex [6] could be explained by the need to integrate information from various brain regions for complex tasks and generalize across those sources of information. If synergy is indeed a less-efficient form of representation, this could also explain the reduction of synergy in other areas of the brain responsible for low-level cognition where unique and redundant information dominate. While these areas may exhibit low-order synergy, the need for higher-level representations may indeed be unnecessary, inefficient, and potentially harmful if unique information is more suitable for processing neuronal input.

What might our findings mean for machine learning? Similar to other work attempting to study the theory of deep learning with information theory, our results provide a new interpretation for the way artificial neural networks distribution information to learn and transfer to new tasks, as well as raise many questions for future study. We show that neural networks compress incoming information into unique representations when possible, but use synergistic information to reuse learned unique mappings and extract novel information about a task. Thus, synergy can be viewed as aiding the process of generalization and adaptation through other 'channels' of information mapping. However, due to the inefficiency of synergistic representations, there is likely a trade-off between the neuronal space allocated to synergistic information and unique information. Thus, synergy is often more concentrated at smaller orders, previously found to be relative to the size of the network [45]. These observations are complementary with the widespread success of neural networks with many parameters and layers. Wide networks provide a larger information space to encode both unique and synergistic information, allowing for less of a constrained partitioning between the two. Deep networks allow for unique and synergistic representations to be built off of those in the previous layers. For example, a subsequent layer could conceivably encode additional synergistic information about two or more pieces of synergistic information in the previous layer, allowing for increasingly higher representations to be created, if favorable.

These results are particularly exciting for continual learning cases, where one could speculate that networks learn representations for a first task in the input layer or first several layers. Then, when switching tasks, perhaps the networks could use later layers to 'extrapolate' from these first representations synergistically. Thus, as shown in our curriculum experiments, we see an increase in the synergy of later layers when agents are trained on subsequent tasks, whereas there is little to no synergy when trained on a single task. Our results also show that different training histories reflect not only different capabilities in terms of which tasks an agent can perform, but also the internal structure of the information in the system. We are able to distinguish between these agents by the way they allocate their information space.

Notably, our results have only pertained to the unidirectional case of transfer. Studying whether models can perform bidirectional transfer using learned unique and synergistic representations is of particular importance in the context of generalization.

Another interesting question is whether initial synergy could provide greater flexibility upon the learning process. We see that randomly initialized networks begin with a small amount of synergy and, throughout the learning process, either increase or decrease this initial level. Could synergy provide a buffer for changing between different tasks rather than specializing on one particular task? For instance, if network weights are initialized to have higher levels of synergy within them, could networks learn faster and generalize better? If so, this would clearly be a very interesting result for the machine learning community. It would also parallel our observations about the synergistic brain and its ability to perform complex cognition.

In addition to theoretical understanding, further analyzing the relationship of synergy and learning has relevant applications. It could potentially inspire new approaches to improving current architectures and optimization methods which may improve the way information is extracted, represented, and partitioned in networks.

Limitations

We emphasize that our study is limited and that our observations do not necessarily generalize to all settings. We make no claim that the assertions made in this work are universal proof for information dynamics in neural networks. Instead, we hope to use this work as a proof of concept for previous neuroscience research and as a basis for inspiring future investigations in synergy's relation to learning and generalization in biological and artificial neural networks. Our results could be influenced by a number of different factors, including optimization method (ADAM), activation function (ReLU), discretization method (3 bins from 0 to 5), network size (20 parameters), network architecture (feedforward), RL model (PPO), synergy measures ($I_{\min}, I_{MMI}$), and specific task design. Furthermore, synergy measures are influenced by the sample of episodes tested on, making the comparison across models tested on different samples (i.e., different tasks) limited. Thus, the most definitive evidence exists at the level of within-task comparisons and overall task-dynamic comparisons. Additionally, due to the computational cost of training RL models in complex environments, our sample size of 10 was somewhat small, yielding somewhat high variance in some measurements. Allotted task difficulty and corresponding performance were also constrained by the small size of the actor-critic network, necessitated for tractably computing full-order synergy.

Future Work

Beyond experimental results and analyses, the design of this project provides a strong framework and large capacity for future work. Part of the utility of using the Animal-AI environment comes from the ability to continually create new paradigms for testing particular cognitive abilities and studying various types of tasks. First, future research should seek

to resolve the limitations of this study in order to make more definitive claims about the behavior of synergy in RL models. In particular, reproducibility across different task regimes and types of models is critical to better establish whether the observed patterns are consistent. The potential relationship between synergy and flexibility in generalization should also be rigorously studied, including the evaluation of bidirectional task-generalization. As previously stated, future studies should also investigate whether the initial level of synergy in networks predicts future performance.

Another point of interest would be to study the way information is encoded and localized in networks. Although we believe from this study that input information is predominantly compressed into unique and low-order synergistic/redundant information, we do not have a strong understanding of how this occurs. We also do not know how to determine the order of maximum localization without computing values over several orders. If information is localized in some way, we may be able to better determine how this information is encoded. Further study of such processes may aid in interpreting our models and better extracting from entangled representations (synergistic information).

Limited by our sole use of feedforward networks, we are particularly interested in expanding our study to include recurrent neural networks. The brain is known to compute information recurrently and a comparison with feedforward networks would be useful for studying the role of memory on network information decomposition. This would also allow for tasks with an increased-reliance on integrating different sources of information across time. Another extension of interest would be the use of models implementing global workspace-inspired architecture to further investigate the idea of a synergistic workspace as proposed by [7], merging two major theories of consciousness.

Chapter 8

Conclusion

In this work, we analyzed information decomposition interactions in machine learning settings. We first observed the effects of dropout, a popular form of regularization in deep learning, on synergy and redundancy in supervised models learning two types of logic gates, UNQ and XOR. In our following set of experiments, we studied synergy in reinforcement learning agents as they performed increasingly complex tasks in the Animal-AI environment. Our results supported two main findings. First, synergy is an inefficient representation that is compressed into unique information and low-order synergistic/redundant information when possible. Second, synergy is used to adapt to new tasks, as it allows for additional extraction of input information without significantly changing the unique or redundant representations in the network. As such, settings requiring generalization to new tasks yield more synergy than tasks performed in isolation. Our findings create a new interpretation of synergy as providing the capacity for generalization and ability to extract new information. Furthermore, we re-frame the partition of mutual information into synergistic information and unique information as presenting a trade-off between complex higher-level representations and efficiency of information flow, respectively.

Chapter 9

Appendix

.1 Figures from Logic Gate Network Experiments Using I_{MMI}

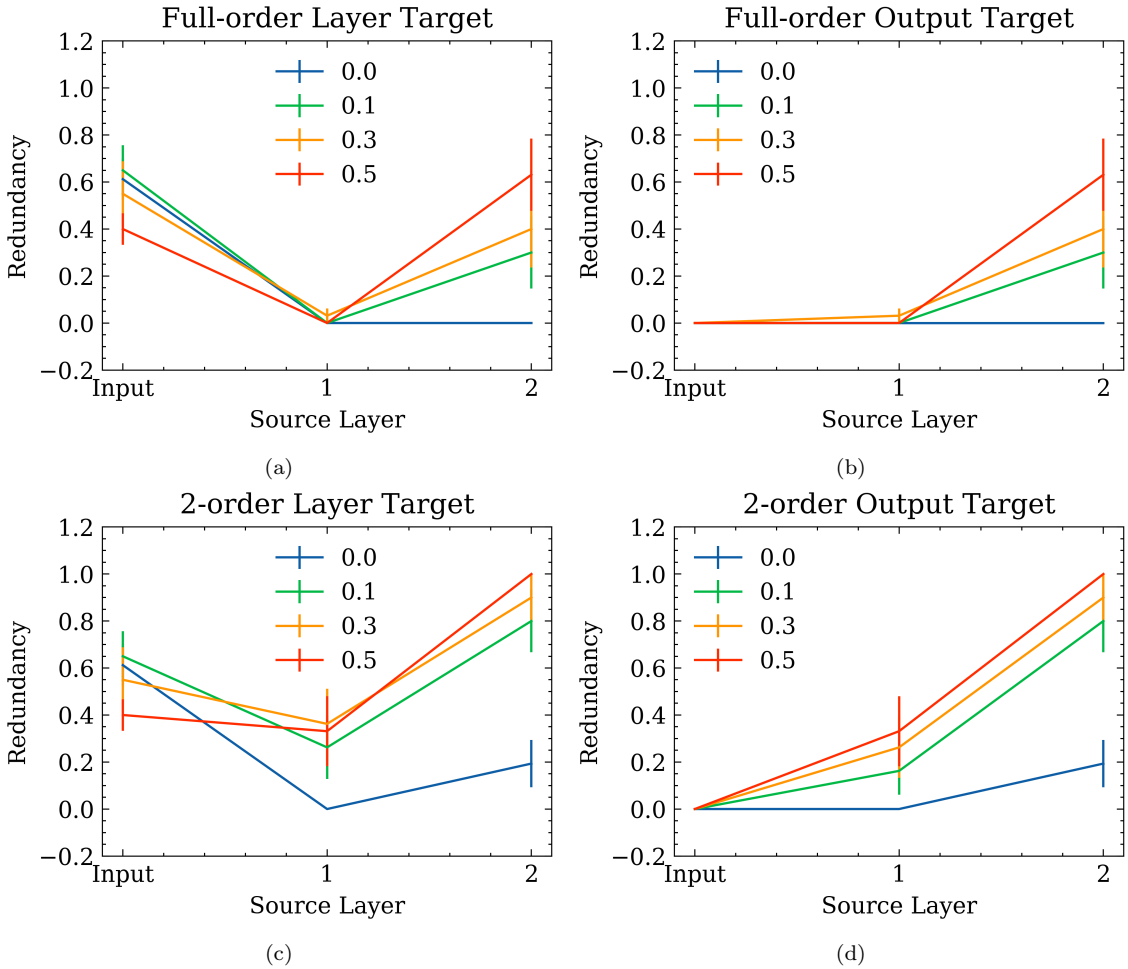


Figure 1: UNQ network redundancy for varying levels of dropout. **(a)** Redundant information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order redundant information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

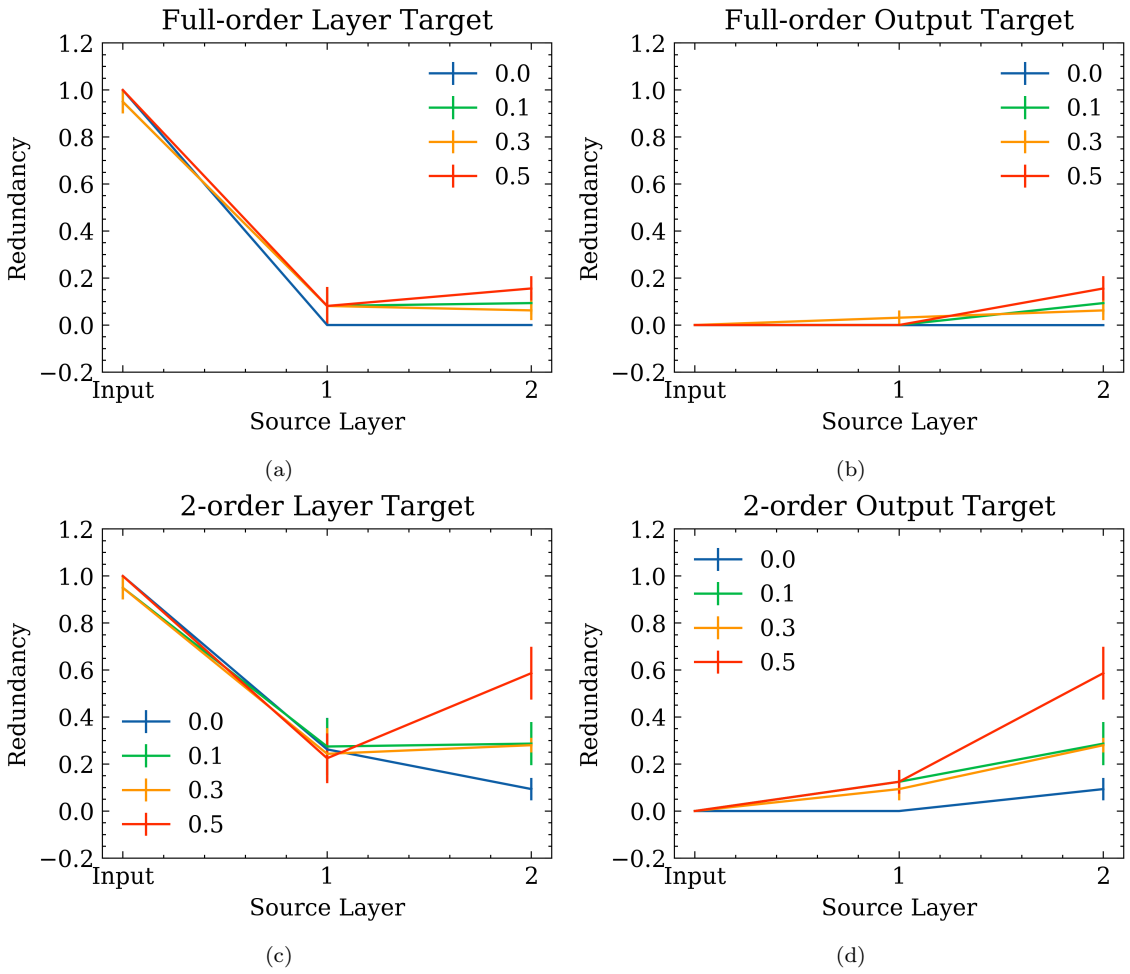


Figure 2: XOR network redundancy for varying levels of dropout. Redundant information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order redundant information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

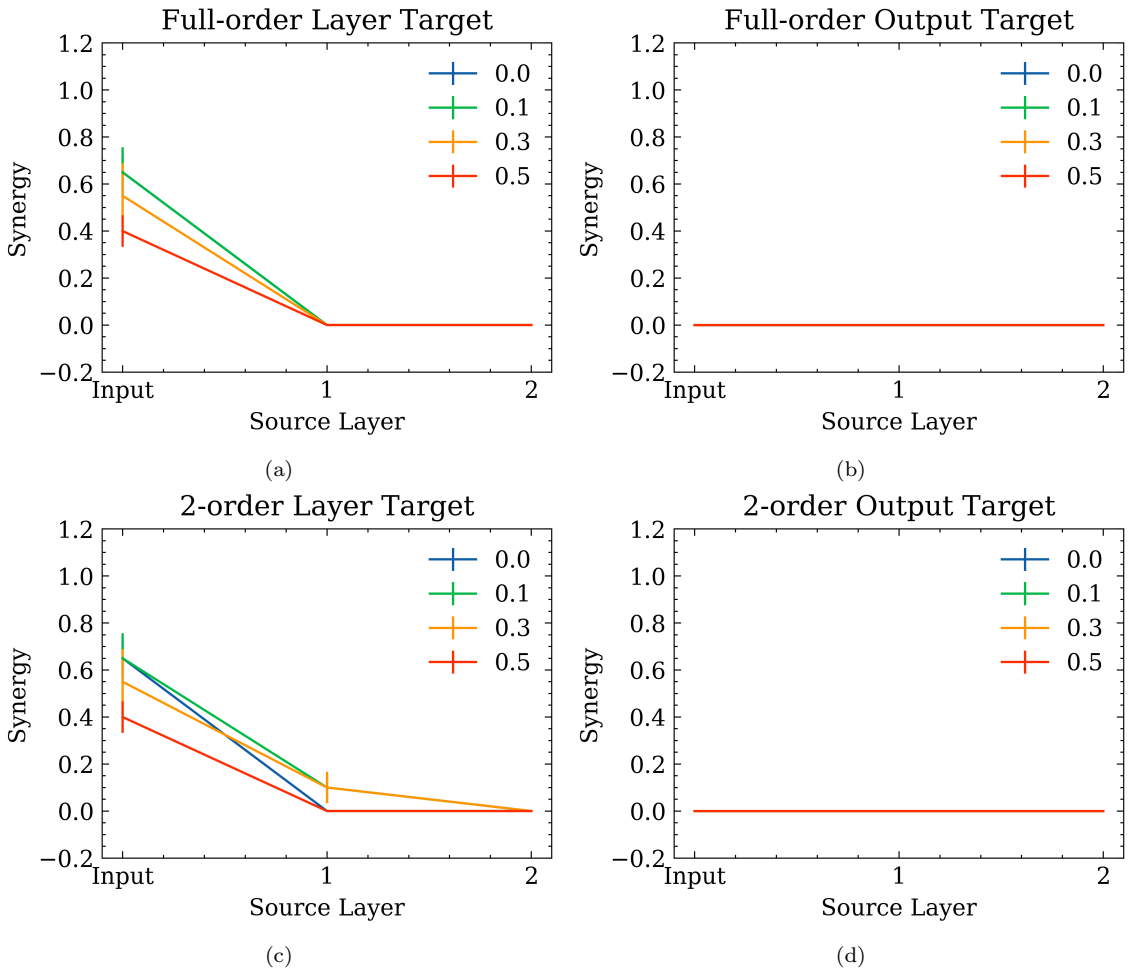


Figure 3: UNQ network synergy for varying levels of dropout. **(a)** Synergistic information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order synergistic information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

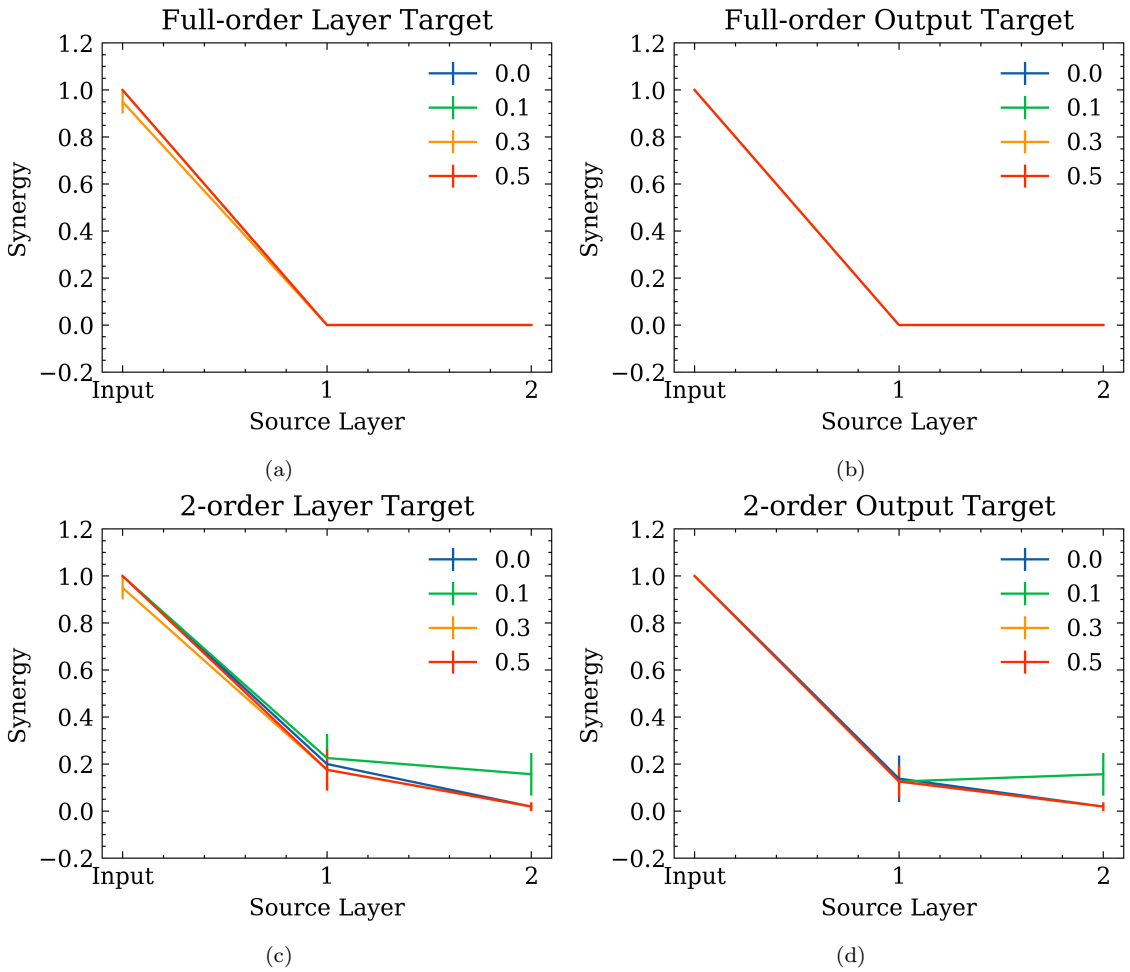


Figure 4: XOR network synergy for varying levels of dropout. **(a)** Synergistic information between each input/layer source and subsequent layer target, and **(b)** input/layer source and output target. **(c)** 2-order synergistic information between each input/layer source and subsequent layer target, and **(d)** input/layer source and output target.

.2 Figures from Animal-AI Experiments Using I_{MMI}

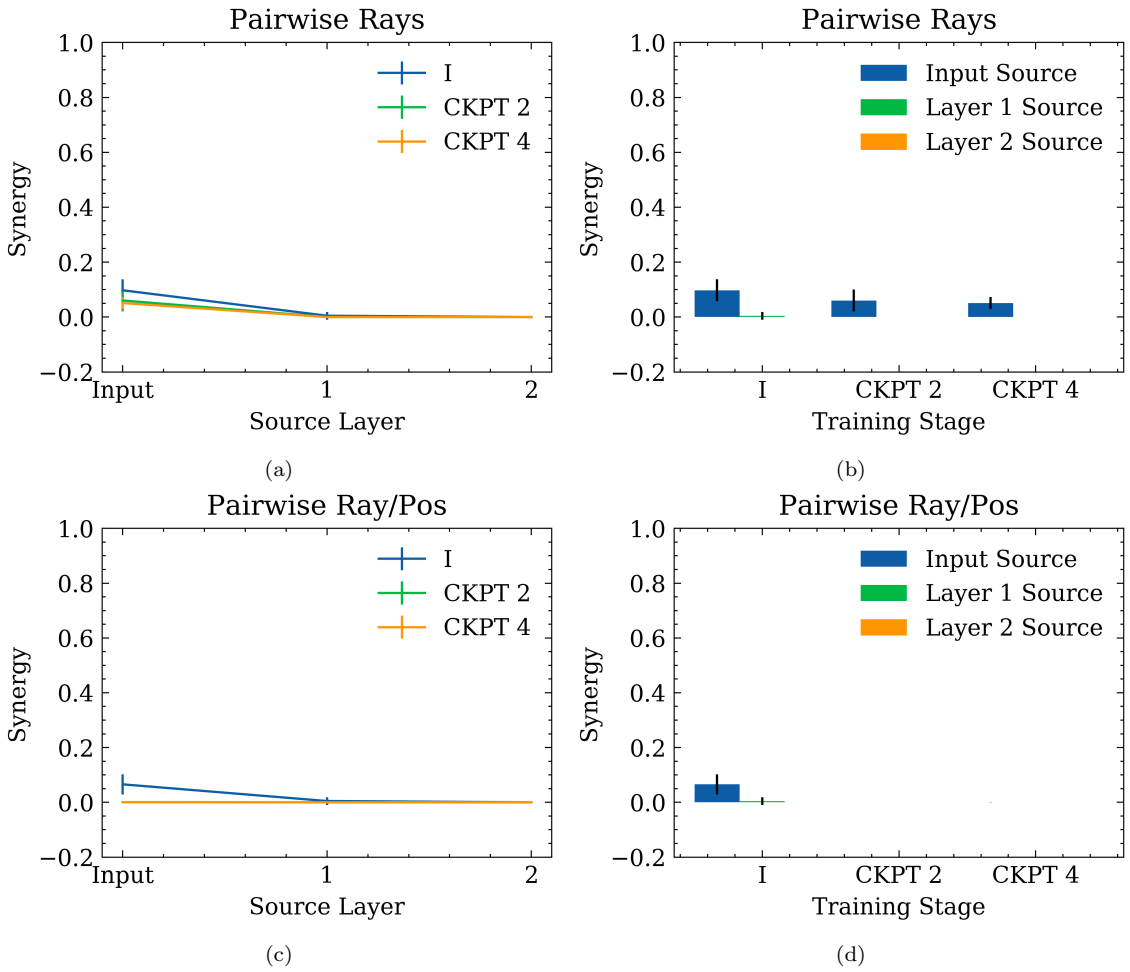


Figure 5: 2-bit UNQ synergy for number of inputs to gate solved. **(a, b)** 2-bit UNQ 2-order synergy between each layer source and the subsequent layer target with pairwise ray input sources, **(c, d)** with pairwise ray and position input sources. Values of layer 1 and 2 sources were too small to display on the graph.

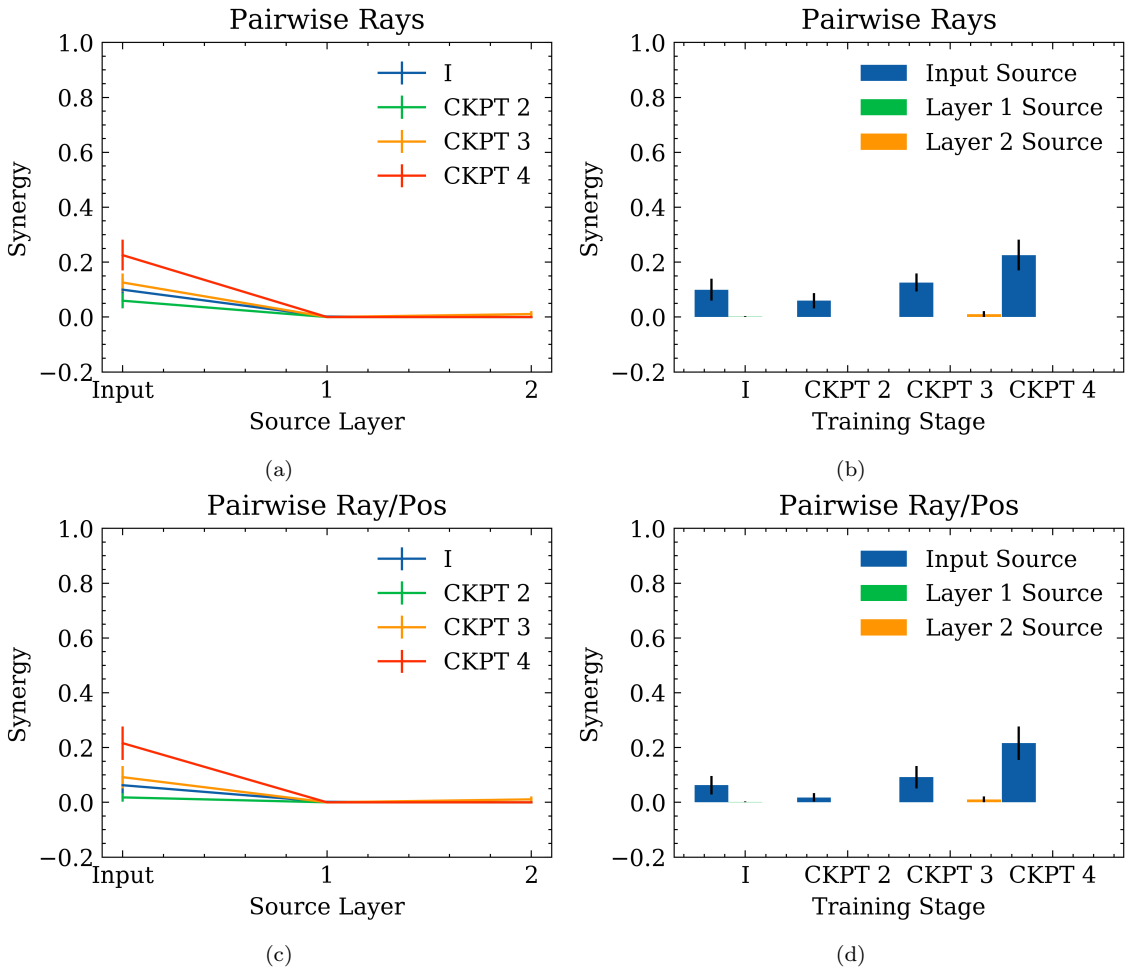


Figure 6: 2-bit XOR synergy for number of inputs to gate solved. **(a, b)** 2-bit XOR 2-order synergy between each layer source and the subsequent layer target with pairwise ray input sources, **(c, d)** with pairwise ray and position input sources.

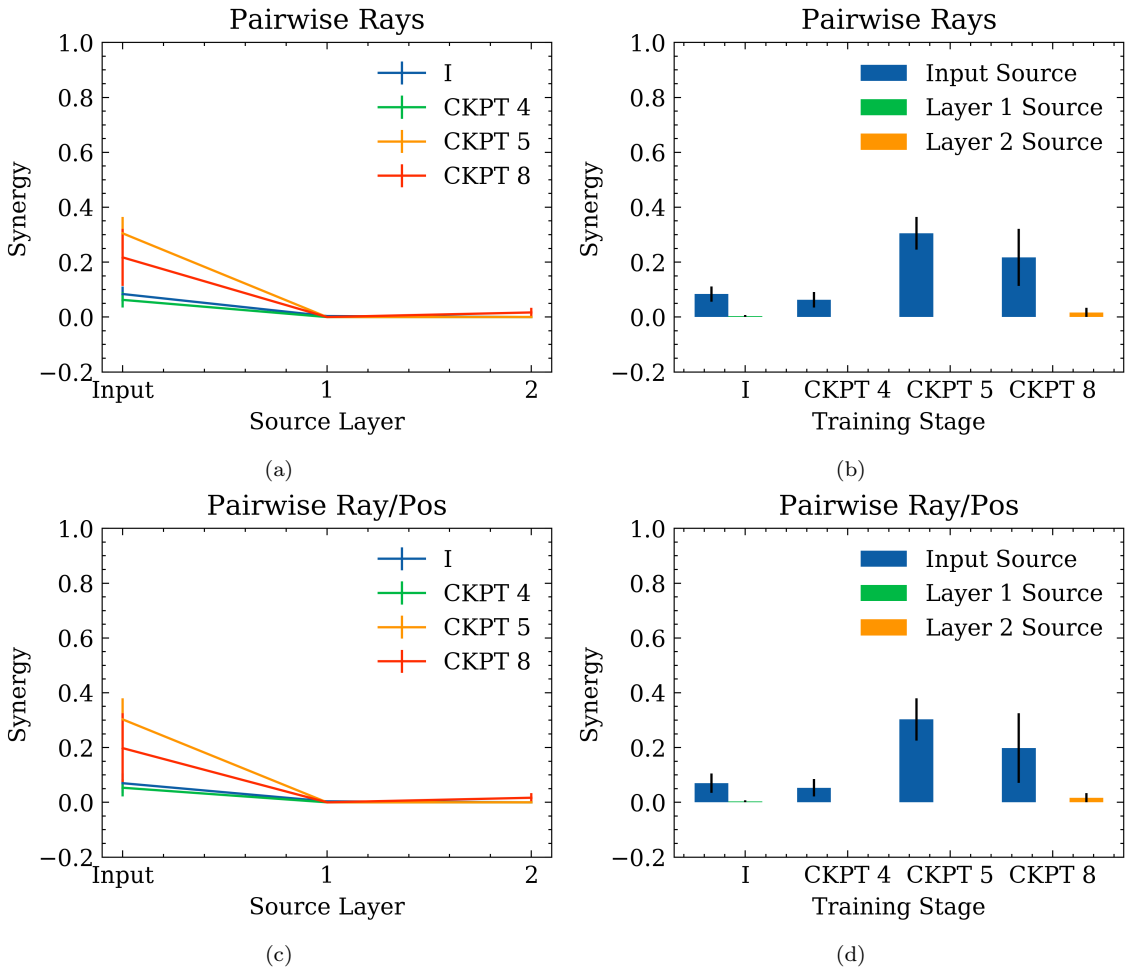


Figure 7: 3-bit XOR synergy for number of inputs to gate solved. 3-bit XOR 2-order synergy between each layer source and the subsequent layer target **(a, b)** with pairwise rays input sources, and **(c, d)** with pairwise ray and position input sources.

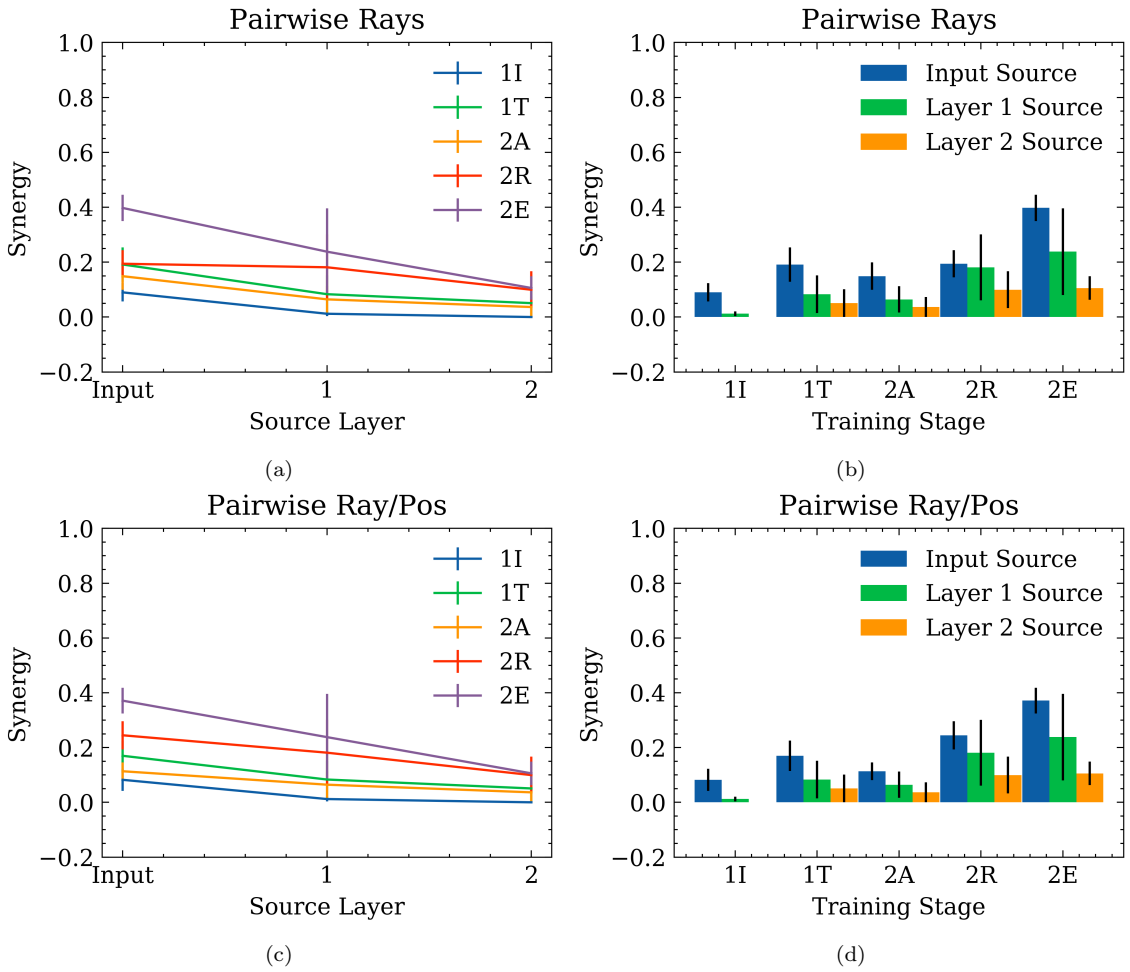
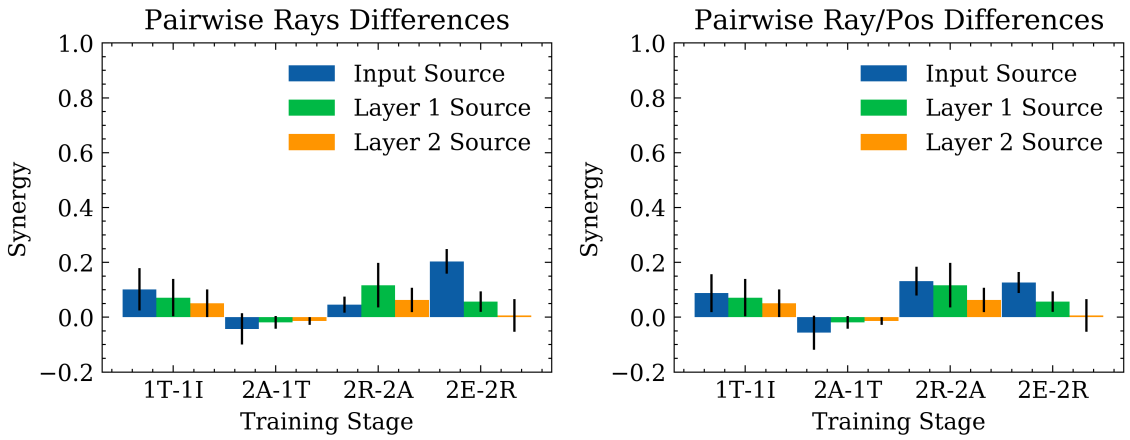


Figure 8: 2-bit to 3-bit XOR curriculum synergy. 2-bit to 3-bit curriculum synergy between each layer source and the subsequent layer target **(a, b)** with pairwise rays input sources, and **(c, d)** with pairwise ray and position input sources.



(a) (b)

Figure 9: Synergy differences between subsequent training points in the 2-bit to 3-bit XOR curriculum **(a)** for pairwise rays input sources, and **(b)** for pairwise ray and position input sources.

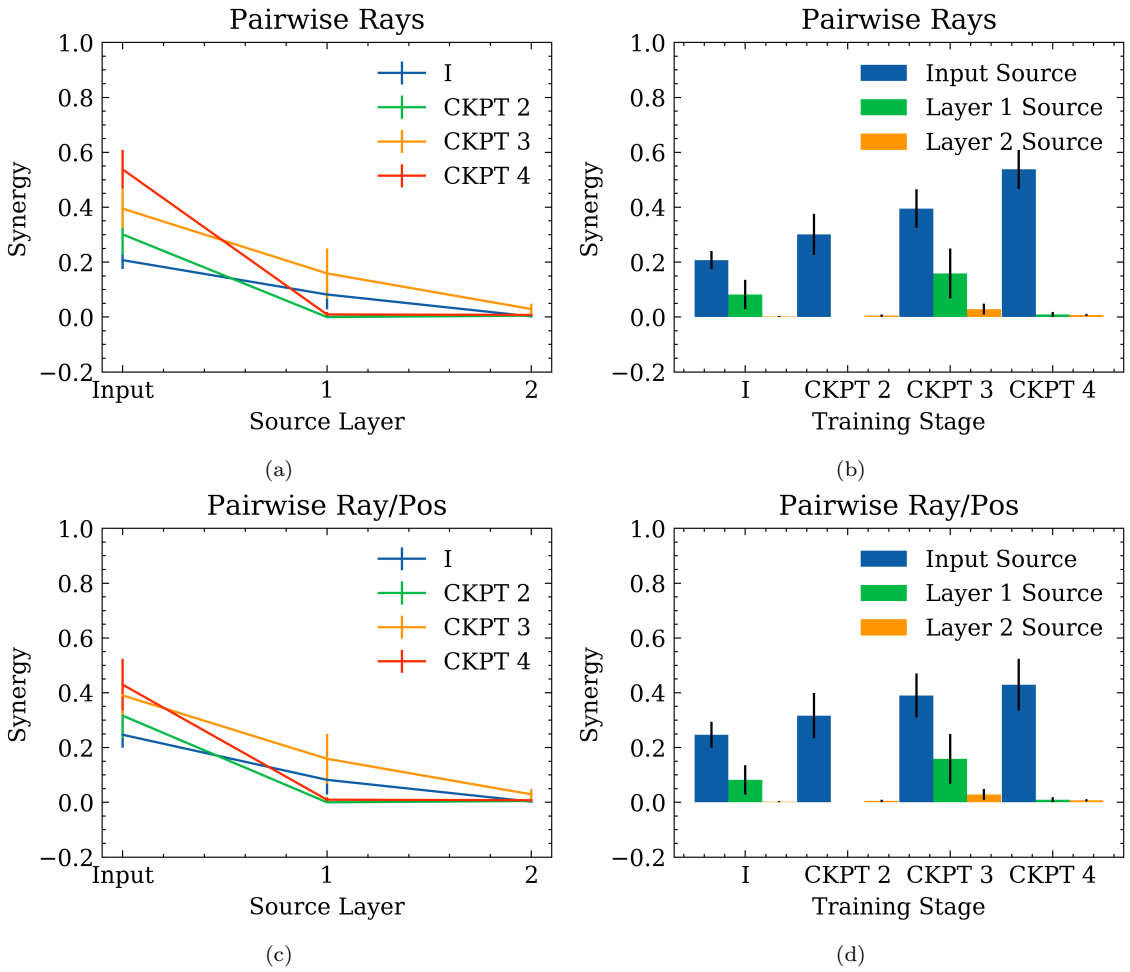


Figure 10: 2-bit distance XOR synergy for number of inputs to gate solved. 2-bit distance XOR 2-order synergy between each layer source and the subsequent layer target **(a,b)** with pairwise rays input sources, and **(c,d)** with pairwise ray and position input sources.

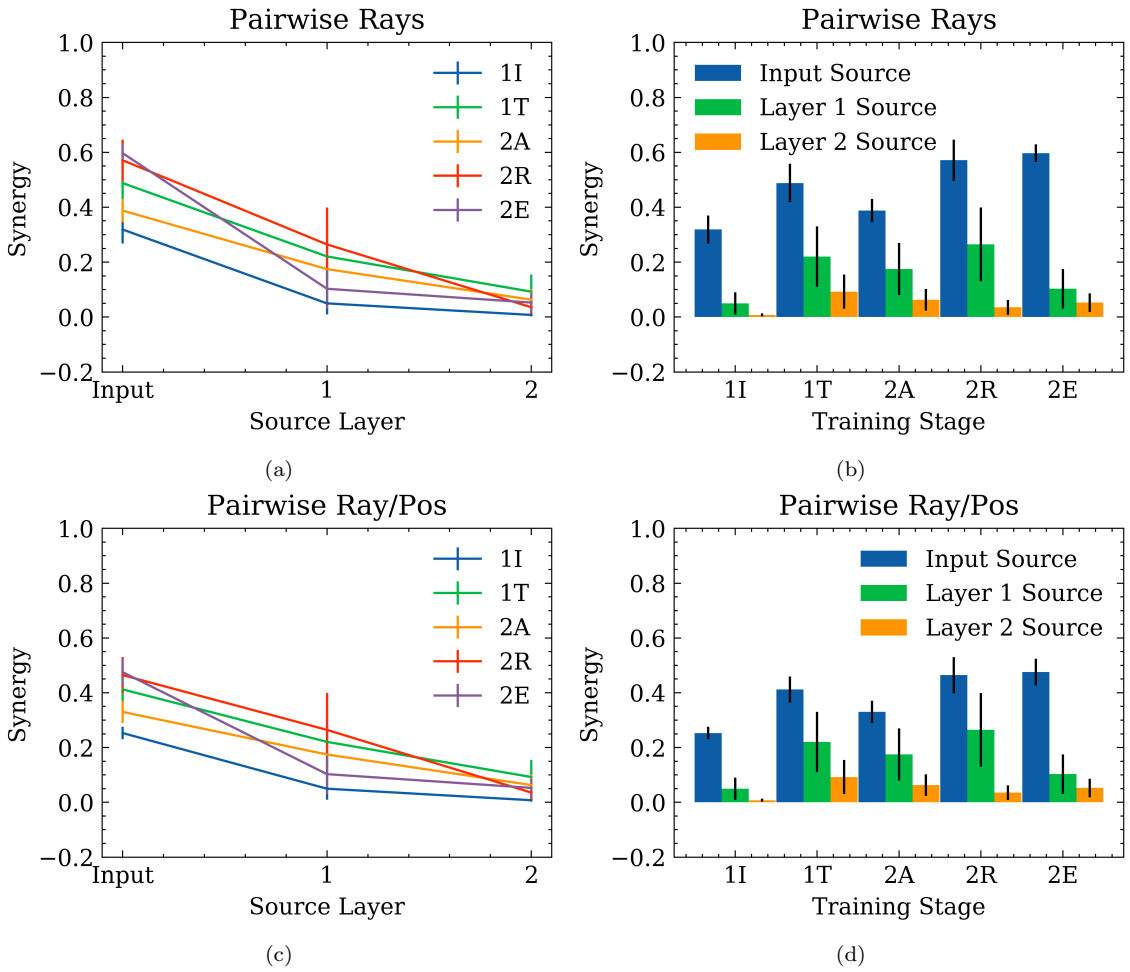
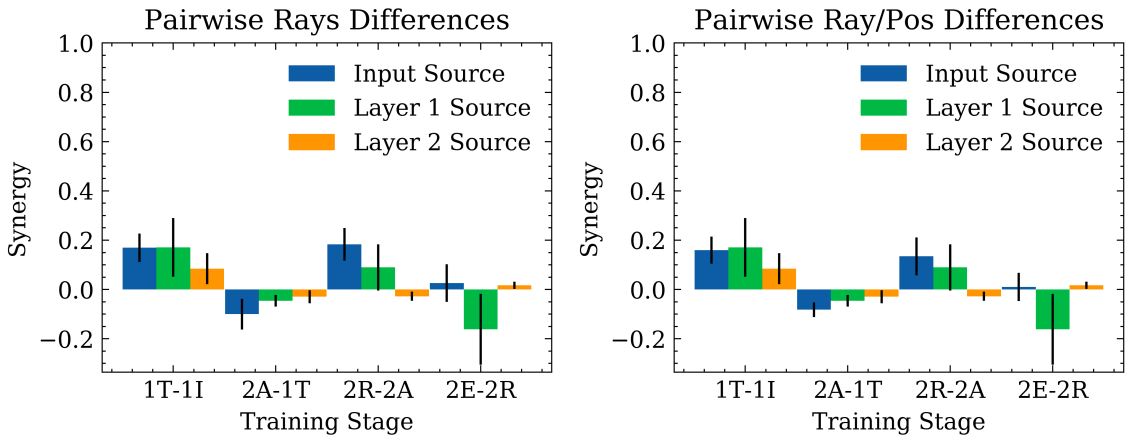


Figure 11: 2-bit distance XOR curriculum synergy. 2-bit distance curriculum synergy between each layer source and the subsequent layer target **(a,b)** with pairwise rays input sources, and **(c,d)** with pairwise ray and position input sources.



(a) (b)

Figure 12: Synergy differences between subsequent training points in the 2-bit distance XOR curriculum **(a)** for pairwise rays input sources, and **(b)** for pairwise ray and position input sources.

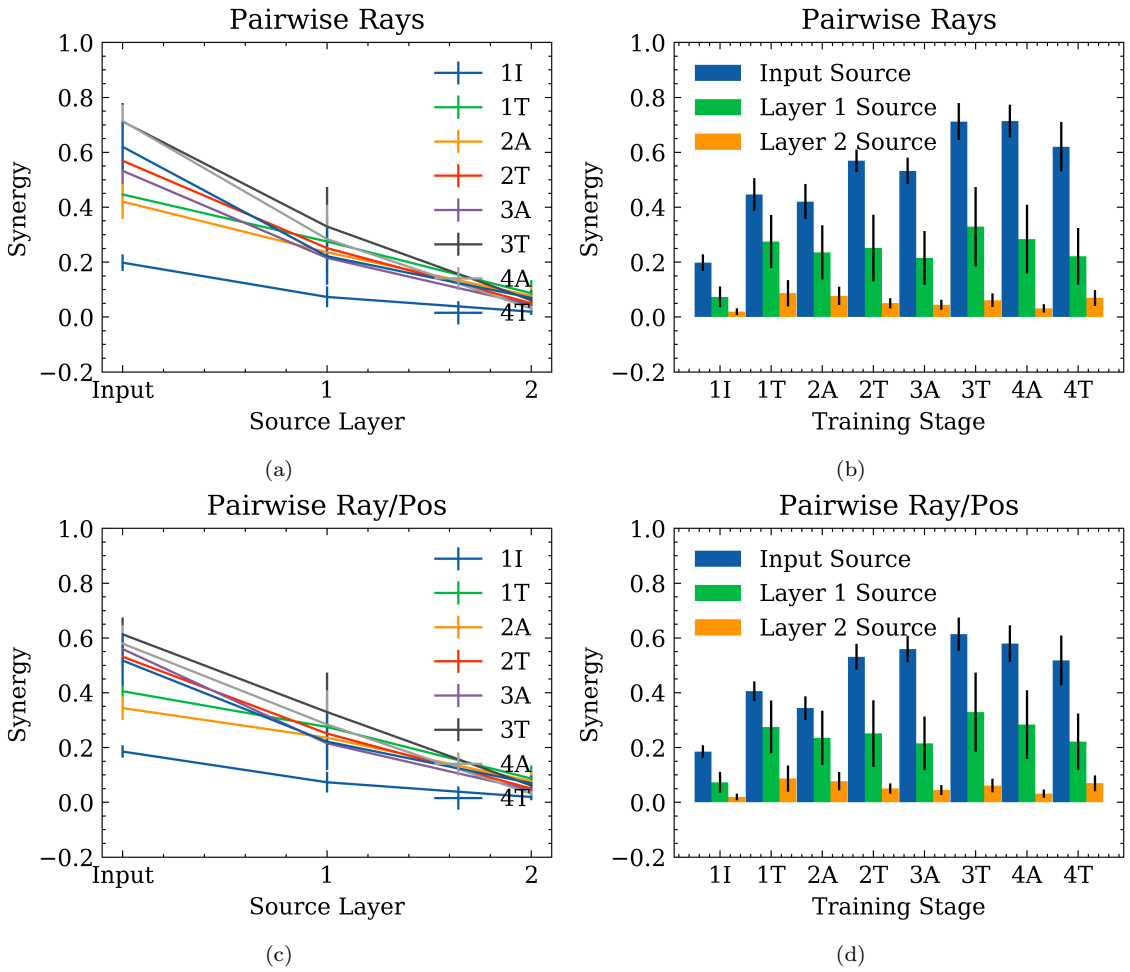
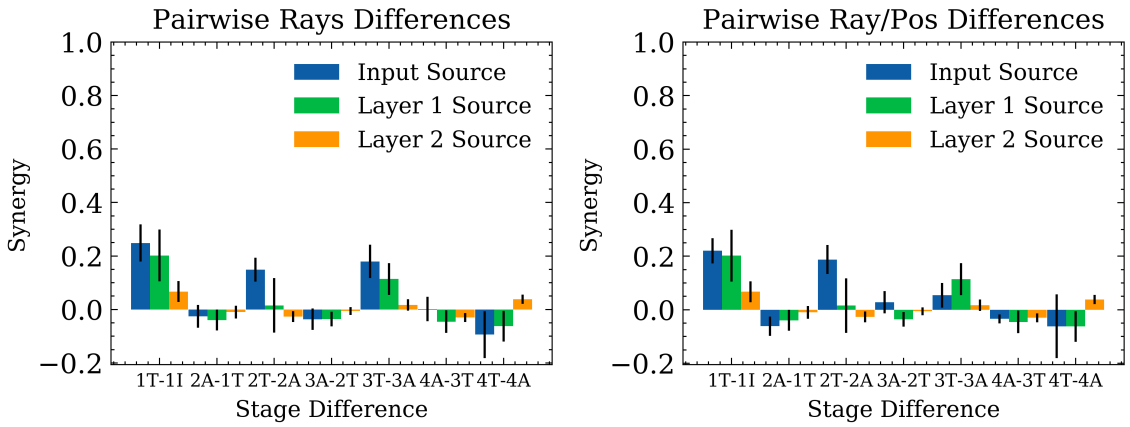


Figure 13: 2-bit increasing distance XOR curriculum synergy between each layer source and the subsequent layer target **(a,b)** with pairwise rays input sources, and **(c,d)** with pairwise ray and position input sources.



(a) (b)

Figure 14: Synergy differences between subsequent training points in the 2-bit increasing distance XOR curriculum for **(a)** for pairwise rays input sources, and **(b)** for pairwise ray and position input sources.

References

[1] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. J. Hudspeth, editors. *Principles of Neural Science*. he McGraw-Hill Companies, Inc., New York, fifth edition, 2013.

[2] Karl J. Friston. Modalities, modes, and models in functional neuroimaging. *Science*, 326:399–403, 2009.

[3] Tomaso A. Poggio, Andrzej Banburski, and Qianli Liao. Theoretical issues in deep networks. *Proceedings of the National Academy of Sciences*, 117(48):30039–30045, 2020.

[4] Quanshi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19:27–39, 2018.

[5] Paul L. Williams and Randall D. Beer. Nonnegative decomposition of multivariate information. *ArXiv*, abs/1004.2515, 2010.

[6] Andrea I. Luppi, Pedro A.M. Mediano, Fernando E. Rosas, Negin Holland, Tim D. Fryer, John T. O'Brien, James B. Rowe, David K. Menon, Daniel Bor, and Emmanuel A. Stamatakis. A synergistic core for human brain evolution and cognition. *bioRxiv*, 2020.

[7] Andrea I. Luppi, Pedro A.M. Mediano, Fernando E. Rosas, Judith Allanson, John D. Pickard, Robin L. Carhart-Harris, Guy B. Williams, Michael M Craig, Paola Finoia, Adrian M. Owen, Lorina Naci, David K. Menon, Daniel Bor, and Emmanuel A. Stamatakis. A synergistic workspace for human consciousness revealed by integrated information decomposition. *bioRxiv*, 2020.

[8] Bernard J. Baars. Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. volume 150 of *Progress in Brain Research*, pages 45–53. Elsevier, 2005.

[9] Stanislas Dehaene, Jean-Pierre Changeux, and Lionel Naccache. The global neuronal workspace model of conscious access: From neuronal architectures to clinical applications. volume 18, pages 55–84, 2011.

[10] George A. Mashour, Pieter Roelfsema, Jean-Pierre Changeux, and Stanislas Dehaene. Conscious processing and the global neuronal workspace hypothesis. *Neuron*, 105:776–798, 2020.

[11] Giulio Tononi. An information integration theory of consciousness. *BMC Neuroscience*, 5:42–64, 2004.

[12] David Balduzzi and Giulio Tononi. Integrated information in discrete dynamical systems: Motivation and theoretical framework. *PLoS Computational Biology*, 4, 2008.

[13] Giulio Tononi, Melanie Boly, Marcello Massimini, and Christof Koch. Integrated information theory: From consciousness to its physical substrate. *Nature Reviews Neuroscience*, 17:450–461, 2016.

[14] Matthew Crosby, Benjamin Beyret, Murray Shanahan, José Hernández-Orallo, Lucy Cheke, and Marta Halina. The Animal-AI testbed and competition. In Hugo Jair Escalante and Raia Hadsell, editors, *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, volume 123 of *Proceedings of Machine Learning Research*, pages 164–176. PMLR, 2020.

[15] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[17] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016.

[18] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing Atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[21] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[22] Cinzia Chiandetti and Giorgio Vallortigara. Intuitive physical reasoning about occluded objects by inexperienced chicks. *Proceedings of the Royal Society B: Biological Sciences*, 278(1718):2621–2627, 2011.

[23] Roger N. Hughes and Christine M. Blight. Algorithmic behaviour and spatial memory are used by two intertidal fish species to solve the radial maze. *Animal Behaviour*, 58(3):601–613, 1999.

[24] Aaron P. Blaisdell, Kosuke Sawa, Kenneth J. Leising, and Michael R. Waldmann. Causal reasoning in rats. *Science*, 311(5763):1020–1022, 2006.

[25] Lucas A. Bluff, Jolyon Troscianko, Alex A. S. Weir, Alex Kacelnik, and Christian Rutz. Tool use by wild new caledonian crows corvus moneduloides at natural foraging sites. *Proceedings of the Royal Society B: Biological Sciences*, 277(1686):1377–1385, 2010.

[26] Edward L. Thorndike. *Animal intelligence; experimental studies*. New York, The Macmillan Company, 1911.

[27] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

[28] Pouya Bashivan, Kohitij Kar, and James J. DiCarlo. Neural population control via deep image synthesis. *Science*, 364(6439), 2019.

[29] Kohitij Kar, J. Kubilius, Kailyn Schmidt, Elias B. Issa, and J. DiCarlo. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature Neuroscience*, 22:974–983, 2019.

[30] Hyodong Lee, Eshed Margalit, Kamila M. Jozwik, Michael A. Cohen, Nancy Kanwisher, Daniel L. K. Yamins, and James J. DiCarlo. Topographic deep artificial neural networks reproduce the hallmarks of the primate inferior temporal cortex face processing network. *bioRxiv*, 2020.

[31] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[32] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015.

[33] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017.

[34] Andrew M. Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. On the information bottleneck theory of deep learning. In *ICLR (Poster)*. OpenReview.net, 2018.

[35] Ziv Goldfeld, Ewout van den Berg, Kristjan H. Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in neural networks. *CoRR*, abs/1810.05728, 2018.

[36] Eugenio Piasini and Stefano Panzeri. Information theory in neuroscience. *Entropy*, 21(1), 2019.

[37] Nicholas M. Timme and Christopher Lapish. A tutorial for information theory in neuroscience. *eNeuro*, 5(3), 2018.

[38] Peter Dayan and Larry F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2005.

[39] Barry E. Stein and Terrence R. Stanford. Multisensory integration: current issues from the perspective of the single neuron. *Nature Reviews Neuroscience*, 9:406–406, 2008.

[40] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[41] Fernando E. Rosas, Pedro A. M. Mediano, Michael Gastpar, and Henrik J. Jensen. Quantifying high-order interdependencies via multivariate extensions of the mutual information. *Phys. Rev. E*, 100:032305, 2019.

[42] Fernando E. Rosas, Pedro A. M. Mediano, Borzoo Rassouli, and Adam B. Barrett. An operational information decomposition via synergistic disclosure. *CoRR*, abs/2001.10387, 2020.

[43] Fernando E. Rosas, Pedro A. M. Mediano, Henrik J. Jensen, Anil K. Seth, Adam B. Barrett, Robin L. Carhart-Harris, and Daniel Bor. Reconciling emergences: An information-theoretic approach to identify causal emergence in multivariate data. *PLOS Computational Biology*, 16(12):1–22, 2020.

[44] Adam B. Barrett. Exploration of synergistic and redundant information sharing in static and dynamical gaussian systems. *Physical Review E*, 91(5), 2015.

[45] Tycho M.S. Tax, Pedro A.M. Mediano, and Murray Shanahan. The partial information decomposition of generative neural network models. *Entropy*, 19(9), 2017.

[46] Larissa Albantakis, Arend Hintze, Christof Koch, Christoph C. Adami, and Giulio Tononi. Evolution of integrated causal structures in animats exposed to environments of increasing complexity. *PLOS Computational Biology*, 10(12):1–19, 2014.

[47] Carlotta Langer and Nihat Ay. Apportionment of work among environment, body and brain of an agent. Preprint, 2021.

[48] Rolf Pfeifer. Morphological computation: Connecting brain, body, and environment. volume 3853, pages 3–4, 2006.

[49] Hagai Attias. Planning by probabilistic inference. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 9–16. PMLR, 2003.

[50] Pedro A. M. Mediano, Fernando E. Rosas, Robin L. Carhart-Harris, Anil K. Seth, and Adam B. Barrett. Beyond integrated information: A taxonomy of information dynamics phenomena, 2019.

[51] John H. Conway. The Game of Life. *Scientific American*, 223(4):4, 1970.

[52] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, page 25–34, New York, NY, USA, 1987. Association for Computing Machinery.

[53] Zenas C. Chao, Yasuo Nagasaka, and Naotaka Fujii. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkeys. *Frontiers in Neuroengineering*, 3, 2010.

[54] Thomas F. Varley and Erik P. Hoel. Emergence as the conversion of information: A unifying theory. *CoRR*, abs/2104.13368, 2021.

[55] Erik P. Hoel, Larissa Albantakis, and Giulio Tononi. Quantifying causal emergence shows that macro can beat micro. *Proceedings of the National Academy of Sciences*, 110(49):19790–19795, 2013.

[56] Erik P. Hoel. When the map is better than the territory. *CoRR*, abs/1612.09592, 2016.

[57] G. Northoff and V. Lamme. Neural signs and mechanisms of consciousness: Is there a potential convergence of theories of consciousness in sight? *Neuroscience & Biobehavioral Reviews*, 118:568–587, 2020.

[58] Stanislas Dehaene and Jean-Pierre Changeux. Experimental and theoretical approaches to conscious processing. *Neuron*, 70:200–227, 2011.

[59] Max E. Tegmark. Improved measures of integrated information. *PLOS Computational Biology*, 12(11):e1005123, 2016.

[60] Adam B. Barrett and Pedro A. M. Mediano. The phi measure of integrated information is not well-defined for general physical systems, 2019.

[61] Pedro A. M. Mediano, Anil. K. Seth, and Adam B. Barrett. Measuring integrated information: Comparison of candidate measures in theory and simulation. *Entropy*, 21, 2019.

[62] Pedro A. M. Mediano, Fernando E. Rosas, Andrea I. Luppi, Robin L. Carhart-Harris, Daniel Bor, Anil K. Seth, and Adam B. Barrett. Towards an extended taxonomy of information dynamics via integrated information decomposition, 2021.

[63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[64] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, 2016.

[65] Stefano Panzeri, Riccardo Senatore, Marcelo Montemurro, and Rasmus Petersen. Correcting for the sampling bias problem in spike train information measures. *Journal of neurophysiology*, 98:1064–72, 2007.

[66] Ryan G. James, Christopher J. Ellison, and James P. Crutchfield. dit: a Python package for discrete information theory. *The Journal of Open Source Software*, 3(25):738, 2018.

[67] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3. `https://github.com/DLR-RM/stable-baselines3`, 2019.

[68] Todd E. Feinberg and Jon M. Mallatt. *The Ancient Origins of Consciousness: How the Brain Created Experience*. The MIT Press, 2016.