

UNIVERSITY COLLEGE LONDON

DEPARTMENT OF COMPUTER SCIENCE

Cognitive Maps in Language Models: A Mechanistic Analysis of Spatial Planning

Author

Caroline BAUMGARTNER

Academic Supervisors

Dr Neil BURGESS

INSTITUTE OF COGNITIVE NEUROSCIENCE
UNIVERSITY COLLEGE LONDON

Dr Eleanor SPENS

INSTITUTE OF COGNITIVE NEUROSCIENCE
UNIVERSITY COLLEGE LONDON
UNIVERSITY OF OXFORD

Dr Petru MANESCU

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY COLLEGE LONDON

September 22, 2025



This dissertation is submitted as part requirement for the MSc Artificial Intelligence for
Biomedicine and Healthcare degree at UCL.

Abstract

Large language models exhibit remarkable spatial reasoning capabilities despite being trained only on next-token prediction, raising fundamental questions about how complex cognitive abilities emerge from simple training objectives. This thesis investigates whether transformers learn generalizable "cognitive maps" or develop task-specific spatial heuristics by conducting the first systematic mechanistic analysis of spatial intelligence in neural networks. Through controlled experiments on three GPT-2 Small models trained on different spatial navigation paradigms, we reveal how training objectives fundamentally shape the computational strategies that emerge during learning.

We compare three distinct approaches to spatial learning: (1) the Foraging Model, trained on random walks to simulate passive exploratory learning; (2) the SP-Hamiltonian model, trained on optimal shortest path demonstrations using structured Hamiltonian contexts; and (3) the SP-Random Walk model, fine-tuned from SP-Hamiltonian on unstructured random walks. Using a comprehensive three-part analysis framework—behavioral evaluation, representational analysis, and causal mechanistic interventions—we dissect the internal algorithms underlying spatial reasoning in each model.

Our analysis reveals two fundamentally different computational strategies. The Foraging Model develops a sophisticated three-stage processing pipeline: early directional processing (Layer 1), spatial integration into a self-sufficient coordinate system (Layer 7), and functional refinement into action-oriented representations (Layers 8-12). Crucially, direction ablation experiments demonstrate a sharp phase transition at Layer 7, where the model becomes causally independent of explicit directional information, indicating the emergence of a genuine cognitive map. The model exhibits adaptive computational strategies, switching between local heuristics for minimal context and global reasoning for extended context, achieving robust generalization to larger grids and novel spatial configurations.

In stark contrast, both SP models employ continuous, path-dependent computational strategies that never achieve self-sufficiency. Despite perfect in-distribution performance, the SP-Hamiltonian model fails catastrophically on unstructured data or novel spatial configurations, exhibiting horizontal mirroring patterns that exploit training distribution regularities but break down under generalization. The SP-Random Walk model shows improved robustness through fine-tuning but retains the same path-dependent algorithm, suggesting that representational adaptation does not always translate to algorithmic change.

These findings demonstrate that training objectives function as algorithmic scaffolding, constraining the space of possible computational strategies. Exploratory learning fosters the development of generalizable, allocentric spatial representations akin to cognitive maps, while goal-directed training produces specialized but brittle heuristics optimized for specific task distributions. The work provides the first mechanistic evidence for the cognitive map versus heuristic distinction in artificial neural networks, revealing how the exploration-exploitation trade-off observed in biological systems applies to transformer architectures.

Our results have significant implications for AI development, suggesting that robust spatial intelligence requires balancing optimization pressures with exploratory learning. The mechanistic insights gained through systematic analysis of underlying computational mechanisms provide a framework for understanding how complex capabilities emerge from simple training objectives, advancing our understanding of intelligence in artificial neural networks.

Acknowledgments

Write any acknowledgments that you wish to include here e.g. supervisors/family/etc. You should also include a mention of any grants, sponsorship or other funding that you have received to help you undertake your MSc course.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Generative pre-trained transformers (GPTs, ?), a type of large language model, often exhibit emergent cognitive capabilities despite being trained on next-token prediction, an objective with no explicit mechanism for planning or reasoning (??). How does this happen? Models trained solely to predict the next word in a sequence can solve complex tasks that appear to require multi-step planning, abstract reasoning, and global understanding. An example of this is in-context learning (ICL), the ability of a language model to learn a new task from examples provided directly in the prompt, without gradient updates or parameter changes (?). This apparent contradiction is central to understanding the nature of ‘intelligence’ in neural networks and has implications for AI safety and development.

Spatial navigation provides an ideal case study for investigating this phenomenon. Accurate navigation requires maintaining knowledge of position, understanding spatial relationships, and planning efficient routes. If a model can navigate effectively, it must have learned something about spatial structure from purely local prediction. We can use spatial reasoning as a lens to examine how complex cognitive capabilities emerge from simple training objectives, and whether different training approaches lead to fundamentally different internal algorithms.

We currently lack a mechanistic understanding of how these capabilities arise. Most research evaluates what models can do rather than how they do it, leaving critical questions unanswered about the computational mechanisms that produce complex behaviour. The central question driving this thesis is: How does local next-token prediction give rise to global spatial understanding? Answering this requires looking inside these models to understand the algorithms they learn and how training paradigms shape those algorithms.

1.2 Research Questions

This thesis addresses one primary research question and several supporting questions that probe different aspects of spatial reasoning in transformers.

Primary Research Question: *Do generative pre-trained transformers learn generalisable ‘cognitive maps’ or develop task-specific spatial heuristics when trained on spatial navigation tasks?*

This question is motivated by the tension between two possible explanations for observed spatial competence. On one hand, the model could develop a cognitive map, an idea originating from cognitive psychology (?). A cognitive map represents a flexible, generalisable internal model of the environment’s spatial relationships—akin to a mental map a person might use to find new shortcuts. On the other hand, the model could rely

on spatial heuristics, which are specialised, rule-of-thumb procedures that work well for specific task distributions but fail to generalise beyond their training regime (e.g., ‘always turn right at a T-junction’).

Secondary Research Questions:

1. *What are the computational circuits underlying spatial reasoning in GPTs?*
2. *How do training objectives and data characteristics shape the internal algorithms that emerge during learning?*
3. *What are the generalisation boundaries of different training approaches?*

These questions probe the mechanistic basis of spatial reasoning, the causal relationship between training data and learned algorithms, and the practical limits of different approaches to spatial learning.

1.3 Research Framework

This thesis investigates spatial reasoning through a comparison of different training approaches. We train identical GPT-2 (?) models on various types of spatial data to understand how training objectives and data characteristics influence the learned algorithms. Our first model, the Foraging Model, establishes a baseline for passive, exploratory learning. Trained to predict the next step in random walks, it learns spatial structure without an explicit goal. In contrast, our two other models represent active, goal-directed learning: models learn to generate shortest paths (SP) between specified start and goal points, developing spatial understanding through planning tasks. The SP-Hamiltonian model learns to find optimal paths between two nodes on a grid using Hamiltonian paths as context—sequences that visit every location in the grid exactly once. This provides the model with complete and highly structured information about the environment. Crucially, we also introduce a hybrid model, fine-tuned from SP-Hamiltonian to perform the same planning task but using unstructured, partial random walks. This third variant allows us to isolate the effects of the training *objective* from the statistical properties of the training *data*.

Our investigation proceeds through three complementary analyses. First, we conduct behavioural analysis to answer ‘What can the models do?’ We designed a suite of experiments to systematically evaluate spatial reasoning capabilities and their boundaries across both in-distribution and out-of-distribution tasks. Second, we perform representational analysis to answer ‘How is knowledge encoded in the models?’ We use principal component analysis (PCA) and linear probing to examine how spatial information is represented in the models’ hidden states, revealing whether spatial information is organised in interpretable patterns. Finally, we conduct mechanistic analysis to answer ‘How do the computational processes work?’ By performing targeted ablations and patching experiments, we move beyond correlation to establish the causal role of individual layers and attention heads, allowing us to trace the computational pathway of spatial reasoning through the network. These interventions provide causal evidence about which components are functionally important and allow us to test hypotheses derived from our other analyses.

1.4 Contributions

This thesis makes several contributions to understanding how transformers develop spatial reasoning capabilities by dissecting the algorithms that emerge from different training

approaches.

Empirical Contributions: We address a key gap in the literature: whilst we know *that* transformers can be trained to perform spatial navigation, we do not yet understand *how* they do it. Our mechanistic analysis identifies two distinct computational strategies learned by the models. The first, emerging from passive, exploratory training, involves consolidating spatial information into a self-sufficient, map-like representation by the middle layers of the network. We find that a single model can adaptively switch between strategies based on context, defaulting to local heuristics when information is limited and engaging its global, map-based system when richer context is available, suggesting a form of hierarchical reasoning. The second, emerging from goal-directed training, relies on a continuous, path-dependent computation that remains reliant on explicit directional inputs throughout all layers.

Theoretical Contributions: This work provides a theoretical account of how training objectives and data interact to determine the learned algorithm. We show that the objective fundamentally determines the type of computational strategy that emerges—for instance, promoting a general world model over a specialised path-following heuristic. Within the context of that strategy, we demonstrate that the statistical properties of the data are a primary driver of the solution’s robustness. Unstructured, exploratory data encourages the development of generalisable representations, mitigating the brittleness induced by training on narrow, structured distributions. This analysis provides a mechanistic basis for the trade-off between specialisation and generalisation, and parallels the exploration–exploitation dilemma studied in cognitive science and reinforcement learning (?). Just as agents must balance exploring novel states to learn a general model of their environment versus exploiting known strategies to achieve immediate reward, our results suggest that the structure of training data and the chosen objective jointly bias the algorithm toward generalisable “exploration” strategies or specialised “exploitation” heuristics.

The work provides insights into the computational requirements for robust spatial reasoning and suggests principles for designing training procedures that encourage the development of generalisable spatial intelligence rather than brittle task-specific heuristics.

1.5 Thesis Structure

This thesis is organised into six chapters that progress from theoretical background through experimental analysis to broader implications.

Chapter 2 reviews the relevant literature on transformer architectures, next-token prediction limitations, and spatial cognition in AI systems. It establishes the theoretical foundation and identifies gaps that this research addresses.

Chapter 3 describes the training framework, data generation, and experimental design. It explains the rationale for our approach and presents the models used in the investigation.

Chapter 4 analyses the Foraging Model, which learns from random walks. It examines the model’s behaviour, internal representations, and computational mechanisms through a series of targeted experiments.

Chapter 5 analyses the Shortest Path models, which learn from goal-directed tasks. It compares their capabilities and internal algorithms with the Foraging Model to understand how training objectives, data characteristics and fine-tuning shape spatial reasoning.

Chapter 6 synthesises the findings, discusses their implications for understanding emergent capabilities in neural networks, and outlines directions for future research.

Chapter 2

Background & Related Work

The ability of large language models to perform complex reasoning tasks, despite being trained on the simple objective of next-token prediction, is a key area of investigation in modern artificial intelligence. This capability raises a question regarding the algorithms learned by these networks: whether they acquire flexible, generalisable internal models of the world, or instead rely on a large set of task-specific, brittle heuristics. This question can be framed by examining its origins in cognitive science and its contemporary investigation through mechanistic interpretability, providing a framework for understanding how different training approaches influence the learning process in transformers. This section reviews the concepts of cognitive maps and emergent world models, contrasting the inductive biases of passive, exploratory learning with the optimisation pressures of active, goal-directed training.

2.1 Planning in Large Language Models

This section focuses on the application of transformers to spatial navigation and planning tasks. The use of simplified, synthetic environments has become a common method for understanding the internal mechanisms of these models. This approach allows for controlled experimentation and detailed analysis of the algorithms that transformers learn. However, this line of inquiry has also revealed limitations of the standard autoregressive paradigm, particularly in tasks that require multi-step reasoning.

2.1.1 Synthetic Worlds as Mechanistic Testbeds for Spatial Reasoning

The use of synthetic grid worlds, mazes, and other algorithmic tasks as experimental environments is a well-established and highly effective framework in AI research for probing the reasoning capabilities of neural networks ??????. These environments offer an advantage over complex, real-world data in that they are fully specified and controllable. This allows researchers to systematically vary task parameters—such as grid size, path complexity, or the amount of available information—and observe the corresponding effects on model behaviour and internal representations. This level of control is essential for performing the kind of rigorous, mechanistic analysis that aims to reverse-engineer a model’s learned algorithm. For instance, Nolte et al. (2024) trained transformers from scratch to navigate mazes of varying complexity, isolating the effect of the training objective on planning capabilities (?).

2.1.2 The Fragility of Planning in Autoregressive Models

While transformers can be trained to perform planning tasks, the standard next-token prediction (NTP) objective, particularly when combined with teacher-forcing during training, has inherent limitations that can lead to non-generalisable solutions ?.

One of the most significant challenges is the problem of compositional generalisation (?). This refers to the ability to combine known components into novel structures to solve new problems. Research has shown that while transformers may learn to execute individual, single-step operations seen during training, they often fail to compose these operations into correct, multi-step algorithms when faced with novel or more complex problem instances ???.

A theoretical explanation for this failure is that teacher-forcing induces shortcuts. Teacher-forcing is the standard training procedure for autoregressive models, where the model is trained to predict the next token, y_t , given the ground-truth prefix, $y_{1:t-1}$. This creates a discrepancy between the training and inference conditions. During inference, the model must generate tokens autoregressively, conditioning its predictions on its own, potentially erroneous, previous outputs. This mismatch, known as ‘exposure bias,’ can lead to a ‘snowballing’ or compounding of errors, where a single mistake can derail the entire generation process, leading to catastrophic failures in long-horizon tasks. While this critique of NTP is well-established, a deeper limitation lies within the training process itself.

Recent work has argued that the teacher-forcing objective actively disincentivises the learning of robust, generalisable algorithms for tasks that require planning or lookahead. As articulated by Bachmann & Nagarajan (2024), NTP encourages the model to learn simple, local heuristics rather than engaging in multi-step planning ?. In lookahead tasks, where a later token in a sequence must be implicitly planned before an earlier token can be correctly generated, teacher-forcing provides a shortcut. By revealing the ground-truth prefix, the model learns to answer the question, ‘What token is statistically likely to follow this prefix of the correct answer?’ instead of, ‘Given the problem statement, what is the first step of the correct solution?’ This creates a ‘Clever Hans cheat,’ where the model appears competent by exploiting superficial cues in the teacher-forced context, but has not learned the underlying algorithm required for robust generalisation.

The reliance on local statistics and shortcuts has been characterised by Dziri et al. (2023) as a process of reducing complex, multi-step compositional reasoning into ‘linearised subgraph matching’ (?). Instead of learning a systematic, rule-based problem-solving procedure, the model learns to match fragments of the current problem to similar sub-problems and their corresponding solution fragments seen during training. This approach can achieve high performance on in-distribution data where such subgraphs are frequent, but it fails to generalise to more complex or novel instances that require a different compositional structure. Dziri et al. (2023) provide a comprehensive analysis of this failure across several compositional tasks, including multi-digit multiplication and logic puzzles. By representing tasks as computation graphs, they quantify complexity via ‘reasoning depth’ (the number of sequential steps) and ‘width’ (the number of parallel items to maintain). They find that performance, even for state-of-the-art models like GPT-4, degrades exponentially with reasoning depth. For instance, while models excel at 2x2 digit multiplication, their accuracy plummets on 3x3 digit problems (?).

In spatial navigation specifically, Nolte et al. (2024) note that transformers trained with a standard NTP objective on maze-solving tasks struggle significantly as maze size and path complexity increase (?). The models often fall prey to the ‘Clever Hans’ shortcuts predicted by theory, learning to follow local cues rather than developing a global plan. Their performance saturates at low accuracy levels for complex mazes, indicating a failure

to generalise the navigation skill.

2.1.3 Alternatives to Next-Token Prediction

In response to the well-documented limitations of standard next-token prediction, researchers have developed alternative training objectives designed to foster more robust planning and reasoning capabilities.

One promising direction is the use of multi-token prediction objectives. Instead of predicting only the single next token, these methods train the model to predict multiple future tokens simultaneously (?). This forces the model to look further ahead, creating a stronger learning signal for long-term dependencies. For example, the MLM-U objective, which involves masking and predicting arbitrary subsets of a sequence, has been shown to improve the ability of transformers to navigate complex mazes compared to standard next-token prediction (?).

This leads back to a more nuanced understanding of goal-conditioned models like the Decision Transformer (?). These models can be viewed not just as a way to apply transformers to RL, but as a solution to the inherent planning failures of standard autoregressive models. A standard next-token predictor is given only a history and must infer the goal implicitly. This is a difficult, under-specified problem. A goal-conditioned model, in contrast, is explicitly provided with the global context it needs to plan effectively.

The choice of training objective can be seen as a form of algorithmic scaffolding, constraining and guiding the search space of possible algorithms that the model can learn. A standard, local, next-token prediction objective provides a backward-looking signal, scaffolding the learning of reactive, path-dependent algorithms that are prone to shortcuts. In contrast, a goal-conditioned objective provides a global, forward-looking signal, scaffolding the learning of planning algorithms that can causally link a current state to a desired future state (?). Meanwhile, an exploratory objective on high-entropy, unbiased data provides a dense signal covering the entire state space. This scaffolds the learning of a complete world model, as building a single, unified representation is the most efficient way to handle diverse and unstructured data.

2.2 Cognitive Maps and Emergent World Representations

The distinction between learning a general environmental model versus specific routes was first investigated in the mid-20th century by psychologist Edward Tolman. Through a series of experiments, Tolman challenged the prevailing behaviourist view that learning was merely the formation of stimulus-response associations. He proposed that animals, specifically rats navigating mazes, construct an internal ‘cognitive map’—a comprehensive, map-like representation of their environment, rather than learning specific responses to individual stimuli (?). In his experiments, rats that were allowed to passively explore a maze without any reward later demonstrated the ability to find efficient, novel shortcuts to a food source when their familiar path was blocked. This behaviour could not be explained by a simple chain of learned motor responses; instead, it suggested the rats had developed a flexible, allocentric (i.e., world-centred and viewpoint-independent) model of the maze’s spatial layout (?).

A key concept arising from this work was *latent learning*: the acquisition of knowledge that is not immediately apparent in an animal’s behaviour but manifests when a suitable motivation or task is introduced. The rats learned the structure of the maze during the exploratory phase, even without reinforcement, and later applied this knowledge in a goal-directed context (???).

This classic concept from cognitive science has found a modern analogue in the study of emergent world representations in large language models. A growing body of research suggests that transformers, even when trained on simple sequence prediction tasks, can develop internal models of the underlying data-generating process. Spens & Burgess (2024) propose a framework where this exact process is simulated by training a GPT-style transformer on various types of sequential data, including navigation path sequences (?). They propose a model of memory consolidation, inspired by the brain, where a large generative network (analogous to the neocortex) is trained by repeatedly replaying sequential experiences that were first stored in a rapid-learning memory buffer (the hippocampus). The core insight is that the simple, self-supervised objective of predicting the next item in a sequence forces the network to extract the underlying statistical regularities and structure from these experiences. Over time, this process distills specific, episodic traces into a generalised schema or ‘cognitive map’ of the environment’s structure. As noted in the paper, this is interesting because decoder-only models, which lack an explicit architecture for latent variables (such as variational autoencoders, (?)), still manage to learn a sophisticated world model implicitly (?).

Another notable study in this area is the work on Othello-GPT, a model trained exclusively to predict legal moves in the game of Othello (?). Despite having no explicit knowledge of the game’s rules or board structure, the model was found to develop an internal representation of the full 8×8 board state. Subsequent investigations revealed that this representation was not only present but also linear and causally linked to the model’s output; by directly intervening on the model’s hidden states to ‘flip’ the representation of a piece on a specific square, researchers could reliably alter the model’s subsequent move predictions to be consistent with the new, counterfactual board state (?). The Othello-GPT experiments provide evidence that a local, predictive objective can lead a model to learn a global, coherent world model as an efficient solution for minimising its prediction error.

Research in reinforcement learning has further explored the idea of ‘world models’, where an agent explicitly learns a generative model of its environment in an unsupervised manner (??). As proposed by Ha and Schmidhuber, this learned world model—a compressed spatial and temporal representation—can then be used to train a much smaller, more efficient policy, sometimes entirely within the ‘dream’ generated by the world model itself (?). Here, a ‘dream’ refers to simulated trajectories produced by the world model, allowing the agent to practice and improve its policy without interacting with the real environment. This explicitly separates the process of world-building (unsupervised, exploratory) from policy-learning (goal-directed, exploitative).

These distinct lines of research converge on a general principle. A passive, next-token prediction objective, when applied to a sufficiently rich and diverse dataset like random walks or game transcripts, appears to create a strong inductive bias towards the formation of a comprehensive world model. The model is not being optimised for any single, narrow goal but for general predictive accuracy across a vast state space. In such a regime, a parsimonious and computationally efficient strategy is not to memorise a vast collection of specific input-output patterns or heuristics, but to learn a single, unified model of the environment’s underlying rules and structure. This learned model of the data-generating process is more generalisable and ultimately requires fewer parameters than a lookup table of surface-level statistics.

2.2.1 Exploration vs. Exploitation

The distinction between exploratory and goal-directed learning can be formalised within the exploration-exploitation framework from reinforcement learning (?). This framework

describes the trade-off an agent faces between exploration—gathering new information about its environment to potentially discover better strategies—and exploitation—using its current knowledge to choose actions that are already known to yield high rewards. An agent that only exploits may get stuck in a suboptimal policy, while an agent that only explores may never capitalise on its discoveries.

This framing of reinforcement learning tasks as sequence modelling problems has gained significant traction with the rise of the transformer architecture. A key development in this area is goal-conditioned reinforcement learning (GCRL), where a policy is learned to achieve specified goals (?). An example is the Decision Transformer (DT), which casts RL as a conditional sequence modelling problem (?). Instead of learning value functions or policy gradients, the Decision Transformer is an autoregressive model that takes a sequence of past states, actions, and a desired return-to-go (the target cumulative reward) as input, and is trained to predict the next action in the sequence. By conditioning on a high desired return, the model can be prompted to generate a sequence of actions that constitutes an optimal policy. This demonstrates that explicitly providing a goal as part of the model’s input is a powerful technique for eliciting specific, optimised behaviours from a sequence model.

2.3 Mechanistic Interpretability: Deconstructing Learned Algorithms

To move beyond observing what a model can do and understand how it does it, the field of mechanistic interpretability (MI) offers a suite of powerful techniques. This research paradigm aims to reverse-engineer the computational processes within neural networks, decomposing their operations into human-understandable algorithms ?. This section reviews the core concepts of MI, including the search for ‘circuits’, the use of causal interventions to validate hypotheses, and the analysis of the geometric properties of learned representations.

2.3.1 Reverse-Engineering Transformer Circuits

The central goal of mechanistic interpretability is to decompose a neural network’s function into its constituent parts, often conceptualised as circuits. A circuit is a specific subgraph of the model’s overall computational graph that is responsible for implementing a particular, human-interpretable function ?. This approach has been successful in identifying such circuits for a variety of tasks. One influential case study is the discovery of the Indirect Object Identification (IOI) circuit in the GPT-2 Small model ?. In the IOI task, the model must complete a sentence like ‘When Mary and John went to the store, John gave a drink to...’ with the correct indirect object, ‘Mary’. Researchers were able to identify a circuit of 26 specific attention heads that work in concert to solve this task. They categorised these heads into distinct functional groups, such as ‘Name Mover Heads’ that copy the correct name to the final position, and ‘S-Inhibition Heads’ that prevent the model from copying the subject’s name. By tracing the flow of information between these components, they constructed a mechanistic explanation of a linguistic capability.

Another foundational discovery in MI is the identification of induction heads ?. These are specialised attention heads that learn to recognise and complete repeating patterns. For example, given a sequence containing the pattern $A \ B$, an induction head will strongly attend to B when it later encounters A , effectively implementing the algorithm $\dots A \ B \ \dots A \rightarrow B$. ?. The emergence of induction heads is hypothesised to be a key mechanism underlying the in-context learning capabilities of large language models ?.

2.3.2 From Correlation to Causation: Probing and Intervening on Representations

Mechanistic interpretability provides a methodology for moving from correlational observations about a model’s internal states to causal claims about its computational algorithm. The first step typically involves correlational methods, which aim to characterise the structure of a model’s internal representations. Techniques such as Principal Component Analysis (PCA) and linear probing are used to project the high-dimensional hidden states of the model into a lower-dimensional space where human-interpretable patterns may become visible (?). These observational techniques are useful for generating hypotheses about what information the model is encoding.

To validate the functional role of these observed representations and the components that produce them, MI relies on causal methods. These techniques involve targeted interventions on the model’s internal activations to test specific hypotheses about information flow. Activation patching, for instance, involves running the model on two different inputs (a ‘clean’ input and a ‘corrupted’ input) and swapping the activation of a specific component (e.g., an attention head’s output) from the clean run into the corrupted run. If this patch restores the correct output on the corrupted input, it provides strong causal evidence that the patched component is necessary for the behaviour in question ?. A simpler form of intervention is ablation, where the output of a component is zeroed out or replaced with a mean value to observe its effect on performance. This progression from observing correlations to establishing causality exemplifies the approach of the MI framework.

2.3.3 The Geometry of Learned Representations

A line of inquiry within MI focuses on the specific geometric structures that emerge within the activation space of transformers. This research posits that models often learn to represent concepts not just as directions in a vector space, but as points on more complex, non-linear manifolds that are mathematically suited to the task at hand.

An example of this principle comes from work on transformers trained to perform modular arithmetic. Nanda et al. (2023) and subsequent work by Kantamneni & Tegmark (2025) showed that these models learn to represent numbers on a generalised helix, a geometric structure composed of multiple circles of different frequencies corresponding to different moduli ????. The model implements addition by performing rotations on this helix, using trigonometric identities to combine the representations of the input numbers ?c. This discovery is significant because it shows that a transformer can learn a non-trivial geometric representation that captures the underlying mathematical structure of the task.

2.4 Synthesis and Positioning of the Current Research

The preceding review highlights a tension in our understanding of transformer capabilities. On one hand, studies on tasks like maze-solving and multi-digit multiplication demonstrate that the standard autoregressive objective often leads models to acquire brittle, non-compositional heuristics. The ‘Clever Hans’ shortcuts induced by teacher-forcing show that models can achieve high in-distribution performance by exploiting local statistical cues rather than learning a robust, underlying algorithm. On the other hand, research on emergent world models, such as the internal board state discovered in Othello-GPT, provides a modern computational analogue to Tolman’s theory of cognitive maps (?). It suggests that a predictive objective on diverse, exploratory data can encourage a model to learn a coherent, allocentric representation of its environment as the most efficient compression of the data.

While these two outcomes—brittle heuristics versus generalisable world models—are well-documented, they have largely been studied in isolation. The critical gap lies in a direct, comparative analysis of the internal mechanisms that produce them. We understand that different data distributions and objectives lead to different capabilities, but we lack a mechanistic account of how the learned algorithms themselves differ.

This thesis is positioned to bridge this gap by conducting a comparative mechanistic analysis. We operationalise the distinction identified in the literature through two controlled experimental conditions:

- The **Foraging Model**, trained on random walks, directly simulates the conditions of passive, exploratory ‘latent learning’ hypothesised to produce cognitive maps. Its objective is purely predictive, lacking an explicit goal.
- The **Shortest Path models**, trained on an optimal planning task, exemplify goal-directed, exploitative learning where the model is optimised to solve a narrow, well-defined problem.

By applying the causal intervention tools of mechanistic interpretability to both model types, this work moves beyond simply cataloguing behavioural differences. The central aim is not to produce a complete reverse-engineering of the learned algorithms, but rather to focus on the mechanistic question of how these different models compute their solutions. By applying causal interventions like activation patching and targeted ablation, we can trace the flow of spatial information and identify critical differences in the computational strategies that emerge from each training objective. In doing so, this research aims to provide a concrete, mechanistic explanation for how training paradigms shape learned algorithms in the domain of spatial reasoning.

Chapter 3

Training Framework and Experimental Methods

To investigate how training objectives and data characteristics shape the emergence of spatial intelligence in transformers, we designed a controlled experimental framework that isolates the effects of different learning paradigms. This chapter describes our approach to modelling spatial navigation, the rationale for our experimental design, the specific training procedures used for each model variant, and the comprehensive methodological toolkit employed for analysis. The methodological sections (3.4-3.8) provide general frameworks that readers may reference when examining the specific experimental applications in subsequent chapters.

3.1 Modelling Spatial Navigation

3.1.1 Theoretical Foundation

We model spatial navigation as a sequence prediction problem over graph structures, following the framework established by Spens & Burgess (2024) and Whittington et al. (2020). In this approach, spatial environments are represented as graphs where nodes correspond to locations and edges represent valid movements between locations. The model’s task is to predict sequences of movements through these graphs, learning to navigate based on the geometric relationships between nodes rather than memorised patterns.

This approach provides several advantages for investigating spatial reasoning. First, it allows us to control the complexity of the spatial environment while maintaining realistic navigational challenges. Second, it enables us to generate near-unlimited training data with known spatial properties. Third, it provides a clear framework for evaluating spatial understanding through both behavioural and mechanistic analyses.

3.1.2 Grid Environment Design

We select 4×4 grid environments as a pragmatic compromise between computational complexity and interpretability. While this choice is somewhat arbitrary, it provides sufficient spatial structure to require genuine reasoning (16 nodes, 24 edges) while remaining tractable for detailed mechanistic analysis. Each grid is represented as a graph where nodes correspond to grid positions and edges represent valid movements in the four cardinal directions (NORTH, SOUTH, EAST, WEST). To encourage the model to learn generalisable spatial rules rather than memorising specific node sequences, we assign random two-letter identifiers to each node in each training example. This design choice prevents the model from learning spurious correlations between node names and spatial positions, forcing it

to rely on the underlying geometric structure. Without this randomisation, the model might learn to associate specific node names with spatial locations, undermining our goal of understanding how spatial reasoning emerges from structural relationships.

3.1.3 Sequence Format and tokenization

Spatial navigation sequences are formatted as alternating node-direction pairs, following the pattern: `node_name DIRECTION node_name DIRECTION . . .`. For example, a sequence might be: `ab EAST cd SOUTH ef NORTH gh WEST`. This format allows the model to learn both spatial relationships (which nodes are connected) and directional information (how to move between them). To ensure clean mechanistic analysis, we trained a custom Byte Pair Encoding (BPE) tokenizer on our synthetic training dataset, so that each node name and direction token is represented as a single token. This prevents the model from learning spurious patterns based on subword boundaries and ensures that our mechanistic interventions target meaningful computational units.

3.2 Training Objectives and Model Variants

3.2.1 Cognitive Science Motivation

Our training approaches reflect two modes of spatial behaviour observed in mammals: exploration and goal-directed navigation. Animals usually first build spatial knowledge through undirected exploration. Rats engage in ‘latent learning’ where they explore novel environments without rewards, developing spatial representations that later enable efficient navigation (??). The Foraging Model mimics this exploratory learning paradigm. Once animals have built spatial knowledge, they engage in efficient goal-directed navigation—returning to nests, finding food sources, or navigating to safety. The Shortest Path models (SP) reflect this goal-directed navigation paradigm. This distinction creates different training curricula for our models. The first is exposed to a broad, unbiased distribution of spatial transitions and is rewarded for correctly predicting any valid movements, which encourages learning the full spatial structure of the environment. In contrast, our SP models are rewarded only for producing direct paths to a goal, exposing them to a smaller subset of the environment’s connectivity. We hypothesise that this narrow data diet, while efficient, may prevent models from forming a complete world model because they never see the suboptimal connections that are crucial for generalisation.

Our experimental design tests whether the exploration-exploitation trade-off observed in biological systems also applies to artificial neural networks. By comparing models trained on exploration versus exploitation, we can observe how different learning modes produce different types of spatial reasoning capabilities.

3.2.2 The Foraging Model: Passive Spatial Learning

The Foraging Model represents our exploration-based approach to spatial learning. This model is trained on random walks through grid environments, learning to predict the next valid step in a sequence of movements. The name ‘Foraging’ reflects the model’s task of navigating through space without explicit goals, similar to how animals explore their environment to build spatial knowledge.

Training Objective

The Foraging Model is trained using standard causal language modelling on random walk sequences. Given a sequence of movements through a grid, the model learns to predict the

next valid direction-node pair. This objective requires the model to understand both the current spatial context (where it is) and the valid movements from that location (where it can go).

Data Generation

The training data consists of 1,000,000 random walk sequences, each containing 120 steps. The 120-step length was chosen to approximate the expected cover time for visiting all 16 nodes in a 4×4 grid, ensuring the model has sufficient context to learn global spatial relationships during training. For each training example, a new 4×4 grid is generated with random node names. A 120-step random walk is then created by starting at a random node and repeatedly selecting valid moves in random directions. This process ensures that the model sees diverse spatial patterns and cannot rely on memorised sequences. The primary advantage of random walks is that they provide an unbiased sampling of the environment’s connectivity. By generating meandering, suboptimal paths, it exposes the model to the full graph of possible movements. This is fundamentally different from a goal-oriented approach, which would only show the model a very small, highly structured subset of paths.

3.2.3 The Shortest Path Models: Active Goal-Directed Learning

The Shortest Path models instantiate our goal-directed, or ‘exploitation-based’, learning paradigm. In contrast to the passive, exploratory learning of the Foraging Model, the SP models are trained on an active planning task: given a context walk that partially (or fully) reveals a grid’s structure, they must generate the optimal path between a specified start and goal node. This objective poses a more complex challenge than ‘foraging’. It requires a two-stage reasoning process: first, the model must infer the underlying spatial graph from the limited context provided; second, it must plan an optimal route within this inferred world model. This directly tests the model’s ability to flexibly use partial spatial knowledge for active, goal-directed problem-solving, much like an animal using its cognitive map to find an efficient new route.

3.2.3.1 SP-Hamiltonian Model

The first SP model variant is trained on shortest path tasks with Hamiltonian context walks. In this setup, the model is given a context walk that visits every node in the grid exactly once, and must learn to predict the shortest path between a specified start and goal node. This setup provides the model with complete nodal information but critically, only partial edge information. In a Hamiltonian path, each node (except the start and end) is seen with only two of its potential four connections (one entry, one exit). Therefore, the task is not a simple string search. To find an optimal path between two arbitrary nodes, the model cannot simply follow connections it has already seen. It must infer the full, unseen grid connectivity from the partial information provided. This design choice forces the model to develop a robust internal representation of the 2D grid structure, using the Hamiltonian path to build a complete world model before planning within it.

Data Generation

Since there are only 552 unique Hamiltonian paths on a fixed 4×4 grid, the data generation process is carefully designed to prevent memorisation and encourage the learning of a general, reusable spatial algorithm. A naive model could attempt two forms of memorisation:

1. **Naive Memorisation:** This involves memorising the exact mapping from an input string ‘context_walk, start_node, goal_node’ to an output string. With a vocabulary of 676 possible two-letter tokens (26×26), the number of ways to choose 16 unique names for the grid positions is given by:

$$P(676, 16) = \frac{676!}{(676 - 16)!} \approx 7.6 \times 10^{31} \quad (3.1)$$

Since there are 552 unique Hamiltonian paths and 240 start-goal pairs per grid (16×15), the total number of distinct examples is approximately 1×10^{37} . Our training set of one million examples is a vanishingly small fraction of this space, making this type of memorisation impossible.

2. **Structural Memorisation:** A more plausible failure mode is for the model to ignore the random node names and instead memorize the underlying *shape* of the Hamiltonian path (the sequence of directions). This would enable a lookup table strategy mapping ‘shape, start_node_index, goal_node_index’ to an output path. The size of this potential lookup table is:

$$552 \text{ shapes} \times 16 \text{ start positions} \times 15 \text{ goal positions} = 132,480 \text{ combinations} \quad (3.2)$$

A 124M parameter model has more than enough capacity to store these 132k key-value pairs. Furthermore, since our training set size (10^6) is far larger than the number of unique structural problems it’s drawn from, the model is almost guaranteed to see every single one of these combinations multiple times.

The data generation for the SP-Hamiltonian model is designed to test structural generalisation. First, we exhaustively enumerate all unique Hamiltonian path ‘shapes’ on a canonical 4×4 grid. This set of all possible shapes is then split, with 90% used for training and 10% held out for evaluation. This ensures that the model is tested on path structures it has never encountered during training. For each training example, the following procedure is used:

1. A new 4×4 grid is procedurally generated with randomised two-letter node identifiers.
2. A Hamiltonian path shape is randomly selected from the training set of shapes.
3. This shape is used to generate a concrete Hamiltonian path on the new grid (i.e. adding the relevant node names), which serves as the context.
4. A random start and goal node are chosen from within this context path.
5. The true shortest path between these nodes is computed using a standard graph search algorithm and formatted as the target sequence.

This methodology forces the model to learn the general geometric principles of Hamiltonian paths on a grid, rather than memorising specific instances. Crucially, we note that since there are 52 solutions starting at a corner space, 25 solutions starting at one of the edge locations, and 36 solutions possible from one of the four center locations, we note that its possible that the model memorises rotations, but

3.2.3.2 SP-Random Walk Model

The second SP model variant is fine-tuned from the SP-Hamiltonian model using random walk contexts of variable length (10-50 steps). This approach tests the model’s ability to extract spatial information from partial, unstructured contexts and perform goal-directed planning within that inferred spatial map. This model is critical to our experimental design as it allows us to ask a more nuanced question: is robust spatial generalisation a product of the passive, exploratory objective of the Foraging Model, or the unstructured, high-entropy data of random walks? Furthermore, it allows us to probe the effects of fine-tuning, examining whether a specialised algorithm can be adapted for greater flexibility when exposed to a new data regime.

Data Generation

Generating valid training data for this model requires creating examples that are both solvable and unbiased. A task is considered solvable if the context walk contains all nodes required to form at least one valid shortest path between chosen start/goal nodes. The challenge stems from our use of short random walk contexts (10-50 steps), which provide a sparse and incomplete view of the grid, far below the ≈ 120 steps of its expected cover time. A naive approach would be to generate a random walk and then picking start/goal nodes from it, but this would heavily bias the dataset towards pairs that are close together. This is because short walks are far more likely to contain the nodes for a short path than for a long one, which would prevent the model from learning to solve more difficult planning tasks. To overcome this, we use a filtering methodology that decouples task selection from context generation. This ensures an unbiased distribution of task difficulties across the dataset. The procedure is as follows:

1. First, a new 4×4 grid with random node names is generated. A start and goal node pair is randomly sampled from the set of 16 nodes, ensuring that tasks of all difficulties (i.e., all possible Manhattan distances) are selected with equal probability.
2. All possible shortest paths between the chosen start and goal nodes are computed. This identifies one or more sets of ‘required nodes’ that must be present in the context for the task to be solvable. Crucially, we do not ensure that all *edges* are present in the context; the model is often required to infer unseen connections to find the optimal route.
3. A fixed context walk length is randomly chosen from the range $[10, 50]$ for the current example.
4. Random walks of the chosen length are repeatedly generated until one is found that contains all nodes from at least one of the required sets. Fixing the context length beforehand prevents bias towards longer walks.
5. Once a valid context is found, one of the shortest paths it supports is randomly chosen as the ground-truth target sequence.

3.2.3.3 Loss Masking

We use loss masking when training SP models to focus learning on path generation rather than predicting the context prompt (?). In standard autoregressive training, a loss is calculated for the model’s prediction at every token position. Loss masking modifies this by selectively ignoring the loss for certain parts of the sequence. For our SP models, where

the input is structured as [Context Walk] [Start/Goal Prompt] [Target Path], we apply a mask to set the loss to zero for all tokens in the [Context Walk] and [Start/Goal Prompt] sections. Consequently, the model’s parameters are only updated based on its accuracy in predicting the [Target Path]. This is necessary because the context-to-task ratio is high: shortest paths on a 4×4 grid average only 4 nodes while contexts contain 16+ nodes. Without loss masking, the model would spend most of its learning capacity on predicting the context walk rather than the target path. This separation allows us to cleanly distinguish between map-building circuits (context processing) and map-using circuits (path planning), which simplifies later mechanistic analysis.

3.3 Training Configuration

3.3.1 Model Architecture

The goal of our work is not to build the most performant or most interpretable navigation model, but to create an experimental setup that is both mechanistically tractable and relevant to the cognitive capabilities emerging in large-scale LLMs. This goal necessitated a careful choice of architecture. A small toy model would offer tractability but lack relevance. A pre-trained, off-the-shelf LLM would offer relevance but is mechanistically opaque due to confounding knowledge from its pre-training data (for example, semantically meaningful 2-letter tokens such as ‘if’ or ‘to’). Therefore, we chose to train the standard GPT-2 Small architecture (124M parameters) from scratch. This approach provides the ‘clean slate’ experimental control of a toy model, ensuring that all learned behaviours stem directly from our data. Simultaneously, it uses an architecture that is a standard, well-understood proxy for larger models, allowing us to generate insights that are both controlled and relevant.

3.3.2 Training Details

Models are trained using standard transformer training procedures with the configuration shown in Table ?? . We experimented with different learning rates, warm-up schedules, and optimisation strategies but found no significant differences beyond minor variations in convergence speed. The models were trained until convergence on their respective objectives.

Table 3.1: Training Configuration for All Models

Parameter	Foraging	SP-Hamiltonian	SP-RW
Batch Size	64	256	128
Learning Rate	5e-4	5e-4	5e-4
Epochs	2	12	12+20
Optimizer	AdamW	AdamW	AdamW
Weight Decay	0.01	0.01	0.01
Warmup Steps	1000	1000	1000
Context Length	120	16	10-50
Training Examples	1M	1M	1M

3.4 Analysis Framework Overview

To understand how spatial intelligence emerges in these models, we employ a systematic three-part analysis approach that progresses from observable capabilities to internal

mechanisms. This framework allows us to address complementary aspects of the central question: how do training objectives shape learned algorithms?

Behavioural Analysis addresses ‘What can the models do?’ by evaluating spatial reasoning capabilities, generalisation boundaries, and task performance across various conditions. This establishes the functional capabilities of each model and identifies interesting phenomena that warrant deeper investigation.

Representational Analysis addresses ‘What knowledge have the models encoded?’ by examining the structure and organisation of internal representations. Through techniques like Principal Component Analysis and linear probing, we can visualise how spatial information is organised across network layers and determine whether interpretable geometric patterns emerge.

Mechanistic Analysis addresses ‘How do the computational processes work?’ by using causal interventions to test hypotheses about internal algorithms. Through activation patching, ablation studies, and other interventional techniques, we move beyond correlation to establish the causal role of different components in spatial reasoning.

This progression from behaviour to representation to mechanism ensures that our investigation is grounded in observable phenomena while building toward mechanistic understanding. The following sections provide detailed methodological frameworks for each analysis type, establishing the technical foundations that will be applied in subsequent chapters.

3.5 Behavioural Analysis Methods

3.5.1 Inference Procedures

The fundamental difference between our models lies in their inference objectives. The Foraging Model performs open-ended spatial prediction: given a sequence of spatial movements, it must predict any valid next step. This tests the model’s ability to maintain spatial awareness and generate locally coherent movements without specific goals.

In contrast, the SP models perform goal-directed inference: given spatial context and explicit start/goal nodes, they must generate optimal paths between specified locations. This tests planning capabilities and the ability to reason about spatial relationships to achieve specific objectives.

Both models use autoregressive generation where each predicted token is appended to the prompt for subsequent predictions. The input format follows the training structure with alternating node-direction pairs (e.g., ‘ab EAST cd SOUTH ef NORTH’). For evaluation on novel grids, we generate new random node identifiers.

3.5.2 Core Evaluation Tasks

Next-Step Prediction (Foraging Model): The fundamental task measures whether models can predict valid next moves given spatial context. The Foraging Model receives a sequence of spatial movements and must predict the next valid node-direction pair. This tests the model’s ability to maintain spatial awareness and generate locally coherent movements without specific goals. Accuracy is measured as exact token matches for valid moves only.

Path Generation (SP Models): SP models generate complete paths between specified start and goal nodes given spatial context. This tests planning capabilities and the ability to reason about spatial relationships to achieve specific objectives. We measure both local validity (each step represents a legal move) and global success (reaching the target destination).

Loop Completion: This task tests abstract geometric reasoning by presenting models with paths that form closed loops, requiring prediction of the final node to return to the starting position. For example, given ‘aa NORTH bb EAST cc SOUTH dd WEST’, the model must predict ‘aa’. The task tests spatial abstraction by requiring the model to understand geometric constraints and complete partial patterns. We test loops of varying sizes (2-12 hops), where a ‘hop’ represents a single directional movement. For example, a 2-hop loop is a simple back-and-forth, while a 12-hop loop represents a 4x4 square. An extension of this is a more complex, Hamiltonian cycle completion, where the path visits every node in the grid exactly once before returning to start, testing whether models develop complete global representations of spatial topology. For SP models, both tasks are adapted by providing start and goal nodes as the loop endpoints. Loop completion and Hamiltonian tasks are particularly interesting because they represent the minimum information required—models must infer edges not explicitly seen in the context and navigate without a global understanding of the grid structure.

Context Length Robustness: We systematically vary the amount of spatial context provided (from minimal 2-step contexts to maximum training length) to identify how performance fluctuates with different amounts of information. This reveals how models adapt their reasoning strategies based on available context, from relying on minimal information to leveraging rich spatial context.

3.5.3 Generalisation Testing

Grid Size Generalisation: Models trained on 4x4 grids are tested on 3x3, 5x5, 6x6 and 7x7 grids to assess whether spatial understanding scales to different environment sizes. This reveals whether learned spatial representations are truly geometric or tied to specific grid dimensions.

Extended Loop Completion: We test NxN square loops on larger grids, where N ranges from 3-7. This tests whether geometric reasoning transfers to novel spatial scales, revealing whether models learn general geometric principles or grid-specific patterns.

High Manhattan Distance Tasks (SP Models): We evaluate shortest-path prediction on 5x5 grids where the start and goal nodes are separated by a Manhattan Distance (MD) of 7–8. The MD between two nodes with coordinates (x_1, y_1) and (x_2, y_2) is defined as:

$$\text{MD} = |x_1 - x_2| + |y_1 - y_2| \quad (3.3)$$

On the 4x4 training grids, the maximum possible distance is MD=6 (a path between diagonally opposite corner nodes). This means the model never encountered paths requiring more than six steps during training. By testing on MD 7–8, we assess whether the model can plan longer routes than it has experienced before, effectively probing its ability to extend multi-step reasoning beyond the training horizon.

Edge-to-Edge Navigation (SP Models): We also test shortest paths between nodes on opposite edges of the 5x5 grid. Unlike the high MD tasks, these paths never exceed the maximum MD seen during training (≤ 6), so the total number of planning steps remains within the model’s experience. However, the paths require consecutive moves in the same direction that cannot fit on the 4x4 training grid, testing whether the model can generalise spatial strategies to larger environments while staying within its learned planning complexity.

3.6 Representational Analysis Methods

3.6.1 Principal Component Analysis

PCA identifies the principal directions of variation in high-dimensional neural representations. Given representations $\{h_i\}$ where $h_i \in \mathbb{R}^d$, we first center them by computing the mean $\bar{h} = \frac{1}{N} \sum_{i=1}^N h_i$, and then calculate the covariance matrix:

$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^N (h_i - \bar{h})(h_i - \bar{h})^\top. \quad (3.4)$$

The principal components are obtained as the eigenvectors \mathbf{v}_k of \mathbf{C} corresponding to the largest eigenvalues λ_k :

$$\mathbf{C}\mathbf{v}_k = \lambda_k \mathbf{v}_k. \quad (3.5)$$

In this case, we extract hidden states from specific layers and token positions across many examples. When tokens appear multiple times in sequences (e.g., node tokens in long random walks), we may choose to average their representations within each sequence to isolate stable spatial encodings from positional effects. The resulting principal components can reveal whether models organise spatial information according to interpretable geometric dimensions, such as coordinate axes or spatial relationships. Clustering patterns in the reduced space indicate whether semantically related spatial concepts are represented similarly.

3.6.2 Linear Probing

Linear probing tests whether specific information can be decoded from neural representations using simple linear transformations. For hidden state $h_l \in \mathbb{R}^d$ at layer l , we train a linear function:

$$\hat{y} = Wh_l + b \quad (3.6)$$

where $W \in \mathbb{R}^{k \times d}$ and $b \in \mathbb{R}^k$ for k -dimensional targets.

Probes are trained on held-out data using cross-validation to prevent overfitting. For regression tasks (e.g., predicting spatial coordinates), we use R^2 scores, defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.7)$$

where y_i are the true values, \hat{y}_i are the predicted values, and \bar{y} is the mean of the true values. High performance indicates that target information is linearly accessible in the representation.

3.7 Mechanistic Analysis Methods

3.7.1 Activation Patching

Activation patching tests causal relationships by replacing activations from one context with those from another. Given two input sequences $x^{(1)}$ (donor) and $x^{(2)}$ (recipient) that differ in specific ways, we replace activation $h_C^{(2)}$ from component C in the recipient sequence with the corresponding activation $h_C^{(1)}$ from the donor sequence:

$$\mathbf{h}_i^{(l)} = \begin{cases} \mathbf{h}_{i,donor}^{(l)} & \text{if position } i \text{ is patched} \\ \mathbf{h}_{i,recipient}^{(l)} & \text{otherwise} \end{cases} \quad (3.8)$$

where the patched model processes the recipient sequence but uses donor activations at specified positions and layers.

Effective patching usually requires minimal pairs—inputs that differ only in the aspect being tested. If patching successfully changes the model’s output in the predicted direction, this provides causal evidence for the component’s role in the computation. We can patch at different granularities: entire layers, attention blocks, MLP blocks, or individual attention heads.

3.7.2 Ablation Experiments

Ablation studies test necessity by systematically removing or zeroing components and measuring performance changes. We distinguish between two types of ablation:

Component Ablation: Tests the necessity of specific network components by zeroing their activations. For a component C at layer l , we modify the forward pass by replacing the component’s output with zeros:

$$\mathbf{h}^{(l)} = \mathbf{h}^{(l-1)} + \begin{cases} \mathbf{0} & \text{if component } C \text{ is ablated} \\ f_C(\mathbf{h}^{(l-1)}) & \text{otherwise} \end{cases} \quad (3.9)$$

where f_C represents the function computed by component C (attention, MLP, or subcomponents). This includes layer-wise ablation (entire layers), attention head ablation, and MLP sublayer ablation.

Token Ablation: Tests the necessity of specific input tokens by zeroing their hidden states at the input to each layer. For token positions T to be ablated at layer l :

$$\mathbf{h}_i^{(l)} = \begin{cases} \mathbf{0} & \text{if } i \in T \\ \mathbf{h}_i^{(l)} & \text{otherwise} \end{cases} \quad (3.10)$$

This allows testing whether the model requires specific input information at different processing stages, revealing when information becomes redundant or when it is still necessary for downstream computations.

We measure ablation effects using the same performance metrics as baseline evaluation. Large performance drops indicate that the ablated component or token is necessary for the behaviour. However, ablation only demonstrates necessity, not the specific computation performed by the component.

3.7.3 Attention Analysis

Attention weights reveal which tokens each position attends to during processing. We visualise attention patterns as matrices showing the attention distribution from each query position to all key positions. Head specialisation analysis examines whether individual attention heads perform consistent functions across examples. We look for heads that systematically attend to specific token types, positions, or semantic categories (e.g., start nodes, direction tokens, corner nodes etc.).

3.7.4 Similarity Metrics

For comparing neural representations across different conditions, we use cosine similarity as the primary metric. Given two vectors \mathbf{u} and \mathbf{v} :

$$\text{cosine similarity} = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}| |\mathbf{v}|} = \frac{\sum_{i=1}^d u_i v_i}{\sqrt{\sum_{i=1}^d u_i^2} \sqrt{\sum_{i=1}^d v_i^2}} \quad (3.11)$$

This ranges from -1 (opposite directions) to 1 (identical directions), with 0 indicating orthogonal vectors. Cosine similarity captures directional similarity while being invariant to magnitude differences across layers or conditions, making it particularly useful for comparing neural representations that may have different activation scales.

3.8 Limitations and Caveats

Several important limitations apply to mechanistic interpretability methods. PCA only reveals linear structure in representations, potentially missing non-linear organisation. Linear probing only tests linear accessibility, not whether models actually use information in that format. Activation patching may miss distributed computations requiring coordination across multiple components. Representational analyses (PCA, probing) reveal correlational structure, while mechanistic analyses (patching, ablation) can establish causal relationships. Successful patching demonstrates sufficiency while successful ablation demonstrates necessity. Results may be specific to the particular architecture, training procedure, or task studied. Finding interpretable patterns does not guarantee understanding of the model’s true computational strategy. Models may use computations that appear interpretable but serve different functions than assumed.

3.9 Chapter Summary

This chapter establishes both the training framework and methodological toolkit for investigating spatial intelligence in transformers. The training framework compares two paradigms: exploratory learning through random walks (Foraging Model) versus goal-directed learning through shortest path planning (SP Models). This controlled comparison isolates the effects of training objectives on learned algorithms.

The three-part analysis framework progresses from behavioural evaluation to representational analysis to mechanistic investigation. behavioural methods establish model capabilities and identify interesting phenomena. Representational methods reveal how spatial information is organised internally. Mechanistic methods use causal interventions to test hypotheses about computational processes.

In subsequent chapters, we apply this toolkit to understand how different training paradigms produce different forms of spatial intelligence, moving from observable capabilities to internal mechanisms to establish how training objectives shape the algorithms that emerge.

figures/tasks.png

Figure 3.1: **Example tasks for both models.** Left: Foraging Model training uses random walks as context (left) and predicts valid next steps (right, red arrows). Right: SP-H training uses Hamiltonian paths as context (left, blue arrows) and predicts shortest path between start (red) and end (green) nodes (right, multiple valid paths shown).

Figure 3.2: Examples of core evaluation tasks. (a) Foraging Model next-step prediction: given context walk, predict valid next direction. (b) SP Model path generation: given context walk and start/goal nodes, generate shortest path. (c) Loop completion: complete closed loop by predicting return node. (d) Hamiltonian cycle completion: complete cycle visiting all nodes once. (e) High Manhattan Distance task: find path exceeding training complexity. (f) Edge-to-edge navigation: find path between opposite edges of larger grid.

Chapter 4

The Foraging Model

How does a model trained only to predict the next token in random walks develop sophisticated spatial reasoning? This chapter investigates spatial navigation in its simplest form: passive information integration through exploratory movement. The Foraging Model represents our exploration-based approach to spatial learning, trained solely on random walks without explicit goals. We trace the model’s journey from local pattern matching to global spatial understanding through behavioural, representational, and mechanistic analyses.

The Foraging Model captures exploratory learning by training on random walks through grid environments. Unlike goal-directed navigation, this paradigm requires no explicit planning objective—the model simply learns to predict the next valid step in a sequence of movements. This minimal training setup allows us to isolate the core mechanisms of spatial learning and establish a baseline for understanding how different training approaches shape spatial reasoning.

Our investigation addresses several key questions: Can a model trained solely on random walks develop robust spatial understanding? What internal representations emerge to support spatial reasoning? How does the model transition from local pattern matching to global spatial understanding? By answering these questions, we establish the foundation for comparing different approaches to spatial learning in subsequent chapters.

4.1 Behavioural Analysis

We evaluate the Foraging Model’s spatial reasoning capabilities using the evaluation framework described in Chapter 3, assessing core performance metrics, context length robustness, and generalisation capabilities. Here, the ‘context’ is the historical sequence of nodes and directions provided as the input prompt from which the model must infer its current position.

4.1.1 In-Distribution Tasks

We begin by assessing the model’s performance on its core training objective: next-step prediction on 4×4 grids. The Foraging Model achieves 98.8% accuracy when predicting a valid move (direction and corresponding node) with a context length of 110 nodes (see Figure ??A). However, when predicting only the next node (given both the current position and direction to move in), the model achieves perfect accuracy (100%) across all context lengths.

The model’s capabilities extend to tasks requiring global spatial reasoning that cannot rely on sequence memorisation. On Hamiltonian cycle completion, the model receives a complete path visiting all 16 nodes exactly once and must predict which node completes

the cycle back to the starting position, achieving perfect accuracy (100%). On simpler loop completion tasks, performance remains perfect (100%) for patterns of 2-12 hops. Both tasks require inferring spatial relationships that may never have appeared in training, as the model must understand geometric constraints rather than rely on memorised sequences.

In multistep generation, the model shows a gradual decline in accuracy. When continuing 110-step context paths, performance drops from 98.8% for 1-step generation (direction-node pair) to 97.6% for 2-step generation and 94.2% for longer continuations (10 steps). This decline is unsurprising, and reflects the absence of self-correction mechanisms—errors compound across generation steps without opportunity for recovery.

4.1.2 Context Length Robustness

One of the most striking behavioural findings is the Foraging Model’s non-monotonic relationship between performance and context length (Figure ??A). This pattern suggests that the model adapts its computational strategy depending on the amount of information available: short, intermediate, and long contexts elicit qualitatively different behaviours.

To understand this transition, we analyse the model’s reverse bias (its tendency to reverse the last move) which provides insight into the underlying reasoning strategy. Formally, given a walk ending with direction d , let d_{rev} be its reverse (e.g., SOUTH for NORTH) and \mathcal{D}_{valid} be the set of valid moves from the current node. The reverse bias B is defined as:

$$B = P(d_{rev}) - \frac{1}{|\mathcal{D}_{valid}| - 1} \sum_{d' \in \mathcal{D}_{valid} \setminus \{d_{rev}\}} P(d') \quad (4.1)$$

where $P(d)$ is the model’s predicted probability for direction d . We computed this bias across 500 random walks for each context length from 2 to 115 nodes.

The analysis reveals three regimes:

1. **Minimal context (2–3 steps):** The model achieves near-perfect accuracy (100%), but exhibits a strong preference for reversing the last move (reverse bias = 0.38). Here, the model relies on local heuristics: given very limited context and no global information about the grid, the safest move is often simply to reverse the previous step.
2. **Intermediate context (5–40 steps):** Accuracy decreases (70–85%), while reverse bias drops (0.16 at 11 steps). This regime may represent a transitional phase where the influence of local heuristics is reduced, and the model is exposed to a greater diversity of valid moves. Because the training data consisted of random walks with equal probability for each valid move, the model experiences a pull toward more uniform predictions across directions. This dip likely reflects the fact that intermediate-length walks are long enough to exceed minimal local heuristics but still too short to uniquely identify their position on the grid, creating uncertainty in the model’s predictions.
3. **Long context (40+ steps):** Performance recovers to near-perfect levels (> 97%), with reverse bias approaching zero. With longer context, the model can condition its next-step predictions on a larger portion of the trajectory, producing consistent, context-informed behaviour that aligns with the overall walk structure.

In summary, the reverse bias analysis provides insight into how the model adapts its reasoning with context length. Short contexts favour local heuristics, intermediate contexts reflect a combination of heuristic and task-driven uncertainty, and long contexts allow robust, context-dependent navigation. While the model shows robust spatial generalisation, it exhibits sharp temporal boundaries. Performance remains high until the context length approaches the training limit of 120 steps, where it drops to 14% accuracy.

4.1.3 Generalisation Performance

Despite training exclusively on 4×4 grids, the model generalises effectively to larger environments, though its success depends critically on the nature of the task (Figure ??B).

The model is notably better at tasks that have a deterministic, geometrically constrained answer. Performance on Square Loop Completion remains nearly perfect on grids up to 5×5 (100%) and only degrades slightly on 7×7 grids (80.5%). This makes sense. Completing a square loop is a puzzle with a single, geometrically correct answer. The model’s robust performance shows it has learned an abstract spatial reasoning that transfers well to larger, unseen environments.

In contrast, its performance is weaker on tasks with stochastic, open-ended answers. Continuing a random walk on a large, unfamiliar grid is hard because there are often multiple valid moves. This forces the model to predict a direction where uncertainty is high, leading to a faster drop in accuracy as grid size increases (down to 63.5% on 7×7 grids). This confirms that the model struggles more with predicting the next direction than the next node. Crucially, we can be confident this performance drop is not simply due to the autoregressive challenge of predicting two tokens (direction and node) versus one (node completing the loop). As established in our in-distribution analysis, the model achieves perfect 100% accuracy on node-only prediction across all context lengths. This demonstrates that once a direction is given, the model deterministically infers the resulting node. Therefore, the weaker performance on random walk continuation stems directly from the uncertainty of the directional decision itself, not from compounding errors during generation.

4.1.4 Summary

The behavioural analysis reveals that the Foraging Model achieves strong performance across multiple spatial reasoning tasks. The model demonstrates perfect accuracy on node-only prediction tasks, maintains high performance on its training objective, and shows robust capabilities on abstract spatial reasoning tasks such as Hamiltonian cycle completion and loop completion. The model exhibits a non-monotonic relationship between performance and context length, with graceful degradation when generalising to larger grid sizes. These findings raise questions about the internal representations that support such performance. To investigate this, we next turn to a representational analysis of the model’s hidden states.

4.2 Representational Analysis

We examine the model’s internal spatial representations using PCA and linear probing, as detailed in Chapter 3. For PCA analysis, we sample from 1,000 random walks on unique 3×3 grids, averaging node representations across occurrences to control for positional effects. For linear probing, we train probes on 500 examples per layer to predict true (x,y) coordinates from hidden state representations of node tokens.

4.2.1 Principal Component Analysis

PCA reveals a three-stage evolution in the model’s spatial representations (Figure ??). Early layers (1–3) show a basic, noisy sense of coordinates, reflecting initial processing of local syntactic and positional information without coherent spatial structure. A transformation occurs around Layer 7, where node representations organise into clear spatial patterns. Consistent with Spens & Burgess’ findings, the top two principal components align closely with the grid’s x and y axes (cosine similarity ≈ -0.0415), indicating development of an orthogonal, Cartesian-like coordinate system that represents spatial position independently of the specific path taken to reach it (?). At this stage, the model has a robust representation of where each node is. The clean coordinate system dissolves in late layers (11–12) and shifts to functional clustering based on navigational affordances (Figure ??). The organizing principle shifts from spatial position to geometric function, with nodes clustering by their possible moves. Corner nodes, each having a unique pair of two available directions (e.g., south and east), form four distinct and well-separated clusters. Edge nodes also form four groups, with nodes positioned along the same edge clustering together. For example, all nodes on the top edge (from which one can move east, west, and south) form one group, separate from the nodes on the left edge (from which one can move north, south, and east). Center nodes, which are functionally identical as all four directions are available from them, converge into a single, tightly entangled group. This transformation suggests the model’s final layers are not just tracking location but are computing a more abstract, action-oriented representation. This transition is consistent with the broader pattern in deep learning systems, where mid-layer representations preserve structured embeddings, and later layers reorganise them into task-relevant abstractions. For navigation, functional clustering by available moves is a more directly useful basis for prediction than spatial position.

4.2.2 Linear Probing

Linear probing confirms the PCA findings with quantitative precision. The R^2 score increases steadily through early layers, plateaus around Layer 5 ($R^2 \approx 0.9$), and reaches its peak at Layer 8 ($R^2 \approx 0.93$). This confirms that the model develops a robust, linearly-decodable coordinate system that stabilizes in the middle layers. The plateau at Layer 7 aligns precisely with the PCA results, providing converging evidence for when the spatial representation becomes established. However, a high R^2 score only proves that coordinate information is linearly present in the representation. It is a strong correlation, but it is not causal proof that the model makes use of this coordinate system. The continued high decodability through later layers indicates coordinate information persists even as the representation transforms toward functional clustering.

4.2.3 Summary

These observational findings point to a three-stage pipeline: (1) processing of local information, (2) integration into a coordinate map, and (3) refinement into an action-oriented map. However, these analyses are purely correlational; they show that spatial information is present and linearly decodable, but not that the model causally relies on it in this form. We now use causal interventions to test this hypothesis and uncover the underlying computational mechanism.

4.3 Mechanistic Analysis

Our representational analysis revealed the emergence of structured spatial representations, but correlation does not imply causation. To understand how the computational processes actually work, we now perform targeted interventions that manipulate information flow through the network. The goal of this mechanistic analysis is twofold: first, to test the specific hypotheses generated by our observational findings, and second, to uncover the underlying circuits that implement them. We will systematically dissect the model’s computation to establish the causal role of different components in its spatial reasoning algorithm.

4.3.1 Localising the Direction Update Circuit

Our investigation began with searching for the simplest mechanism underlying spatial reasoning: the circuit that processes a single direction token (e.g., ‘EAST’) and updates the model’s internal spatial state. Understanding where these computations occur can reveal how the model processes directional information and updates its internal representation of space. We started with the simplest hypothesis: the model implements ‘go east’ types of instructions as vector addition in MLPs.

4.3.1.1 Minimal Pair Activation Patching

To test our hypothesis, we constructed minimal pair prompts—identical random walk sequences except for their final direction token (e.g., one ending with ‘EAST’ and another with ‘WEST’). The goal was to see which component, if any, could successfully compute the directional update. We then systematically patched activations from one prompt into the other to identify which components could successfully compute the directional update.

Using the activation patching methodology outlined in Chapter 3, we tested MLP outputs across all layers first. This failed completely, forcing us to reject our initial hypothesis, and suggesting that a more complex mechanism was at play. The failure of the MLPs is theoretically expected. By design, MLP layers process information at each token position independently. However, because node names are randomized in every sequence, this task is an instance of In-Context Learning (ICL). The model cannot memorize that ‘ab is west of cd’; it must infer this relationship from the context of the current prompt. The only mechanism capable of moving information between token positions to establish such context-dependent relationships is *attention*. Therefore, we shifted our search to the attention blocks. Patching the attention output of Layer 1 successfully and consistently controlled the model’s prediction, establishing this component as causally responsible for processing directional information.

Because both prompts only differed in their final direction, this finding raised a key ambiguity: was the output a fully resolved state (‘the next node is X’) or an abstract instruction (‘go east’)?

4.3.1.2 Cross-Context Transfer Analysis

To resolve this ambiguity, we conducted a cross-context patching experiment using completely independent random walks from different grids. We took the hidden state generated by a direction token in one context (e.g., the vector for EAST in ‘... ab EAST’) and patched it into a completely different context (e.g., replacing the vector for SOUTH in ‘... yz SOUTH’). For each trial, we constructed pairs of prompts:

$$\begin{aligned} P_{donor} &= "...n_i \text{ DIRECTION}_1" \rightarrow m_1 \\ P_{recipient} &= "...n_j \text{ DIRECTION}_2" \rightarrow m_2 \end{aligned} \tag{4.2}$$

where n_i, n_j are different nodes, $\text{DIRECTION}_1 \neq \text{DIRECTION}_2$ are different movement directions, and m_1, m_2 are their respective valid next nodes. We extracted the hidden state vector h_{donor} from Layer 1’s output at the DIRECTION token position in P_{donor} and patched it into the same position in $P_{recipient}$:

$$h_{recipient}^{patched} = \begin{cases} h_{donor} & \text{at DIRECTION token} \\ h_{recipient} & \text{otherwise} \end{cases} \tag{4.3}$$

The patched vectors successfully redirected predictions in 97.4% of trials (N=1000). This suggests that Layer 1 computes a near-universal, transferable instruction (like a ‘go east’ command) rather than a node-specific update (like ‘the result of going east from ab’). However, 2.2% of trials produced the donor context’s next node and 0.4% produced unexpected outputs, indicating the representations contain both abstract and context-specific components. While the high success rate suggests Layer 1 often computes directional information that transfers across contexts, the mixed outcomes indicate the representation likely contains both abstract directional components and context-specific elements, rather than being purely universal instructions. Most likely, the model’s internal components are not perfectly modular, reflecting the emergent nature of its capabilities.

4.3.1.3 Head Redundancy Analysis

Finally, a head redundancy analysis showed that 6 of 12 Layer 1 heads could independently drive 60%+ cross-context transfer when isolated (all other heads zeroed out, patched over 1000 trials, see Figure ??B). This indicates a distributed, partly redundant mechanism: rather than a single specialised ‘direction head’, multiple heads encode overlapping variants of the same computation. Such redundancy likely reflects the model’s overparametrisation (GPT-2 small has far more capacity than this task requires) so directional processing is spread across several heads instead of being compressed into a minimal circuit.

4.3.2 Testing the Three-Stage Hypothesis

Having localised basic directional processing and observed structured spatial representations, we now test our hypothesis about the three-stage processing pipeline through targeted causal interventions.

4.3.2.1 Direction Token Ablation

Our representational analyses showed the model forms a stable coordinate system by Layer 7. However, these are correlational findings—they show information is present but not that it’s causally used. To test when (if ever) the coordinate system becomes self-sufficient, we examine when the model no longer requires explicit directional information. We conduct a layer-wise direction token ablation, stratified by the complexity of the task. We use controlled loop completion tasks of varying lengths (2, 4, 6, 8, 10, and 12 hops), where a path returns to its starting node (e.g., ‘aa NORTH bb WEST cc SOUTH dd EAST’). The model’s task is to predict the final node (aa) given the prompt. The intervention involves zeroing out the hidden states of all historical direction tokens at the input to each layer, one by one. The final direction token is always preserved. This design choice isolates

the effect of historical memory from the effect of the immediate, necessary input. This approach, detailed in Chapter 3, tests whether spatial information has been ‘absorbed’ into node representations.

The intervention reveals that the model employs at least two distinct strategies depending on task complexity (Figure ??):

1. **2-Hop Loops:** The simplest, 2-hop loops exhibit qualitatively different behaviour. Performance recovers to 65% when ablating input to Layer 1 and reaches 100% by Layer 2. A 2-hop loop is a simple ‘reverse the last move’ pattern that does not require a global map (e.g., ‘xq EAST fc WEST’). Consistent with our reverse bias analysis, this suggests the model uses a specialised, low-level circuit in its earliest layers to solve these trivial local patterns without engaging its more complex spatial reasoning mechanisms.
2. **4-12 Hop Loops:** For all loops requiring more complex spatial reasoning (4+ hops), we observe a sharp, but slightly staggered, phase transition between Layers 6 and 8.
 - **Layers 1–6:** Ablating historical directions results in near-complete failure, confirming that the model relies on these explicit tokens to build its spatial representation in the early and middle layers.
 - **Layer 7:** This marks the critical point of transition. Performance for all complex loops jumps dramatically.
 - **Layer 8:** By Layer 8, performance for all complex loops, including the longest 12-hop variants, recovers to perfect accuracy. This is the point at which the internal coordinate system is fully self-sufficient and robust enough to solve even the most complex tasks without needing the original direction tokens from the path.

The sharp transition at Layer 7 indicates that the spatial representation becomes self-sufficient—coordinate information has been fully absorbed into node states, making original direction tokens redundant. An important distinction emerges between our correlational and causal measurements. The apparent contradiction between the linear probing plateau at Layer 5 ($R^2 \approx 0.9$) and the direction ablation transition at Layer 7 reveals how the model processes spatial information. Linear probing shows whether coordinate information is linearly decodable in hidden states—a correlational measure. Direction ablation tests whether the model causally depends on explicit direction tokens—a causal measure. Between Layers 5–7, the model likely uses two sources of spatial information: position encoded in node hidden states (detectable by probes) and explicit direction tokens. This underscores how mechanistic interpretability requires complementary techniques: neither measure alone captures the full picture of spatial reasoning in the transformer.

4.3.2.2 Direction Swapping

Having established coordinate system formation by Layer 7, we test whether late layers simply preserve coordinates or refine them into functional representations. Based on our PCA results showing functional clustering, we hypothesise that the model learns not just *where it is* but *where it can go* from that position.

We systematically reverse all direction tokens in random walks (NORTH↔SOUTH, EAST↔WEST) and examine how representations respond to these geometrically mirrored paths. Note that swapping directions creates a new path that is a geometric mirror image of the original (sequence with identical node names and step counts but a completely different spatial trajectory). This allows us to test if the model’s understanding is based on the

geometric meaning of the path or just superficial token patterns. We compare responses at constrained versus unconstrained positions across 100 trials, focusing on how the model’s behaviour varies based on geometric context: specifically comparing unconstrained centre positions (all four directions available), edge nodes (three available directions) and constrained corner nodes (only two directions available), using the intervention methodology from Chapter 3.

This experiment provides a causal explanation for the functional clustering observed in the late-layer PCA. The model’s response to the mirrored path history depends entirely on the geometric context of the final node (Figure ??). At unconstrained positions (i.e. centre node), where the path taken is irrelevant to the four available moves, the hidden states for original and swapped sequences converge to be nearly identical (cosine similarity ≈ 0.99). This reveals the model learns to discard unnecessary historical details, which explains why all centre nodes collapsed into a single representation in the PCA. Conversely, at constrained corner nodes, where path history is critical for determining the two valid moves, the hidden states remained distinct (cosine similarity ≈ 0.31). Edge nodes show intermediate behaviour (cosine similarity ≈ 0.70), consistent with their intermediate geometric constraints (3 available moves).

This differential sensitivity emerges specifically during layers 8-10, coinciding with the shift from coordinate to affordance clustering observed in our PCA analysis. The synchronised timing of cosine similarity drops and representational reorganisation points to a computational phase transition. The model refines representations to emphasise functionally relevant distinctions while allowing irrelevant variations to wash out. Functionally equivalent states (centre node) naturally converge during this refinement process, while functionally distinct states (edges and corners) naturally diverge, producing the observed clustering by navigational affordances.

4.3.3 Attention Pattern Analysis

To explore the mechanisms that might underlie the model’s spatial reasoning, we analysed its attention patterns, focusing on the computation that occurs when predicting the final node in loop completion tasks. We visualised how attention heads allocate weights across sequence positions when processing the final direction token, across different loop lengths (2-12 hops). Across all loop complexities (2–12 hops) tested, we observe a consistent pattern in Layer 1 when querying the final direction token: approximately 6 heads almost exclusively focus attention on the penultimate node, i.e. the node token that appears before the final node in the sequence (Figure ??). Rather than processing static positional information, the model seems to encode trajectory by attending to a two-step history. This pattern holds regardless of task complexity, suggesting a universal low-level circuit for spatial perception which may form the foundation for all subsequent spatial reasoning.

This simple Layer 1 circuit elegantly explains the model’s behaviour on 2-hop loops observed in our direction ablation experiments (Figure ??). In a 2-hop loop (e.g., ‘aa EAST bb WEST’), the penultimate node is simply the origin (‘aa’). By attending to it, Layer 1 provides a direct signal for the required reverse move. This creates a highly efficient heuristic that allows the model to solve these trivial cases in its earliest layers, bypassing the need for a global map. This explains why performance recovers almost immediately by Layer 2 even when the first direction token is ablated, and accounts for the high accuracy and strong reverse bias observed at minimal context lengths. Concurrently, other heads in Layer 1 exhibit a different kind of specialisation based on token type. We observe that certain heads, such as Head 4 and Head 8, selectively attend to either only node tokens or only direction tokens.

For longer, more complex loops, 2-step information alone is insufficient to find a so-

lution. It may instead serve as the initial input to a deeper computational process that unfolds across the later layers of the network. While tracing this process in detail is beyond the scope of our current analysis, we can observe its potential outcome in the model’s final layers. In the final layers (10–12), the computational focus shifts towards action selection, which is expected. This is most evident in the strong self-attention that multiple heads pay to the final direction token itself (Figure ??), likely preserving information for the final output. This specialisation aligns with our finding that late layers implement action-oriented representations.

Taken together, these observations suggest a coherent, though interpretive, mechanistic story. The model may not learn two entirely distinct circuits, but rather one foundational local rule that can be used in two different ways. For simple problems, this rule may be sufficient for a fast, heuristic solution. For more complex problems, it could serve as the building block for a more deliberative, multi-layer computation.

4.3.4 Summary

The mechanistic interventions provide strong causal evidence for our hypothesised three-stage pipeline. We localised a direction-updating circuit in Layer 1 that computes universal movement instructions, demonstrated that by Layer 7 the model relies on a self-sufficient coordinate system, and revealed sophisticated context-dependent processing in the final layers. Having characterised the components of the learned algorithm, we can now assemble these findings to form a complete picture of how the Foraging Model solves its task.

4.4 Chapter Discussion

4.4.1 Three-Stage Processing Pipeline

Our experiments provide initial evidence for how the Foraging Model may achieve spatial reasoning, though the precise computational mechanisms remain partly opaque. The model achieves impressive in-distribution and generalisation performance through what appears to be a three-stage process involving early directional processing, middle-layer spatial integration, and late-layer functional refinement. We emphasise that this ‘three-stage’ description is a useful simplification of what is likely a more distributed and continuous process.

Stage 1 - Directional Processing (Layer 1)

Layer 1 attention implements the most interpretable component of the spatial reasoning pipeline. Cross-context patching transfers directional instructions successfully in 97.4% of trials, providing strong evidence that this layer computes abstract movement operations that generalise across contexts. The high transfer rate suggests these representations capture something closer to universal directional concepts rather than position-specific updates. However, the 2.6% failure rate (where patches produce unexpected outputs or donor context nodes) suggests the computation is not perfectly modular. These failures may reflect interference between abstract directional processing and context-specific information, or they may indicate that our experimental design incompletely isolates the relevant circuits. The distributed nature of this processing (across multiple redundant heads rather than a single specialised circuit) likely reflects the model’s overparameterisation relative to task requirements.

Stage 2 - Spatial Integration (Layers 2-7) Directional updates are progressively integrated into spatial representations. The emergence of coordinate-like organisation in middle layers is distinctive in PCA plots, with the first two principal components aligning surprisingly well with row and column structure. This pattern is consistent with the development of allocentric spatial representations, and shared structure across different experiences, which was also noted by Spens & Burgess in their initial work (?). Linear probing shows gradual improvement in coordinate decodability ($R^2 \approx 0.15$ at Layer 1 to $R^2 \approx 0.93$ at Layer 8), confirming that coordinate information becomes linearly accessible. Critically, the direction ablation experiment provides causal evidence for when this coordinate system becomes functionally sufficient. Performance on complex loop completion tasks (4-12 hops) jumps from near-zero to perfect when historical direction tokens are removed at Layer 7 input, demonstrating that spatial relationships have been consolidated into node representations by this point. This suggests genuine computational restructuring rather than mere representational reorganisation. We use the term ‘coordinate system’ loosely here, since the model was not trained to map nodes to specific coordinates, and is more likely encoding relative positions and geometric constraints.

Stage 3 - Functional Refinement (Layers 8-12)

The map-like system observed in Layer 7 transforms into functional clustering based on navigational affordances. Corner nodes (2 available directions) form distinct clusters, edge nodes group by shared constraints, and centre nodes (4 available directions) converge into a single representation. The direction swapping experiment provides causal evidence: at constrained positions, representations remain distinct for geometrically different paths (cosine similarity ≈ 0.31), while at unconstrained positions, they converge (cosine similarity ≈ 0.98).

4.4.2 Local Heuristics vs Global Planning

Perhaps the most theoretically interesting finding is the model’s adaptive use of different computational strategies based on available information. The U-shaped accuracy curve with respect to context length reveals how the model balances local heuristics against global reasoning. This pattern suggests the model employs qualitatively different computational approaches depending on available information. With short contexts (2-3 steps), the model achieves near-perfect performance with high reverse bias, indicating reliance on local heuristics. With minimal spatial information, the model defaults to reversing the previous move: a conservative strategy that avoids invalid moves in all grid configurations. In the intermediate context regime (5-40 steps), performance drops (70-85% accuracy) while reverse bias decreases. This likely represents a transitional phase where local heuristics become less reliable but global spatial understanding remains incomplete. The model cannot yet build a complete spatial map but is exposed to enough spatial diversity to reduce reliance on simple reversal heuristics. Finally, as context length surpasses 40 steps, performance recovers to near-perfect levels ($\geq 97\%$) with minimal reverse bias. With sufficient context, the model can maintain accurate spatial state and make informed predictions based on global spatial understanding rather than local heuristics.

The loop-completion task provides convergent evidence for this local-to-global transition. The simplest 2-hop loops are solved almost immediately: performance recovers to 65% by Layer 1 and reaches 100% by Layer 2. These loops can be solved by a specialised, low-level circuit implementing the same reversal heuristic identified in the context-length analysis. Our attention pattern analysis provides a direct mechanistic explanation for this behaviour. We found that in Layer 1, a majority of attention heads consistently focus on

the penultimate node token when processing a direction. For a simple 2-hop loop (e.g., aa EAST bb WEST), this circuit may implement the reversal heuristic: by attending to the origin node (aa), it provides both sufficient context and a direct signal for the target, solving the task in the earliest layers without needing a global map. More complex loops (4-12) hops, only succeed after a sharp transition around Layer 7, marking the point at which the internal coordinate system becomes fully self-sufficient. This adaptive strategy reveals a tension in next-token prediction for spatial tasks. Local heuristics provide robust performance when information is limited but become suboptimal as more spatial information becomes available. The model must ‘unlearn’ reliable local strategies to benefit from global spatial reasoning—a process that temporarily reduces performance during the transition. This behaviour has implications beyond spatial reasoning. It suggests that transformers naturally develop hierarchical reasoning strategies that operate at different temporal scales, switching between them based on information availability. However, we observed this pattern in only one model on one task—broader generalisation claims require more evidence.

4.4.3 The Challenge of ‘Hard’ Decisions

The Foraging Model reveals a fundamental asymmetry in navigation decision-making. When predicting the next node given a direction, the model achieves perfect accuracy (100%). However, when required to predict both direction and node, performance drops to 98.3%. This gap, while small, was found to be persistent, and reflects a deeper challenge than simply predicting more tokens. This asymmetry arises from the inherent nature of the decisions. As noted by ?, NTP tends to shift focus on local patterns and overlook ‘hard’ decisions that require planning ahead. When predicting a node given a direction, there is exactly one correct answer determined by the grid structure: this is an ‘easy’ decision with a deterministic outcome. In contrast, direction prediction often presents multiple valid options, requiring the model to look ahead and plan a path through the grid. The asymmetry is also observed in the model’s generalisation to larger grids. Tasks with deterministic, geometrically constrained solutions (loop completion) maintain near-perfect performance, while open-ended tasks (random walk continuation) degrade more rapidly. This pattern also suggests that the model has genuinely learned abstract geometric principles rather than simply memorising statistical patterns from the training data.

4.4.4 Does This Qualify as a ‘Cognitive Map’?

The Foraging Model exhibits behaviours that superficially resemble elements of a cognitive map. Middle-layer representations can be linearly decoded into a Cartesian-like coordinate system, while late-layer states reflect functional clustering consistent with navigational affordances. These properties mirror spatial abstractions often attributed to hippocampal cognitive maps, but caution is warranted in making direct analogies. Classical cognitive maps, as proposed by Tolman, support allocentric, path-independent reasoning that enables flexible planning and inference; the Foraging Model exhibits some of these features (?). The emergence of a stable, orthogonal coordinate system around Layer 7 shows that position representations become largely independent of path history, while perfect performance on Hamiltonian cycle completion indicates the model maintains global spatial relationships beyond local pattern matching. The transition from coordinate tracking to functional clustering in late layers suggests that the model encodes not just ‘where it is’ but ‘what it can do from here’, which is reminiscent of how spatial information from the hippocampus is used by downstream regions like the prefrontal cortex (PFC) for planning, decision-making, and action selection (?).

However, these analogies have limits. The model’s spatial reasoning is tightly coupled to the statistics of its training environment; generalisation is strongest for tasks with deterministic geometric constraints and degrades when multiple valid continuations exist, highlighting that it has learned abstract spatial reasoning but not a fully flexible, relational map of space. In sum, the Foraging Model develops structured, functionally useful spatial representations that capture key aspects of allocentric encoding and action-oriented reasoning, but these representations fall short of the full flexibility, goal-directed inference, and environmental adaptability characteristic of biological cognitive maps.

4.4.5 Limitations

Several important limitations should be considered when interpreting these results. All findings are based on GPT-2 small architecture and 4×4 grids. The generalisability to larger models or more complex environments remains to be tested. Our experimental findings reveal a pattern of redundancy that provides important insights into the model’s internal architecture. The head redundancy analysis showed that 6 of 12 Layer 1 heads can independently achieve 60%+ cross-context transfer success, indicating that directional processing is distributed across multiple redundant circuits rather than compressed into a minimal, elegant solution. This redundancy is further evidenced by layer ablation experiments, which demonstrated that multiple layers can be removed without breaking the model’s performance, suggesting the computation is spread across redundant pathways rather than concentrated in essential components.

The distributed, redundant architecture likely reflects the model’s substantial overparameterisation (124M parameters for what is essentially a 16-node navigation problem). Rather than converging on one minimal pipeline, the model appears to have distributed spatial processing across redundant circuits, a byproduct of having far more capacity than the task requires. While this excess capacity complicates mechanistic analysis by creating multiple overlapping circuits, it also provides important insights into how transformers actually work. Finally, while we identify the endpoints of spatial processing (Layer 1 updates and 2-step attention mechanism, Layer 7 coordinate stabilisation), the exact transformations occurring in intermediate layers remain largely unexplored. The sharp performance drop at training context length suggests fundamental working memory constraints that limit the model’s spatial reasoning despite its robust cognitive map.

4.4.6 Broader Implications

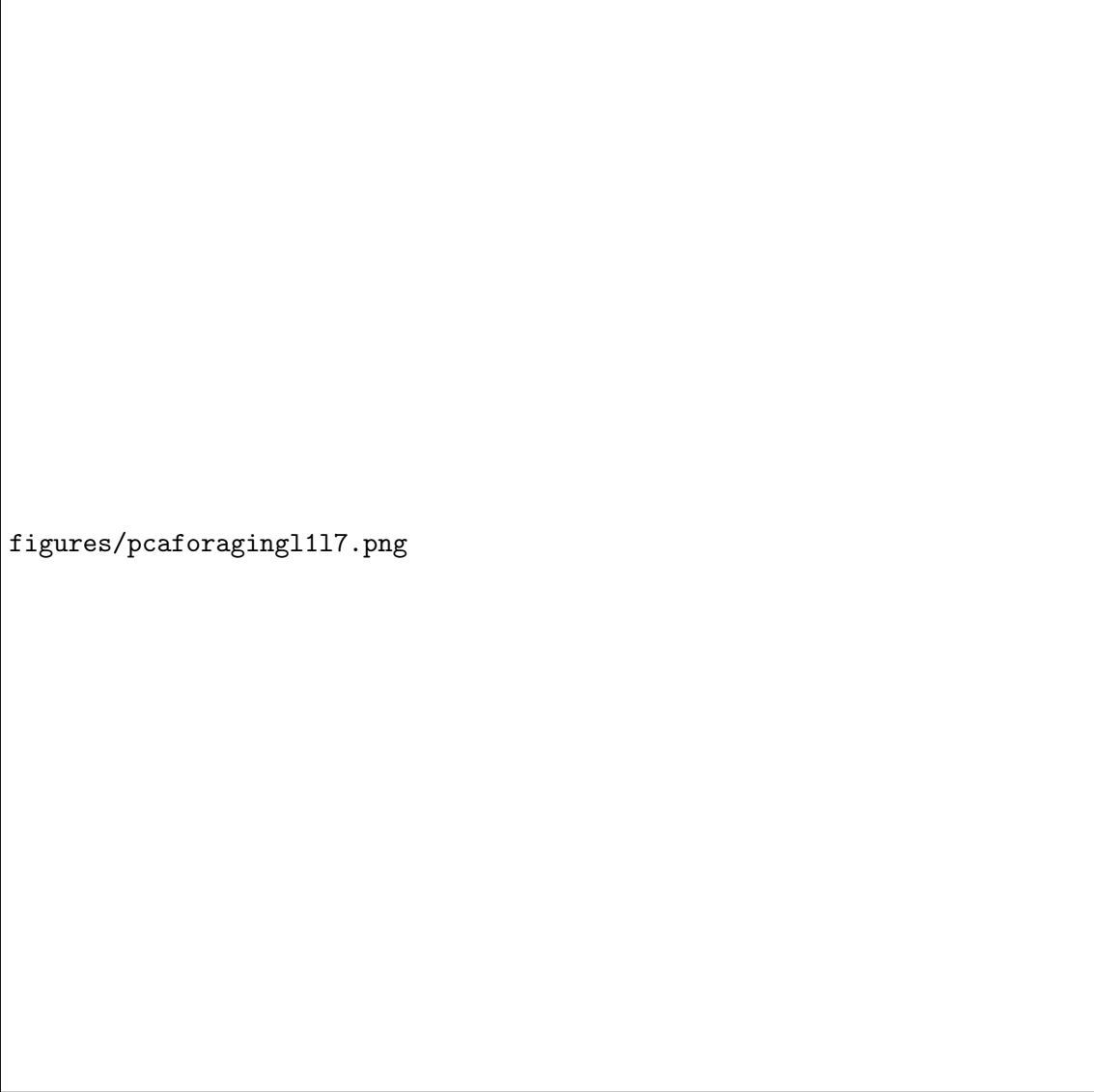
The Foraging Model demonstrates that sophisticated spatial reasoning can emerge from simple training objectives. Passive exploration can produce more robust spatial representations than might be expected from next-token prediction alone. The three-stage processing pipeline reveals general principles for how local predictions can give rise to global understanding—through progressive abstraction and information integration across network layers. We see this at the most granular level with the simple Layer 1 attention pattern—a local, two-step computation that serves as the building block for both fast heuristics and, when integrated across layers, a robust global map. Even if the resulting implementation is less clean than a neatly hierarchical algorithm, the model clearly goes beyond simple pattern matching, integrating local predictions into coherent global understanding. These findings suggest a practical design principle: rather than hardcoding spatial inductive biases, we might design training objectives that naturally encourage the natural emergence of map-like internal representations.

Our analysis of the Foraging Model establishes a clear baseline: passive exploration is enough to build a robust world model with a surprisingly interpretable structure. But

this is only half the story. The critical question now is how this learned algorithm changes when the objective shifts from aimless wandering to active, goal-directed planning. We investigate this in the next chapter by analysing the Shortest Path models.

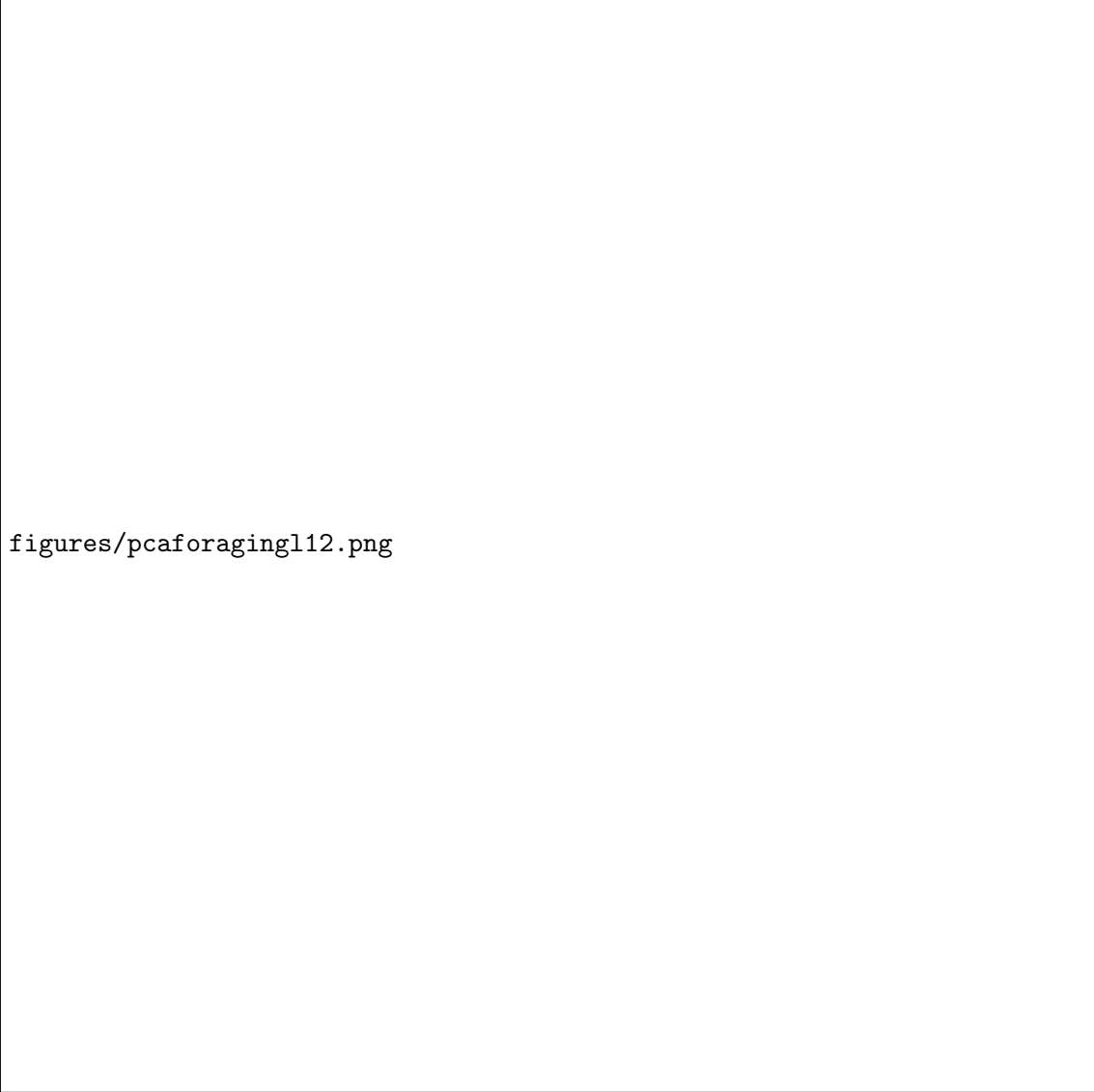
Latex template/figures/foragingperformance.png

Figure 4.1: **Foraging Model performance across context lengths and grid sizes.** (A) Single-step prediction accuracy (purple) and reverse bias (red) as a function of context length (2–115 steps), measured on 500 random walks per context length. The U-shaped accuracy curve shows transition from heuristic (high reverse bias at short contexts) to map-based reasoning (low reverse bias at long contexts). (B) Generalisation performance on larger grids: random walk continuation (1-step, CL=115) and $N \times N$ square loop completion, tested on 100 trials per grid size. Performance degrades gracefully from 98.3% (4×4) to 50% (6×6) for random walks, while loop completion maintains 100% accuracy up to 5×5 grids.



figures/pcaforaging1117.png

Figure 4.2: **PCA of node token hidden states in Layers 1 and 7.** Data from 1,000 random walks of length 120 on unique 3×3 grids. Points coloured by grid coordinates (R,C) where $R,C \in \{0,1,2\}$. Layer 1 (left) shows unstructured spatial representations, while Layer 7 (right) exhibits clear coordinate organisation with top two PCs aligning with grid axes (cosine similarity ≈ -0.0415), indicating the emergence of an orthogonal coordinate system.



figures/pcaforagingl12.png

Figure 4.3: **3D PCA of Layer 12 node token hidden states.** Data from 1,000 random walks of length 120 on unique 4×4 grids. Nodes cluster by navigational affordances: corner nodes (2 available directions, $N=4$), edge nodes (3 available directions, $N=8$), and centre nodes (4 available directions, $N=4$). Functional clustering replaces coordinate organisation, with nodes clustering by possible moves rather than spatial position, demonstrating action-oriented representation.

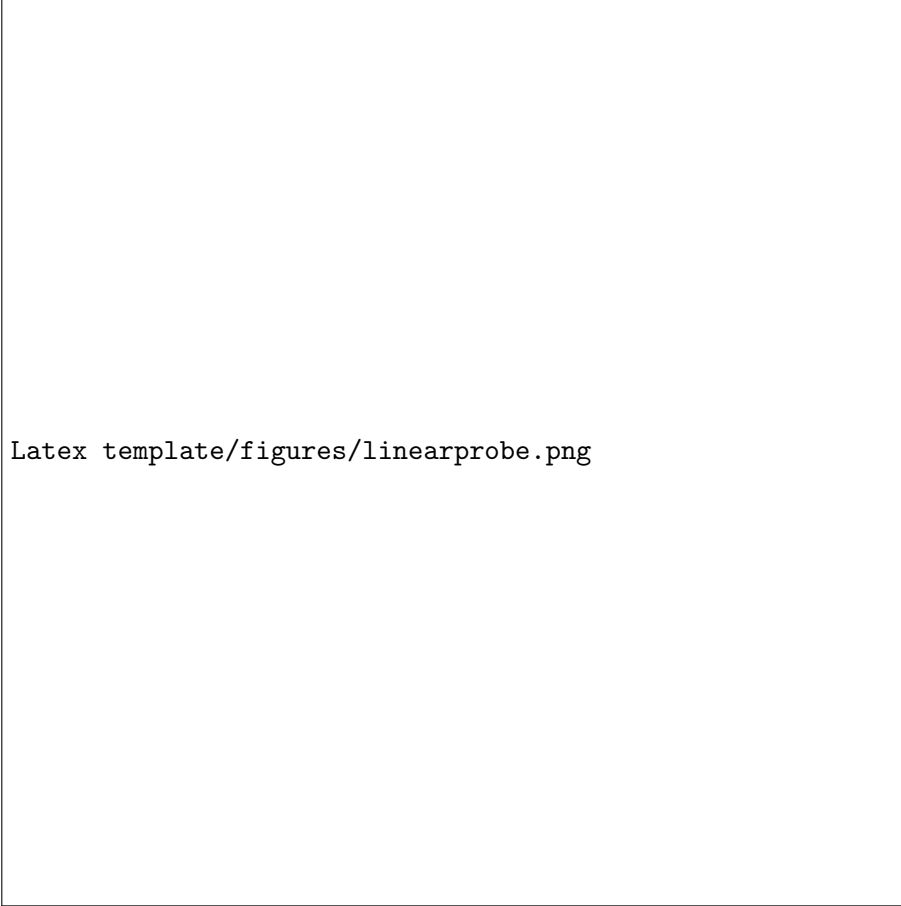
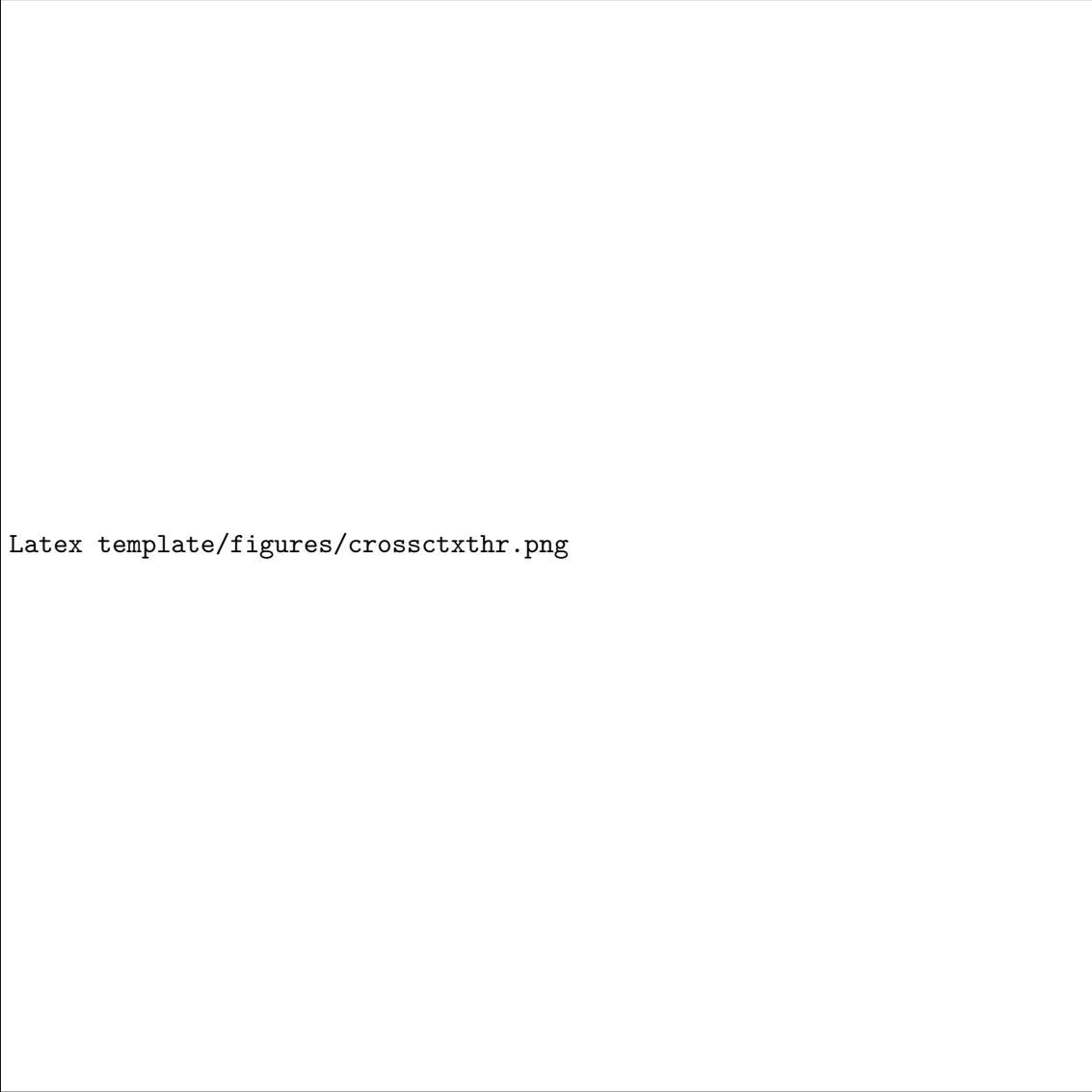
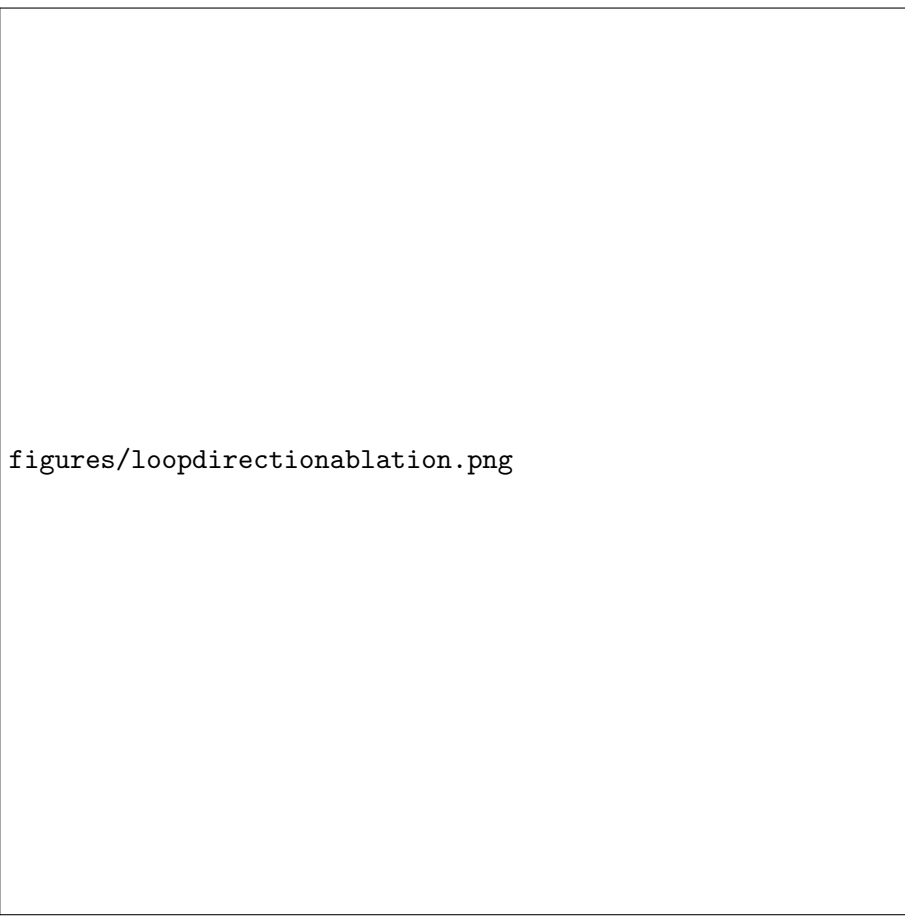


Figure 4.4: **Linear probing performance for coordinate decoding across transformer layers.** R^2 scores computed on 500 examples per layer, training linear probes to predict (x,y) grid coordinates from hidden state representations. Performance increases from $R^2=0.15$ (Layer 1) to plateau at $R^2\approx 0.93$ (Layer 8), suggesting coordinate system emergence. The plateau aligns with PCA findings, indicating stable spatial representation by middle layers.



Latex template/figures/crossctxthr.png

Figure 4.5: **Cross-context activation patching and head redundancy analysis.** (A) Outcome distribution from 1,000 cross-context patching trials, where Layer 1 attention outputs from donor contexts were transplanted into recipient contexts with different directions. 97.4% trials show universal directional transfer, 2.2% reproduce donor context nodes, 0.4% produce unexpected outputs. (B) Individual head performance in cross-context transfer when other heads are zeroed out. 6 of 12 Layer 1 heads achieve 60%+ success independently (N=100 trials per head), indicating distributed rather than specialised directional processing.




figures/loopdirectionablation.png

Figure 4.6: **Stratified direction token ablation results reveal multiple computational strategies.** Historical direction tokens were zeroed out at each layer’s input on 1,000 loop completion trials, stratified by loop length (2 to 12 hops). The results show two distinct patterns: **(1)** Trivial 2-hop loops are solved by Layer 2, suggesting a fast, local heuristic circuit in early layers. **(2)** Complex 4-12 hop loops show a consistent phase transition between Layers 6-8, with performance recovering significantly at Layer 7 and reaching near-perfection for all lengths by Layer 8. This indicates that the global coordinate system becomes fully self-sufficient by Layer 8, independent of task complexity.

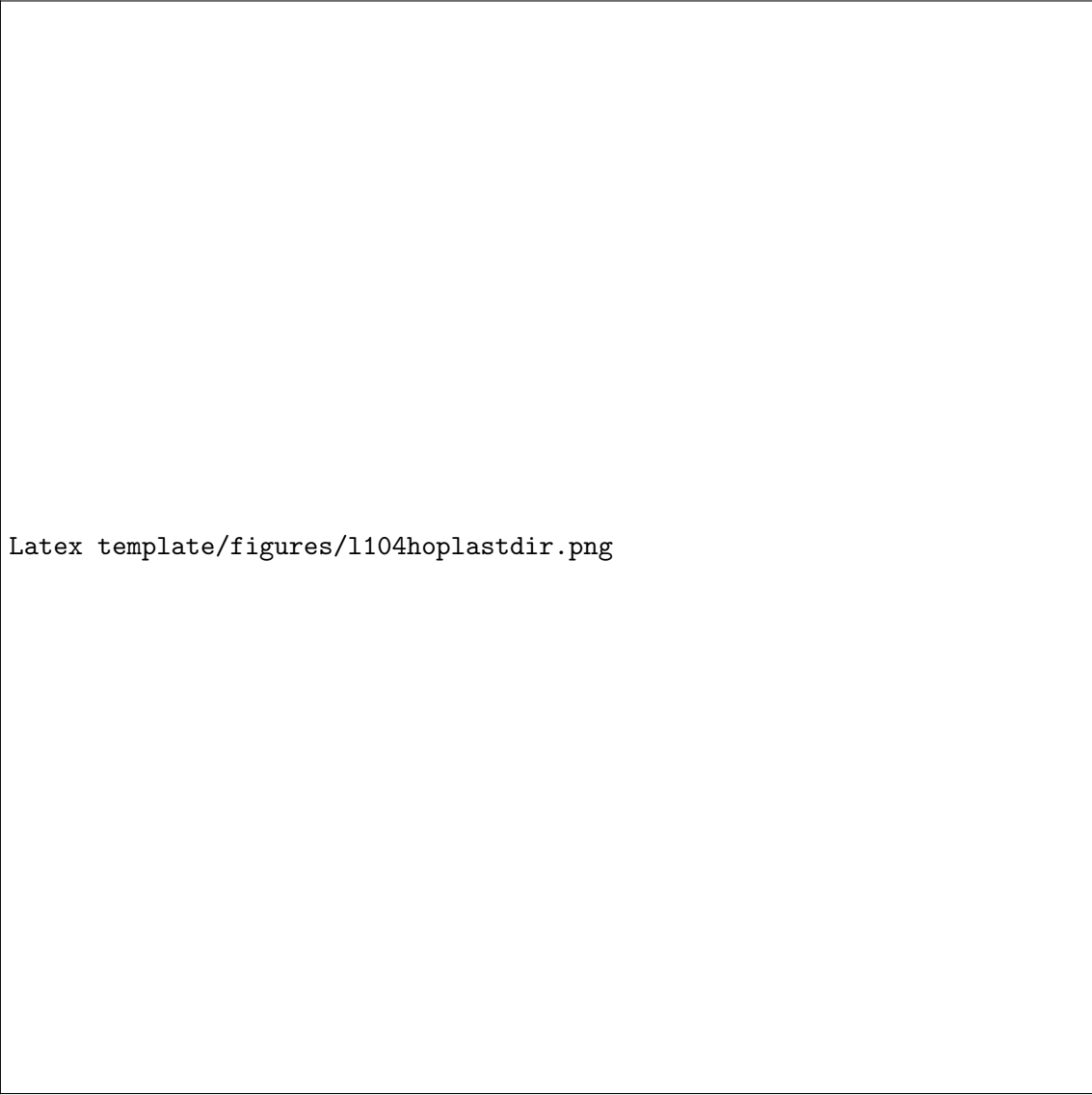


Figure 4.7: **Direction swapping experiment results by node constraint type.** Cosine similarity between hidden states from original and geometrically mirrored paths (all directions reversed) on 500 trials per node type. Unconstrained centre nodes (4 available directions) show high similarity ($\cos \approx 0.98$), indicating path history discarded when irrelevant. Constrained corner nodes (2 available directions) show low similarity ($\cos \approx 0.31$), demonstrating selective preservation of path history when geometrically necessary.



Latex template/figures/4hoplastdir.png

Figure 4.8: **Layer 1 attention patterns for the final direction token in a 4-hop loop.** A majority of heads (2, 5, 7, 9-12) focus their attention almost exclusively on the penultimate node token. Head 4 attends exclusively to node tokens, while Head 8 attends to direction tokens. Averaged over 1,000 trials with randomised node names; error bars = ± 1 SD.



Latex template/figures/l104hoplastdir.png

Figure 4.9: **Layer 10 attention patterns for the final direction token in a 4-hop loop.** Most heads strongly attend to the final direction token itself. Averaged over 1,000 trials with randomised node names; error bars = ± 1 SD.

Chapter 5

Shortest Path Models

The previous chapter demonstrated that a transformer can develop a robust, three-stage spatial reasoning algorithm from passive, exploratory learning alone. The Foraging Model successfully built a map-like representation of its environment simply by predicting the next step in a random walk. This chapter now asks a critical question: what happens to this learned algorithm when the model is subjected to an active, goal-directed planning objective?

We now shift from passive exploration to active exploitation. The Shortest Path (SP) models are not trained on the purely local objective of predicting a valid next step, but on the global planning task of finding the most efficient route between two points. This imposes a strong optimisation pressure. Does this pressure force the model to develop a more sophisticated planning algorithm that leverages the underlying coordinate map? Or, does it encourage the model to abandon general map-building in favour of brittle, task-specific heuristics, effectively learning shortcuts instead of the map itself? To dissect this question, we analyse two variants of SP models. The SP-Hamiltonian (SP-H) model tests whether the agent can build and use a complete world model when given a structured, complete view of the environment: a direct test of its optimal planning capabilities. In contrast, the SP-Random Walk (SP-RW) model creates a more direct comparison to our baseline, forcing the agent to plan using the same kind of sparse, partial information the Foraging Model was trained on.

By applying the same behavioural, representational, and mechanistic toolkit used in the previous chapter, we can perform a direct, comparative analysis of the learned algorithms. This chapter investigates whether the introduction of a planning objective fundamentally alters the emergence of spatial intelligence, providing crucial insights into how different training paradigms shape the computational strategies learned by transformers.

5.1 Behavioural Analysis

We evaluate the SP models’ spatial reasoning capabilities using the evaluation framework described in Chapter 3, assessing core performance metrics, context length robustness, and generalisation capabilities. The analysis reveals distinct behavioural patterns that contrast sharply with the Foraging Model’s robust generalisation.

5.1.1 Core Performance and Context Robustness

The SP-H model achieves perfect accuracy on its training distribution: 4×4 grids with Hamiltonian context walks of exactly 16 steps, regardless of path complexity (Manhattan distance between start/goal nodes, as seen in Fig. ??A). This performance extends to a

hold-out test set of unseen Hamiltonian ‘shapes’, suggesting that the model has learned to solve the specific task rather than memorising individual examples. However, this success comes at a cost: the model fails catastrophically when context length deviates from 16 steps, even by a single step. To test whether this brittleness stems from the content of the context (unstructured vs. structured) or merely its length, we evaluated the SP-H model on 16-step random walks, providing it with a context that is the same length as its training data but lacks the ordered, non-repeating structure of a Hamiltonian path. Critically, we guaranteed task solvability: each random walk was generated to contain all necessary nodes for at least one valid shortest path between randomly selected start/goal nodes. On this task, its performance was limited, with 36.5% accuracy on average, dropping with Manhattan distance (MD) (Fig. ??A).

The SP-RW model, finetuned from SP-H on variable-length random walk contexts (10–50 steps), demonstrates more flexible performance. It maintains high accuracy across the full range of its training context lengths (99.5% at 10 steps to 97% at 50 steps). When tested just beyond its finetuning window at a context length of 55 steps, performance drops to 13.5%. This is a classic example of a transformer’s failure to extrapolate beyond its training context window, a limitation also observed in the Foraging Model at its 120-step limit. Nevertheless, the SP-RW model achieves 99% accuracy on the original Hamiltonian context task, indicating that it retains prior knowledge while improving flexibility during the fine-tuning process. It also exhibits somewhat consistent performance for both random walk and hamiltonian contexts on a 4x4 grid, regardless of path complexity (Fig. ??A).

5.1.2 Generalisation Performance

5.1.2.1 High Manhattan Distance Tasks

A critical challenge emerges when evaluating SP-H on larger grids due to its sensitivity to context structure. Unlike SP-RW, which was trained on variable-length random walks, SP-H was trained exclusively on 16-step Hamiltonian paths where each node appears exactly once. On a 5×5 grid containing 25 nodes, generating a traditional 16-step Hamiltonian path becomes impossible, creating an unfair evaluation scenario. To enable fair comparison, we adapt the evaluation protocol for SP-H by generating simple random walks (non-repeating paths) of length 16 on 5×5 grids. These context walks are constrained to be solvable as mentioned previously. This ensures that the model has sufficient spatial information to solve the task, while maintaining the non-repeating structure it was trained on.

The results reveal notably different generalisation capabilities between the models (Fig. ??B). SP-RW exhibits a stable performance degradation, starting at 90.4% accuracy on MD 1 tasks and showing a gradual, nearly linear decline to 21.4% at MD 8. This reasonable performance beyond MD 6 (the maximum possible distance on its 4×4 training grid), shows meaningful transfer of spatial reasoning beyond training constraints. In contrast, SP-H exhibits a much steeper exponential decay pattern. Starting from 82% accuracy on MD 1 tasks, performance drops sharply to 0% by MD 7-8. Hence, it seems that the SP-H model’s reasoning ceiling is MD 6.

5.1.2.2 Edge-to-Edge Tasks

Having observed that SP-H cannot generalise beyond its training reasoning ceiling of MD 6, a new question arose: is its poor generalisation due to the path length exceeding its training data ($MD \leq 6$), or is it a more fundamental inability to apply its reasoning to a novel grid layout? To disentangle these factors, we designed the edge-to-edge navigation task (see Chapter 3 for details). Although edge-to-edge paths cannot be directly mapped to

any configuration within a 4×4 training grid (e.g., four steps east), they require planning only 4 steps ahead: well within the MD 6 boundary that defines the model’s training constraints. This makes edge-to-edge the most basic test of whether a model can apply its learned spatial reasoning to novel spatial configurations that remain within its established planning horizon. SP-RW achieves 78% accuracy, while SP-H achieves only 3.6%. The high performance of SP-RW on edge-to-edge tasks suggests that it has learned robust directional reasoning that can be extended to novel contexts. In contrast, SP-H’s near-complete failure is particularly revealing: despite having the computational capacity to plan 4 steps ahead (Fig. ??A), it cannot generalise this capability to the novel spatial context of an edge-to-edge path on a larger grid. Thus, it is clear that fine-tuning on suboptimal walks not only increased the model’s generalisation to new context lengths, which is perhaps trivial and expected, but also its ability to adapt to larger grids and extend its planning horizon.

5.2 Representational Analysis

Next, we examine the internal representations of the SP models using the same analytical framework as the Foraging Model, with certain methodological differences. Firstly the choice to average across node occurrences is context dependent. Unlike the Foraging Model’s analysis, which averaged node representations across many random walks, here we must be more nuanced. Hamiltonian contexts contain each node only once, so averaging is not possible. For SP-RW, we will analyse both averaged and non-averaged representations to draw the clearest comparisons. Second, PCA can be conducted on either context or task nodes. Our goal is to understand how the pressure of a goal-directed planning objective alters the map-building process we observed in the Foraging Model, so we focus on context nodes here.

Thus, we stratify our analysis by context type (random walk contexts for SP-RW, Hamiltonian contexts for both models), and context length for random walks (10–50 steps).

5.2.1 Foraging Model vs. SP-Random Walk

To create a direct, apples-to-apples comparison with the Foraging Model, we first analyse the SP-RW model’s representations on 3×3 grids using the same methodology: averaging node representations across 500 random walks. The results reveal a shift in the nature of the learned spatial representations, despite shared architectural constraints and the processing of identical walks (Figure ??). The Foraging Model, as seen in Chapter 4, develops a clean world model. Its representations evolve from a noisy sense of coordinates in Layer 1 to a near-perfect Cartesian grid in Layer 7, suggesting the formation of a stable, path-independent coordinate system. The later layers refine this map to emphasise navigational affordances. Interestingly, the SP-RW model’s first-layer representations are nearly identical to the Foraging Model’s, despite its pre-training on completely different, structured Hamiltonian paths. This shared starting point quickly diverges. By Layer 7, the same pattern is still present but slightly sheared for SP-RW, lacking the clean structure of the Foraging Model. By Layer 12, its representations have collapsed into tight, compressed columns. The two models begin with the same raw spatial sense, but develop fundamentally different representational strategies based on their training objectives.

5.2.2 SP-Random Walk

When analysing the SP-RW model’s hidden states without averaging across node occurrences, we retain the contextual information associated with each node; specifically its

arrival direction and its position within the path. This leads to a very different representational geometry compared to averaged analyses. Instead of forming a single stable map of coordinates, the embeddings are strongly trajectory-dependent. We observed that nodes cluster primarily according to their arrival direction (Figure ??, left). For instance, the same coordinate, such as (0,0), does not map to a single point but appears in both the ‘SOUTH’ and ‘EAST’ clusters, depending on the preceding move. This phenomenon is intuitive: the direction from which a node is entered constrains the set of available next moves and thus provides a useful proxy for local navigational constraints. Start nodes, which lack an arrival direction, form their own distinct cluster. Importantly, this trajectory-dependent clustering is not unique to SP-RW; the same pattern is observed when the Foraging model is examined without averaging (Figure ??, right).

Across both RW and Hamiltonian contexts, variation with respect to path index is consistently present in the embeddings. However, this should not be over-interpreted as evidence that the models explicitly encode sequence position for spatial reasoning. Two alternative explanations are plausible. First, the effect likely reflects the influence of positional embeddings, which are a standard component of transformer architectures and therefore expected in any model. Second, the pattern may act as an implicit uncertainty signal: nodes appearing earlier in the sequence, when less contextual information is available, tend to cluster together, while later nodes spread out as more structural constraints accumulate. Crucially, this path index variance is also not unique to the SP models. The Foraging model exhibits the same behaviour, but the effect is masked when averaging over node occurrences. An interesting representational transition occurs at the 29th node of a given path: nodes with path indices below 29 maintain the clean arrival direction clustering pattern, while nodes with path indices above 29 collapse into four entangled clusters with no obvious organisational principle. The mechanism behind this transition remains unclear, a plausible explanation is the training data distribution: random walks were sampled with lengths between 10 and 50, with a midpoint around 30, potentially inducing a change in representational strategy near that range. However, other explanations are possible, and this represents an area for future investigation. Importantly, this representational change does not impair task performance—specialized behavioural tests show that the model maintains high accuracy (93-97%) on tasks requiring information from the ‘late’ portion of context walks, demonstrating that the transition represents a change in representational strategy rather than a functional limitation.

When tested on Hamiltonian contexts, the SP-RW model shows similar patterns to random walk contexts, with arrival direction being the primary organisational principle. We observe sub-clustering based on path index as expected.

5.2.3 SP-Hamiltonian

The PCA analysis reveals a suprisingly different representational pattern in SP-H: the model shows strong horizontal mirroring in its context node representations. Coordinates (0,0) and (3,0) appear nearly identical in the PCA space, as do (0,3) and (3,3), with similar patterns for all coordinate pairs across the central horizontal axis. This pattern is observed for all coordinate pairs: (0,1) with (3,1), (0,2) with (3,2), (1,0) with (2,0), (1,1) with (2,1), (1,2) with (2,2), and (1,3) with (2,3).

This perfect horizontal mirroring across the central horizontal axis of the 4×4 grid could explain why north/south directions are seen to overlap, whereas east/west show more distinct clusters. However, we note that PCA is a linear dimensionality reduction technique, and the models’ representations may lie on non-linear manifolds not fully captured by this analysis. The observed patterns represent projections of potentially more complex high-dimensional structures onto two-dimensional spaces (three principal compo-

nents here), which would explain the apparent symmetrical collapse. Interestingly, this pattern is completely different from the Hamiltonian context node representations of SP-RW, despite it being fine-tuned from SP-H. In contrast, SP-RW exhibits nearly identical PCA patterns for both Hamiltonian and random walk contexts. This consistency makes sense: a Hamiltonian path can be viewed as a subset of a random walk, but not vice versa. During fine-tuning from SP-H, SP-RW appears to have adapted its representations to support both context types simultaneously, rather than switching strategies depending on the context.

5.2.4 Layer-Wise Dynamics

For both SP-RW and SP-H, cluster structure is remarkably stable across layers. The cluster shapes evolve gradually, suggesting iterative refinement of representations. This contrasts with the Foraging Model, which exhibited a sharp representational shift at Layer 7—a qualitative reorganisation of clusters. This indicates that SP models may rely more on incremental spatial reasoning.

5.3 Mechanistic Analysis

The behavioural and representational analyses reveal clear differences between the SP models, but the underlying computational mechanisms remain partially understood. The SP models’ mechanistic analysis is more limited than what we might achieve with more extensive experiments. However, the available evidence provides insights into why finetuning on random walks led to increased robustness.

5.3.1 Direction Token Ablation: Continuous Dependence

Layer-wise direction ablation experiments reveal a key algorithmic distinction between the exploratory Foraging Model and the goal-directed SP models. In the Foraging Model, ablating direction tokens revealed a sudden transition at Layer 7, where its internal spatial map became self-sufficient. Based on our SP models’ PCA analyses, which showed stable internal representations across layers, we hypothesised that the SP models would process directional information through a fundamentally different mechanism. The ablation results confirm this hypothesis. Both SP-RW and SP-H models exhibit continuous dependence on directional information throughout all network layers, with a gradual, monotonic increase in accuracy as the ablation layer increases (Figure ??). There is no transition point where performance suddenly recovers. Instead, accuracy is near 0% when ablating early layers and smoothly approaches baseline performance only when layers 11 and 12 are spared. Interestingly, for SP-H, it never quite recovers its perfect accuracy. This provides strong causal evidence that, unlike the Foraging Model, the SP models never consolidate spatial information into a self-sufficient map. Their algorithm is one of continuous, path-dependent computation, where explicit directional tokens remain critical throughout all 12 layers. This gradual processing is consistent with the PCA results, which showed iterative representational refinement across layers rather than the drastic representational rearrangement seen in the Foraging Model.

A surprising result, however, was the near-identical ablation curves for SP-H and SP-RW. Despite SP-RW’s superior robustness and different representational geometry, its core reliance on direction tokens is mechanistically identical to SP-H’s. This finding highlights an important distinction between representational similarity and algorithmic similarity. Although SP-RW develops representations that are more similar to the robust Foraging model (as evidenced by PCA), it nonetheless seems to employ the same continuous,

path-dependent computational strategy as SP-H. This interpretation aligns with established findings in the literature: rather than fundamentally altering learned algorithms, fine-tuning typically teaches existing computational mechanisms to tolerate a wider variety of inputs while preserving core processing strategies. The robustness improvement appears to stem from better representational organisation rather than a fundamental algorithmic change—the existing algorithm simply learned to work with more Foraging-like representations while maintaining the same mechanistic dependencies on directional information.

5.3.1.1 Representational vs. Functional Symmetry

The SP-H model’s PCA analysis revealed a pattern in context node representations, where coordinates symmetric about the central horizontal axis overlap in the embedding space. This pattern might suggest asymmetric directional processing—perhaps the model distinguishes east-west movements while conflating north-south directions. However, targeted ablation experiments contradict this interpretation. Removing either NORTH/SOUTH tokens or EAST/WEST tokens produced identical performance drops, demonstrating that the model treats all four cardinal directions equivalently during computation. This indicates that the SP-H model’s functional use of directional information is globally symmetric, even if PCA shows subtle separations along one axis. In other words, while PCA captures representational geometry, it does not fully reveal the model’s computational reliance on directional cues. Taken together, these results demonstrate that SP-H embeddings encode positional information in a highly symmetric fashion, which is likely due to the symmetric structured nature of Hamiltonian paths. This likely contributes to the model’s steep decay in generalisation and explains why fine-tuning on unstructured walks mitigated this.

5.4 Discussion

The previous chapter demonstrated that transformers can develop a robust, map-like representation of space through passive, exploratory learning. In this chapter, we investigated what happens when the training objective shifts from passive exploration to active, goal-directed planning. Our primary research question was whether transformers learn universal ‘cognitive maps’ or task-specific heuristics. The answer is not binary; rather, models lie along a spectrum determined by the interaction between their training objective and the statistical structure of their data. This analysis suggests a trade-off: models can develop robust, generalisable spatial reasoning, but doing so may come at the expense of highly optimised, task-specific performance.

5.4.1 Mechanistic Comparison

This high-level strategic difference is causally substantiated by our mechanistic analyses, which reveal two distinct computational algorithms. The Foraging Model’s direction ablation experiments demonstrated a clear phase transition at Layer 7, the point at which its internal coordinate map becomes causally self-sufficient and no longer relies on explicit historical direction tokens. This suggests a process of information consolidation, where local movement cues are integrated into a stable, abstract representation. The SP models show no such transition. For both SP-H and SP-RW, performance recovers gradually and monotonically as direction ablation is moved to later layers, providing strong evidence that their algorithm remains continuously dependent on explicit direction tokens throughout all 12 layers. There is no sharp transition point where the model becomes self-sufficient.

Instead, their reasoning appears to be an ongoing, step-by-step calculation tethered to the sequence of moves provided in the context.

This gradual recovery has important nuances. The SP-H model, for instance, never fully recovers its perfect baseline accuracy, even when directional information is only ablated in the final layer. This suggests its highly specialised algorithm is extremely sensitive, requiring an uninterrupted flow of directional cues through the entire network to execute its procedure flawlessly. The case of the SP-RW model is more ambiguous. On one hand, it recovers to its baseline accuracy by layer 10. Given its training on random walks and its representational similarities to the Foraging Model, this *might* suggest that it has an internal map which becomes sufficiently stable by this point. However, we cannot be certain. The recovery is gradual, lacking the sharp phase transition that characterises the Foraging Model’s shift to map-based computation. Furthermore, because SP-RW’s baseline accuracy is lower ($\approx 92\%$), small deviations in performance are less diagnostic than for the Foraging Model or SP-H, where any drop from 100% reflects a clear mechanistic disruption.

5.4.2 Representational Comparison

This mechanistic difference provides a clear lens through which to interpret the representational findings. The Foraging Model’s three-stage evolution—from noisy coordinates to a perfect Cartesian grid to functional affordance clusters—indicates a representational hierarchy that mirrors the consolidation of a stable, allocentric map. This structure allows the model to support flexible generalisation, as its later layers encode not just where things are, but what can be done from those positions. Crucially, this is consistent with its mechanistic transition at Layer 7, where local directional inputs cease to be necessary, and the learned map becomes causally self-sufficient. In contrast, the SP models show no such phase transition, and their representational footprints reflect this. The perfect horizontal mirroring in the SP-H model’s representations is a particularly telling example. This symmetry is likely not an error but a clever, albeit brittle, feature learned from the highly structured and symmetric nature of Hamiltonian path data. The model discovered a compression strategy that exploits the regularities in its training distribution, but this shortcut breaks down on the unstructured, asymmetric data of random walks or larger grids.

The SP-RW model occupies a middle ground. Its representations begin similarly to the Foraging Model’s, suggesting that fine-tuning on random walks reintroduces pressures toward allocentric map-like organisation. However, instead of converging to a clean grid, they shear, and ultimately collapse into compressed columns, likely reflecting some useful structure that is uninterpretable with linear dimensionality reduction. After position index 29, nodes are further organised into four coarse clusters with no apparent organising principle. Behaviourally, this collapse does not translate into failure, since the model remains accurate on tasks that depend on late-context information. What it does indicate is that the representational strategy of SP-RW is less cleanly defined than that of the Foraging Model. Whether this reflects the limited strength of fine-tuning (competing pressures of Hamiltonian pre-training and random walk fine-tuning may indicate that the model has not converged to a stable spatial encoding), or a consequence of the goal-directed training objective leading to less interpretable representational structure, remains open.

5.4.3 The Effect of Fine-tuning

The relationship between the SP-H and SP-RW models offers a nuanced insight into the mechanics of fine-tuning. Fine-tuning on random walks dramatically improved SP-RW’s

generalisation, allowing it to handle varied context lengths, larger grids, and longer planning horizons. One might assume this improvement came from learning a new, more robust algorithm. However, our causal analysis suggests otherwise. The direction ablation experiments reveal that both models share a similar continuous dependence on direction tokens, pointing to a computational mechanism of path-dependent reasoning, rather than a shift towards the map-based strategy seen in the Foraging Model. Representational analysis offers further insight. Node embeddings show that SP-RW and the Foraging Model share similar patterns: collective embeddings cluster by arrival direction across all layers, and averaged node representations are nearly identical at Layer 7, though the Foraging Model undergoes a subsequent shift. The source of SP-RW’s improved robustness, therefore, appears to be representational rather than algorithmic. Note that a sufficiently large representational change could, in principle, be considered an ‘algorithmic change’ in its own right. Here, we define an ‘algorithmic change’ as a fundamental shift in the causal flow of information through the network, while ‘representational change’ refers to modifications in how information is encoded without altering the underlying computational mechanism.

Rather than inducing a new algorithm, the fine-tuning process appears to have taught its existing path-dependent algorithm to tolerate a wider variety of inputs. By being exposed to the messy, unstructured nature of random walks, the model was forced to develop representations (like the arrival-direction clustering) that were less brittle than the symmetric representations learned from Hamiltonian paths. This finding aligns with the literature, which notes that fine-tuning often adapts existing circuits to new data distributions rather than inducing the formation of entirely new algorithms [?]. This is also consistent with our observation of the loss curves during training for both models, where SP-H exhibited several ‘phase changes’, but SP-RW did not. In the literature, phase transitions are abrupt, non-linear changes in a deep learning model’s training loss over time, which usually signal the emergence of new behaviours or strategies within the model [?]. We examine these phase transitions in more detail in Appendix ??.

5.4.4 Allocentric Maps vs Egocentric Path Integration

Comparing the Foraging and SP models allows us to answer our primary research question directly. The core contrast is not simply that the Foraging Model learns a map and the SP models do not. Rather, the models learn different spatial reasoning strategies that lie on a spectrum from purely egocentric path-dependence to a more allocentric, map-based understanding. At one extreme lies the SP-H model. Trained exclusively on structured, optimal Hamiltonian paths, it develops a highly efficient but brittle algorithm. Its strategy seems to be akin to procedural path integration—continuously tracking movement from a start to a goal without forming a reusable, abstract model of the environment. This approach is highly effective for the narrow task it was trained on, enabling it to achieve perfect accuracy. However, this specialisation renders it fragile; its performance collapses when conditions deviate even slightly, as seen in its failure on non-standard context lengths or unstructured paths. At the other extreme, the Foraging Model, trained on unstructured random walks, is compelled to develop a different solution. To reliably predict the next step from any point in a meandering path, it must consolidate its experience into a coherent, path-independent world model. This results in a self-sufficient, allocentric representation of space—similar to a cognitive map. The SP-RW model sits somewhere in between. Its exposure to random walks during fine-tuning pushes its internal representations towards the allocentric, map-like end of the spectrum, and improved its generalisation capabilities. This is similar to the concept of *latent learning* observed in mammals, where exploration without immediate reward supports the formation of a cognitive map that later facilitates goal-directed behaviour (??), albeit in reverse in this case.

5.4.5 Limitations

Several important questions remain unanswered. Our mechanistic analysis of the SP models, while revealing the core algorithmic difference in directional processing, is less complete than our analysis of the Foraging Model. We have established that the models rely on a continuous, path-dependent heuristic, but have not reverse-engineered the specific computations that perform path planning. Future work could use more granular patching experiments—for instance, patching the representations of start and goal tokens—to trace how the model selects and generates the optimal path. Furthermore, while our evidence suggests that SP-RW’s robustness comes from a representational shift, this is still unclear. The precise nature of that shift and whether it enables the old algorithm to work on new data remains an open question for further investigation. We cannot definitively rule out alternative explanations for the robustness improvement. Finally, as noted in the previous chapter, our analysis is also limited to a specific architecture and environment (4×4 grids). It remains unclear how these patterns would extend to larger models or more complex environments. The bounded generalisation observed in the SP models might be overcome with different architectures or training procedures.

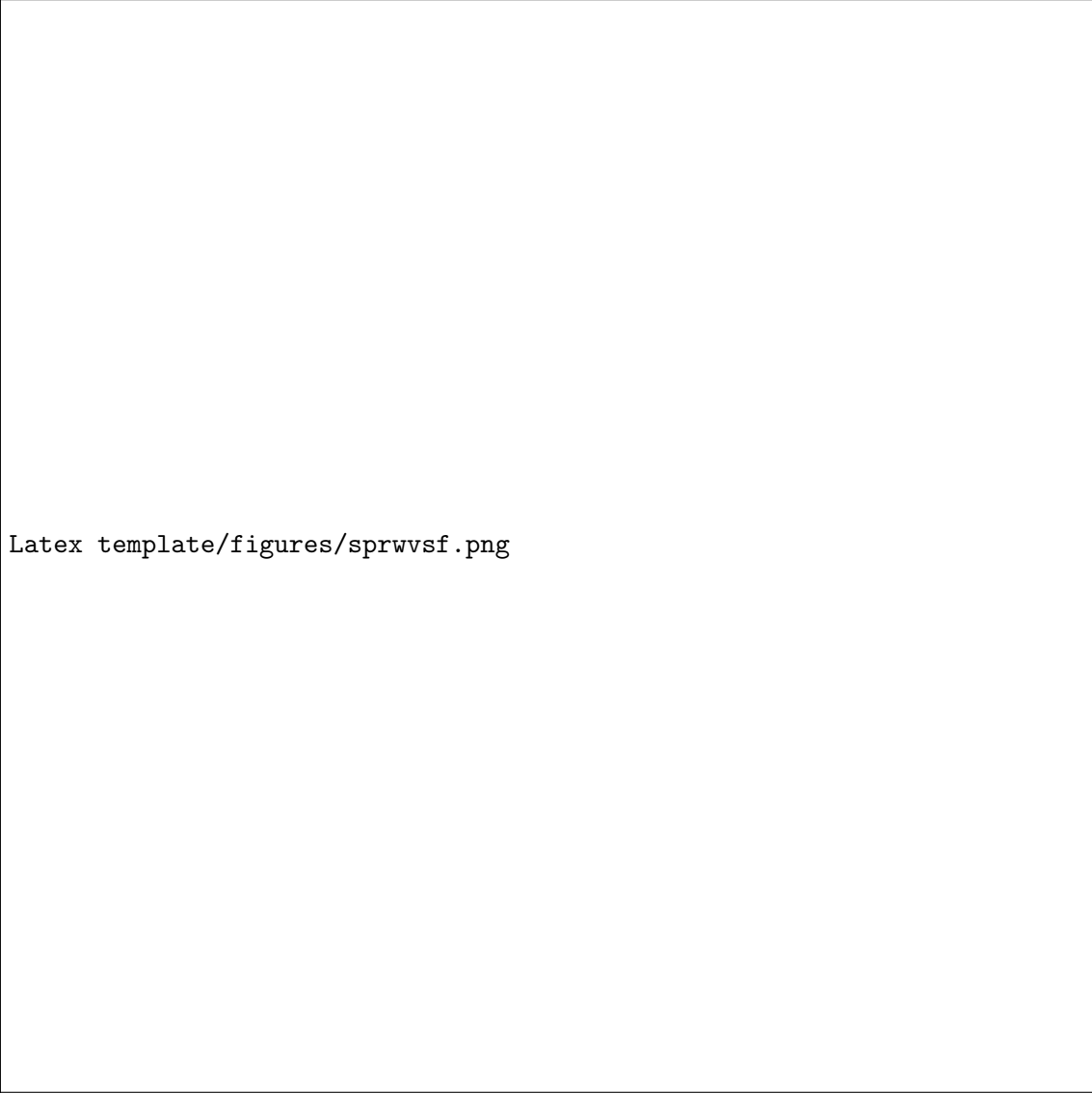
5.4.6 Broader Implications

Taken together, our comparative analysis offers insights into how training objectives and data characteristics shape the emergence of intelligence in transformers. The passive, exploratory objective of the Foraging Model created an inductive bias toward learning general, reusable structures—in this case, a cognitive map. Because it couldn’t know what information would be relevant for predicting the next step in an arbitrary walk, its best strategy was to model the entire environment. Conversely, the active, goal-directed objective of the SP models created an inductive bias toward finding the most computationally efficient solution to a specific problem, leading to specialised, but more brittle heuristics.

This has implications for training AI systems for complex planning and reasoning. If a model is trained exclusively on optimal, expert demonstrations (akin to our Hamiltonian and shortest-path data), it may learn powerful but narrow strategies that fail to generalise to novel situations. In contrast, training paradigms that incorporate exploration, sub-optimal data, and a degree of randomness (like the Foraging Model’s random walks) may be key for fostering the development of robust, generalisable world models. The apparent ‘inefficiency’ of passive exploration may be a necessary catalyst for the kind of compositional learning that underpins true spatial intelligence. This work suggests that to build truly generalist agents, we may need to balance the pressures of optimisation with the creative uncertainty of exploration, ensuring our models learn the map, not just the shortcuts.

Latex template/figures/spmodelperformances.png

Figure 5.1: **Performance across Manhattan Distances on 4×4 and 5×5 grids.** (A) Manhattan Distance between Start/Goal on 4×4 grids showing SP-H (purple) and SP-RW (green) performance across different context types. (B) Manhattan Distance between Start/End on 5×5 grids showing SP-Hamiltonian (blue) and SP-RW (green) generalisation performance. SP-RW shows gradual decline from 95% at MD 1 to 22% at MD 8, while SP-Hamiltonian maintains reasonable performance only for very short paths (83% at MD 1) before rapidly degrading to 0% by MD 7.

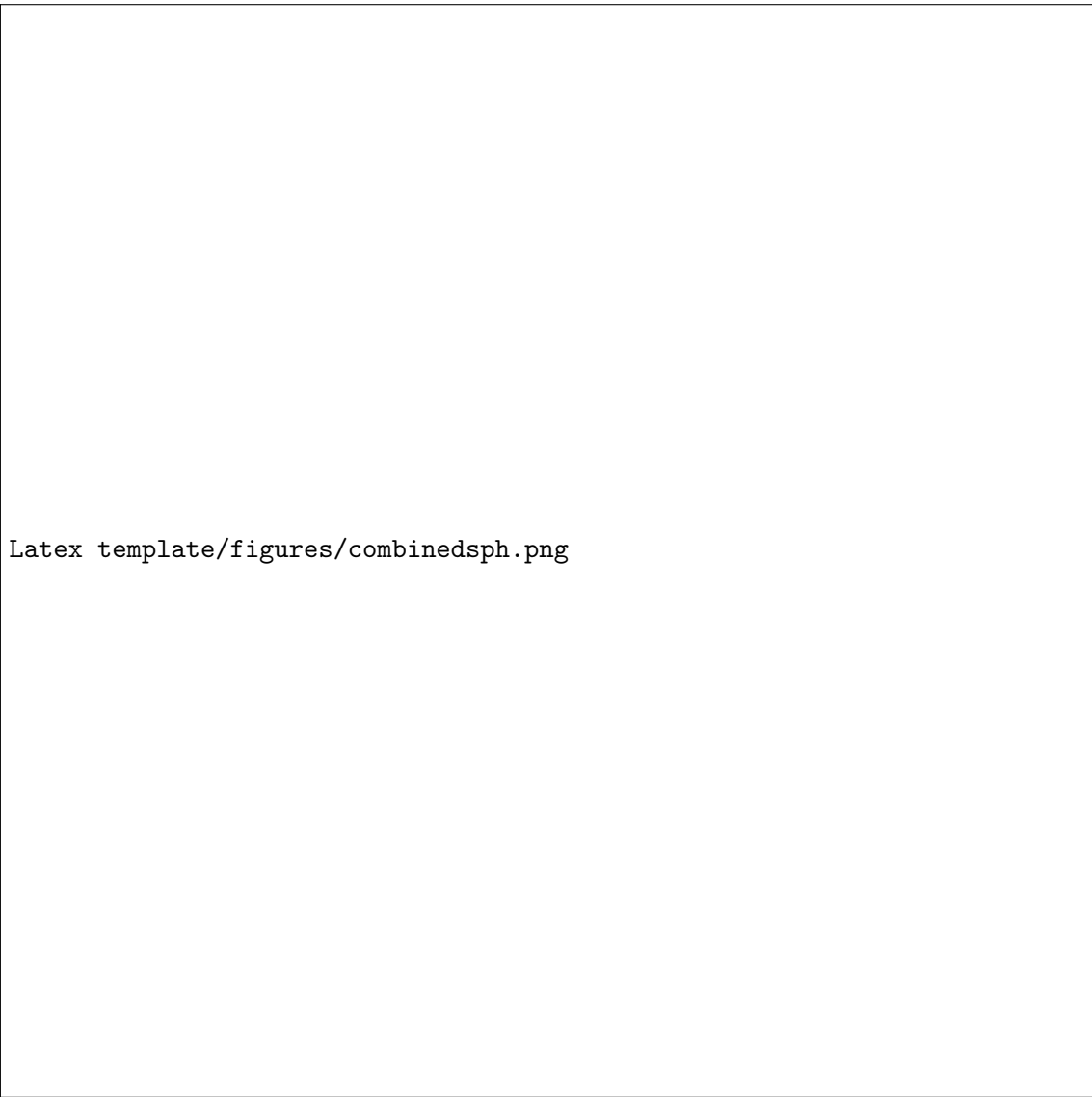


Latex template/figures/sprwvsf.png

Figure 5.2: **PCA comparison between SP-RW and Foraging models.** Top row: Foraging Model showing three-stage evolution from noisy coordinates (Layer 1) to clean coordinate system (Layer 7) to functional clustering by navigational affordances (Layer 12). Bottom row: SP-RW model showing gradual refinement (Layers 1-7) followed by representational collapse into compressed vertical columns (Layer 12). Both models start similarly but develop fundamentally different representational strategies based on their training objectives.

Latex template/figures/29thnode.png

Figure 5.3: **The 29th Node Effect in SP-RW compared to Foraging Model.** Left panel shows SP-RW model PCA representations at different context lengths (CL 28, 29, and 40) for a 3×3 grid, with all points colored by arrival direction. At CL 28, nodes maintain clean arrival direction clustering. At CL 29 (transition point), some nodes begin to be misclassified—note the example node (red box) that arrives from NORTH but appears in the WEST cluster. At CL 40, nodes collapse into four entangled clusters with no clear organizing principle. Right panel shows the Foraging model at CL 50 (unaveraged) for comparison, demonstrating consistent arrival direction clustering across all context lengths with no such transition or confounding effects.



Latex template/figures/combinedsph.png

Figure 5.4: **Horizontal Mirroring Effect in SP-Hamiltonian Model.** Left panel shows PCA representations for layers 1 and 2 (columns) with three different coloring schemes (rows): arrival direction (AD), coordinates, and path index. The horizontal mirroring pattern is evident across all layers and coloring schemes. Right panel visualizes the specific coordinate pairs that demonstrate perfect horizontal mirroring: coordinates (0,0) and (0,2) appear nearly identical in PCA space, as do coordinates (2,0) and (2,2), illustrating the model's symmetric representation across the central horizontal axis of the 4×4 grid. This mirroring effect explains why the model shows overlapping north/south directions while maintaining distinct east/west clusters.



Figure 5.5: **Direction token ablation results for SP models.** Comparison of task accuracy when historical direction tokens are zeroed out at the input to each layer. Both the SP-H (blue) and SP-RW (brown) models show a gradual recovery, demonstrating a continuous reliance on direction tokens throughout the network.

Chapter 6

Conclusion

The investigation of spatial reasoning in transformers has revealed how training objectives and data shape the computational strategies that emerge during learning. What began as a question about whether transformers learn cognitive maps or heuristics has evolved into an understanding of how different training frameworks produce distinct computational architectures. By systematically analysing three models trained on different spatial navigation tasks, we have revealed how training objectives shape not just performance, but the underlying computational strategies that emerge during learning. The work presented here suggests that the question of whether transformers learn universal cognitive maps or task-specific heuristics is not binary, but reveals a spectrum of spatial intelligence that emerges from the interaction between training objectives and data structure. These different forms of intelligence reflect distinct computational architectures that can be causally dissected and understood. The implications extend beyond spatial reasoning to how training paradigms shape intelligence emergence in neural networks.

6.1 The Foraging Model’s Adaptive Strategy

Perhaps the most significant discovery was the Foraging Model’s adaptive computational strategy. Rather than using a single approach, the model appears to employ qualitatively different algorithms depending on context length. With minimal context (2-3 steps), the model appears to rely on local heuristics, specifically a strong bias toward reversing the previous move: a conservative strategy that avoids invalid moves across all grid configurations. This heuristic-based approach achieves perfect accuracy because it exploits the geometric constraints of grid navigation without requiring global spatial understanding. Our mechanistic analysis provides a direct causal explanation for this behaviour. We found a consistent attention pattern in Layer 1 where multiple heads focus on the penultimate node when processing a direction. For a 2-step path, this circuit directly implements the reversal heuristic, solving the task in the earliest layers. Convergent evidence comes from our direction ablation experiments, where performance on 2-hop loops recovers to 100% by Layer 2, confirming the existence of a specialised, low-level circuit for these trivial cases.

However, for longer paths, this local heuristic is insufficient. The model transitions to a more deliberative, map-based strategy. The critical transition occurs around Layer 7, where the model’s internal coordinate system appears to become causally self-sufficient. The direction ablation experiments for complex loops (4-12 hops) reveals a sharp transition to 100% accuracy between layers 6-8, suggesting that the model has consolidated spatial information into node representations, making explicit directional history redundant. The simple Layer 1 attention pattern appears to be a foundational building block used in two distinct ways: as a standalone heuristic for simple problems and as the initial input to a

deeper, multi-layer computation for complex ones.

This dual strategy suggests how transformers can develop hierarchical reasoning capabilities. Rather than learning a single, fixed algorithm, the Foraging Model appears to have developed an adaptive system that switches between computational strategies based on information availability. The model’s ability to maintain high performance across diverse contexts suggests that it has learned not just spatial reasoning, but strategies for selecting appropriate reasoning approaches.

6.2 Training Paradigms as Algorithmic Scaffolding

Our comparative analysis shows that training frameworks act as algorithmic scaffolding, shaping not just performance but the computational strategies a model develops. On one end of the spectrum, we have the Foraging Model. Its three-stage processing pipeline (directional processing, spatial integration, then functional refinement) seems to have emerged from the inductive bias of passive, exploratory learning on high-variance random walks. Without explicit goals or structured data, the model appears to have developed a reusable representation of spatial structure. The emergence of a self-sufficient coordinate system by Layer 7 suggests how exploratory objectives may foster the development of allocentric, map-like representations that support flexible generalisation.

In contrast, the Shortest Path models appear to have developed continuous, path-dependent computational strategies that never achieve the self-sufficiency observed in the Foraging Model. The direction ablation experiments revealed that both SP-Hamiltonian and SP-Random Walk models remain continuously dependent on explicit directional information throughout all 12 layers, with gradual recovery rather than the sharp phase transition observed in the Foraging Model, perhaps implementing something similar to procedural path integration. The SP-Hamiltonian model illustrates how goal-directed training on highly structured data can lead to specialised but brittle computational strategies. The model’s horizontal mirroring in its representations is indicative of a strategy that exploits the regularities in its training distribution (the symmetric, structured nature of Hamiltonian paths) which breaks down when confronted with unstructured data or novel spatial configurations.

The SP-Random Walk model’s intermediate position on this spectrum demonstrates the nuanced effects of fine-tuning. By training on random walks, it was forced to abandon SP-H’s brittle shortcuts, leading to better generalisation to larger grids. This suggests that fine-tuning on random walks encourages the formation of reusable map-like representations. However, despite developing representations more similar to the Foraging Model, mechanistic analysis suggests that it retains the same continuous, path-dependent computational strategy as SP-Hamiltonian. The robustness improvement appears to stem from representational adaptation rather than algorithmic change, where the existing algorithm learned to work with more flexible representations while maintaining the same mechanistic dependencies. However, the comparison between models is complicated by task difficulty differences. The SP models face inherently harder tasks (finding optimal paths) compared to the Foraging Model’s simpler next-step prediction. While the Foraging Model shows superior generalisation to larger grids and novel spatial configurations, this may reflect both its training paradigm and the relative simplicity of its core task. The SP models’ path-dependent strategies, while less generalisable, may be more appropriate for their specific goal-directed objectives.

6.3 Broader Implications for AI Development

The findings of this investigation have implications for how we design and train AI systems for complex reasoning tasks. The exploration-exploitation trade-off observed in biological systems appears to apply to artificial neural networks as well. Training exclusively on optimal, expert demonstrations may lead to powerful but narrow strategies that fail to generalise to novel situations. The SP-Hamiltonian model’s brittleness despite perfect in-distribution performance illustrates this principle clearly. Conversely, incorporating exploration, suboptimal data, and randomness into training paradigms may be essential for encouraging the development of robust, generalisable world models. The Foraging Model’s success in developing cognitive map-like representations from random walks suggests that the apparent ‘inefficiency’ of passive exploration may be a necessary catalyst for structural learning.

The discovery of adaptive computational strategies in the Foraging Model suggests that robust AI systems may need to develop meta-cognitive capabilities—the ability to select appropriate reasoning strategies based on available information. This adaptive approach, rather than fixed algorithms, may be key to building systems that can handle diverse and novel situations. The fine-tuning analysis reveals important insights about how to improve existing systems. Rather than expecting fine-tuning to fundamentally alter learned algorithms, we should anticipate representational adaptation that allows existing computational strategies to work with new data distributions.

6.4 Limitations and Future Directions

Several important limitations constrain the scope of these findings and suggest directions for future research. The analysis is limited to GPT-2 Small architecture and 4×4 grid environments, raising questions about how these patterns would extend to larger models or more complex environments. The mechanistic analysis of all models, but particularly the Shortest Path models, remains incomplete. While we have suggested that these models rely on continuous, path-dependent computation, we have not confirmed nor fully reverse-engineered the specific computations that perform path planning. Future work could use more granular patching experiments—for instance, patching the representations of start and goal tokens—to trace how models select and generate optimal paths. The 29th node transition in the SP-RW model represents an intriguing phenomenon that remains unexplained. Whether this reflects the limited strength of fine-tuning, competing pressures from different training paradigms, or some other mechanism requires further investigation.

The generalisability of these findings to other domains and architectures remains to be established. While spatial reasoning provides an ideal testbed for mechanistic analysis, it remains unclear how these insights would apply to other cognitive capabilities or different model architectures. Future research could extend this analysis to larger models, more complex environments, and different cognitive domains. The methodological framework developed here—combining behavioural, representational, and mechanistic analysis—could be applied to study other emergent capabilities in neural networks. The connection to biological intelligence remains largely unexplored. While the findings suggest parallels between artificial and biological learning, deeper investigation of these connections could provide insights into both artificial and biological intelligence.

Chapter 7

Appendices

7.1 Loss Curves and Training Dynamics

Analysis of training loss curves reveals interesting phase transitions during training, particularly for the SP-Hamiltonian model. Detailed examination of the model’s output during training reveals a clear progression through four distinct learning phases that correspond to the loss curve transitions.

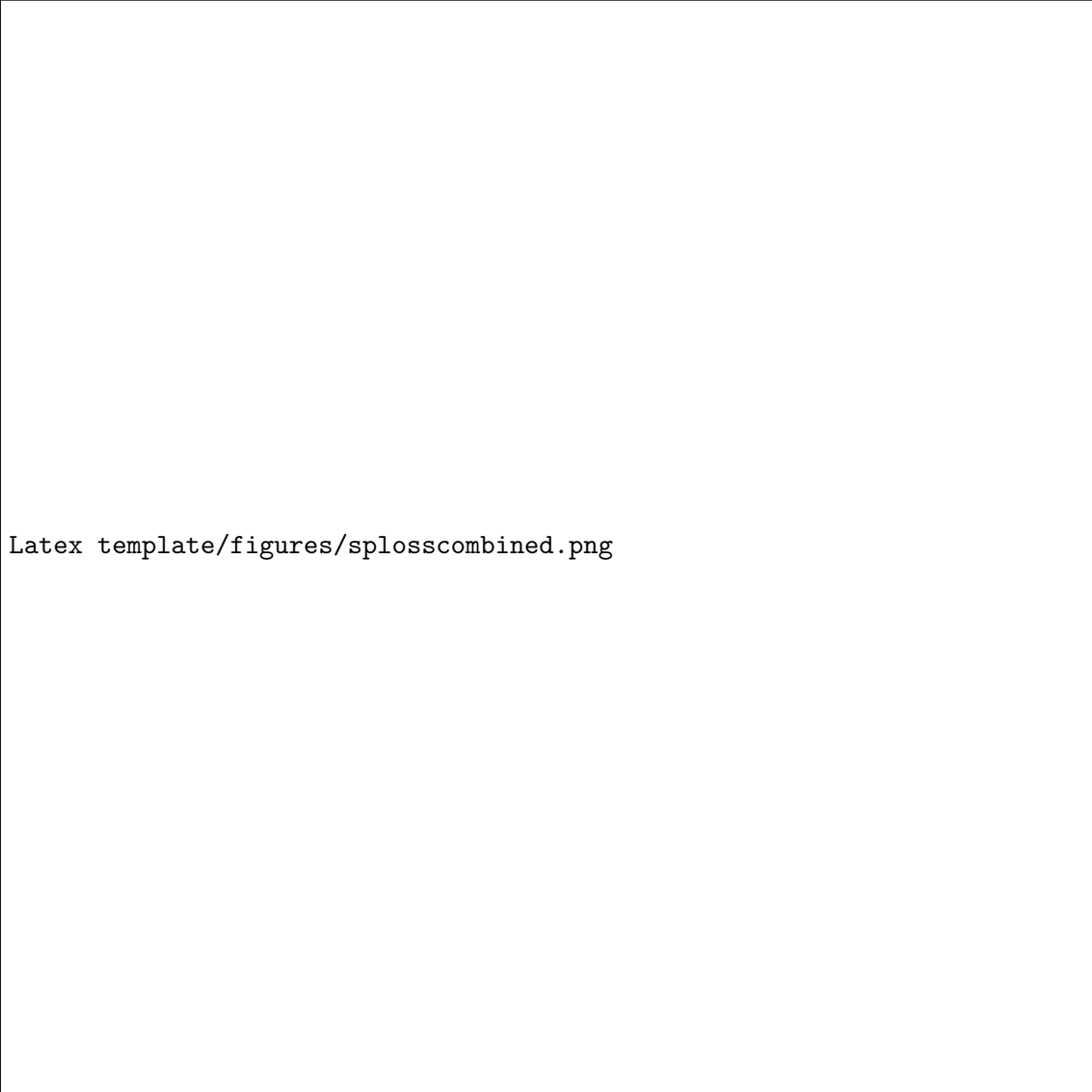
Observations of the SP-Hamiltonian model’s output during training suggest a phases of learning progression that appears to correspond to the loss curve transitions. In the first phase, the model appears to learn the basic node-direction-node syntax pattern, producing outputs that follow the correct grammatical structure but with random content. The second phase shows the model beginning to correctly predict start and end nodes while the middle path remains nonsensical, potentially indicating initial spatial boundary learning. The third phase represents what appears to be a critical transition where the model begins to get directions correct but hallucinates invalid node names, possibly suggesting attention mechanism failures during intermediate path generation. Finally, the fourth phase demonstrates the emergence of valid complete paths as the model appears to successfully integrate spatial reasoning with proper node prediction. This observed progression does not necessarily reflect the model’s final solution strategy but rather may reveal how it converges to spatial reasoning capabilities through what appears to be incremental learning of different task components.

In contrast, the SP-Random Walk model exhibits no such phase transitions during training, showing smooth convergence without the phase transitions observed in SP-Hamiltonian. This difference supports the claim in Chapter 4 that the this model does not undergo algorithmic changes during learning, instead learning to adapt its strategy for random walk contexts.

7.2 Loop Completion Task Templates

The loop completion tasks used throughout this thesis follow systematic templates that create geometrically valid paths returning to their starting nodes. These templates are designed to test the models’ ability to understand spatial relationships and geometric constraints across different levels of complexity.

These templates create systematic variations that test different aspects of spatial reasoning. The 2-hop loops represent simple reversal patterns that can be solved using local heuristics, while longer loops (4-12 hops) require more sophisticated spatial understanding and global reasoning. Each template is designed to create geometrically valid paths on a grid, ensuring that the models must understand spatial relationships rather than simply



Latex template/figures/splosscombined.png

Figure 7.1: **Training loss curves for SP models showing convergence patterns and phase transitions.** SP-Hamiltonian shows multiple phase changes during training, while SP-RW (fine-tuned from SPH) shows smoother convergence.

memorizing token sequences. The systematic nature of these templates allows for controlled testing of how model performance scales with task complexity, providing insights into the computational strategies employed by different models.

7.3 Additional Experimental Results

7.3.1 Extended PCA Analysis

Detailed PCA analysis examining representational evolution across all 12 transformer layers for both the Foraging Model and SP-Random Walk model. The analysis uses un-averaged node token representations extracted from 1000 unique random walks of length 50 on 3×3 grids.

Both models exhibit similar initial representational patterns in Layer 1, with four distinct clusters corresponding to each arrival direction and start nodes forming separate clusters. However, the models diverge in their representational evolution across layers. The Foraging Model undergoes a computational transition between layers 7 and 12, where the organising principle shifts from coordinate-based spatial structure to functional clustering based on available navigational directions. This transition reflects the model’s progression from spatial position encoding to action-oriented representation. In contrast, the SP-Random Walk model maintains remarkably stable representational patterns across all layers, showing only minor geometric transformations (slight cluster shearing) without fundamental changes in organisational structure. This stability suggests that SP-RW develops a consistent representational strategy that remains effective throughout the network.

7.3.2 Layer Redundancy and Ablation Analysis

This section presents comprehensive layer ablation analyses to understand the redundancy and criticality of different layers in the Foraging Model. We perform two complementary analyses: (1) individual layer ablation to understand the contribution of each layer to overall performance, and (2) systematic ablation of all possible layer combinations to identify minimal required layers.

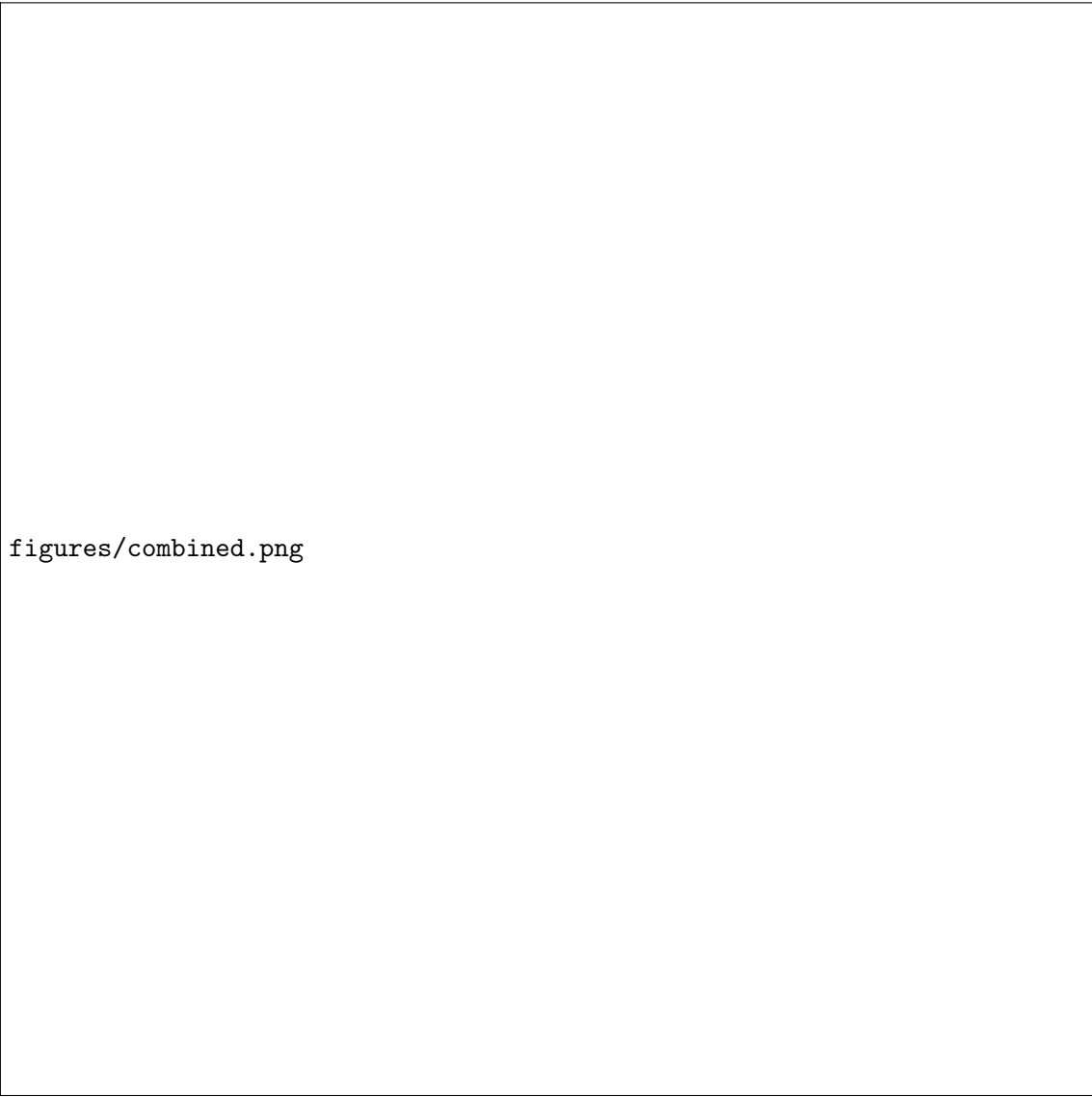
7.3.2.1 Individual Layer Ablation

We tested all 12 layers individually using 500 evaluation cases, measuring next-token prediction accuracy for each ablation. We systematically zeroed out layer outputs during inference, with 95% confidence intervals calculated for binomial proportions.

The results reveal that Layer 1 is indispensable, causing complete performance collapse (0% accuracy) when ablated, highlighting its foundational role in processing information for the task. Layers 5-12 seem to be redundant, with zero impact on performance (100% accuracy maintained), indicating that 67% of the model’s layers can be individually ablated with no impact on task performance. Layers 2-4 show minor contributions with small performance drops (96-99% accuracy).

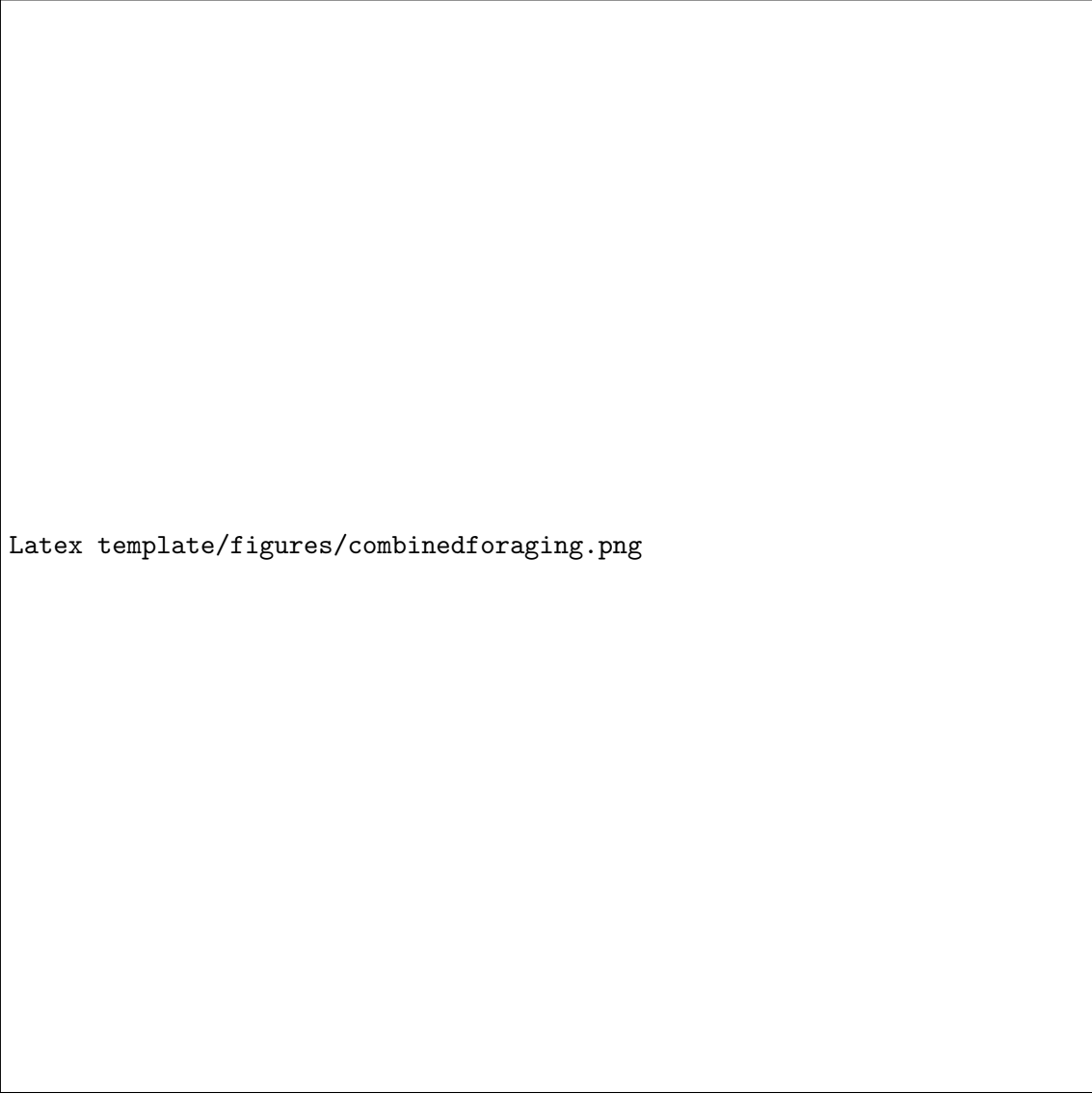
7.3.3 Systematic Layer Combination Ablation

We tested all possible combinations of k layers ($k = 1$ to 11) using 50 evaluation cases, totalling $\sum_{k=1}^{11} C(11, k) = 2^{11} - 1 = 2047$ combinations. Crucially, we exclude Layer 1 from this experiment, since it was found to be indispensable. This systematic approach reveals how the model’s performance degrades as more layers are removed simultaneously.



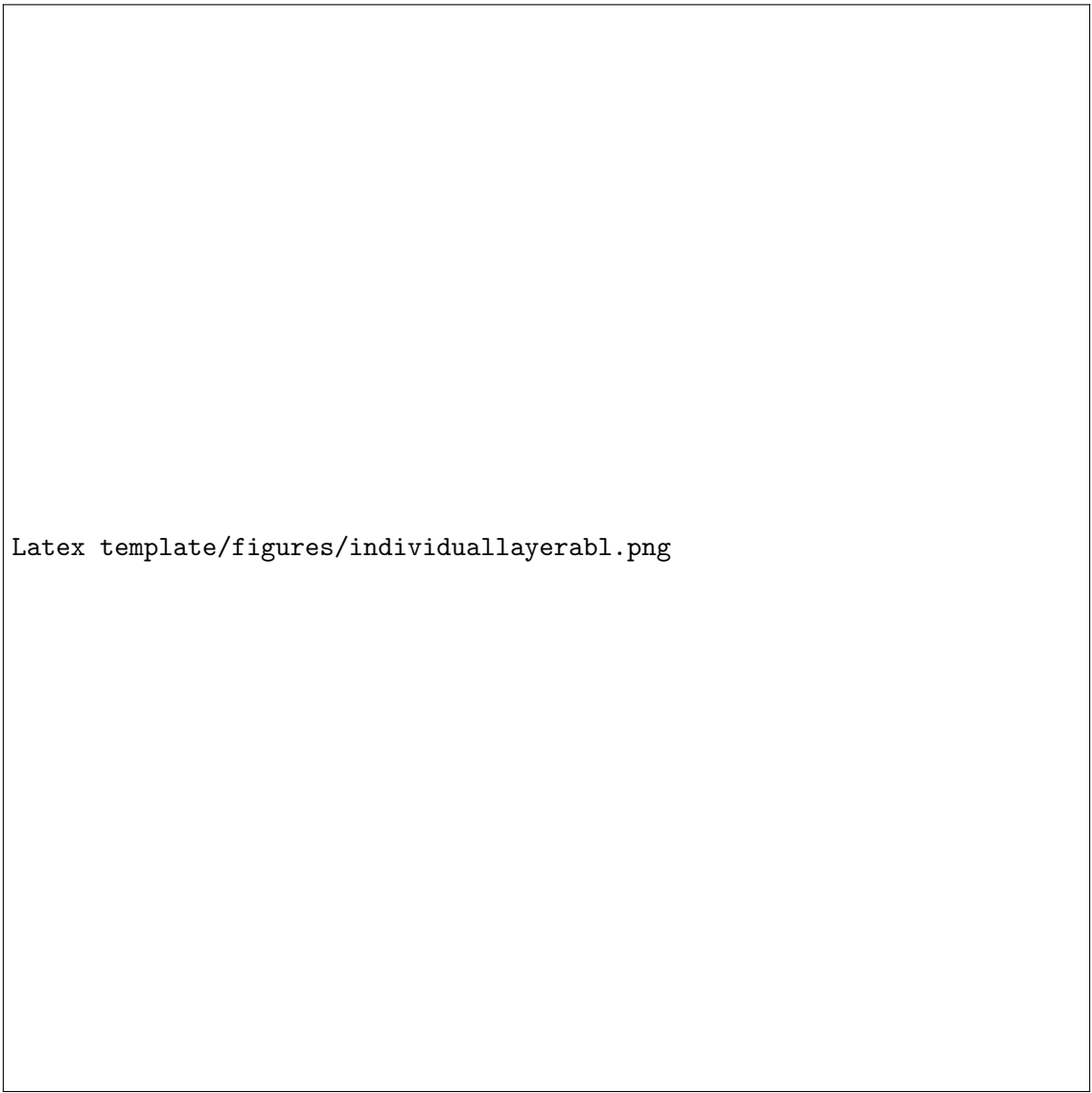
figures/combined.png

Figure 7.2: **Detailed PCA Analysis of SP-RW Model.** Three columns show layers 1, 7, and 12 (left to right). For each layer, the top row shows points coloured by arrival direction (AD), the middle row shows points coloured by coordinates, and the bottom row shows points coloured by path index.



Latex template/figures/combinedforaging.png

Figure 7.3: **Detailed PCA Analysis of the Foraging Model.** Three columns show layers 1, 7, and 12 (left to right). For each layer, the top row shows points coloured by arrival direction (AD), the middle row shows points coloured by coordinates, and the bottom row shows points coloured by path index.



Latex template/figures/individuallayerabl.png

Figure 7.4: **Individual layer ablation analysis showing the impact of ablating each layer on task accuracy.** Layer 1 is absolutely critical (0% accuracy when ablated), while layers 5-12 are individually redundant (100% accuracy maintained). Layers 2-4 show minor contributions with slight performance drops.

The systematic ablation reveals a clear performance degradation pattern: ablating 1-2 layers has minimal impact (95-100% accuracy), 3-4 layers causes moderate degradation (75-85% accuracy), 5+ layers leads to severe performance collapse (50% or below), and 9+ layers results in near-complete failure (5% or below). While individual later layers are redundant, the model still requires multiple layers working together, supporting the hypothesis of distributed rather than localised computation.

The analysis reveals significant overparameterisation, with 67% of layers being completely redundant for individual ablation. Only Layer 1 is crucial for reasoning performance, suggesting the model could potentially be compressed without significant performance loss. Individual ablation results support the Chapter 4 finding that Layer 1 implements critical directional processing circuits, while the redundancy of layers 5-12 supports the claim that Layer 7 represents a transition point where the coordinate system becomes self-sufficient. Layer redundancy suggests distributed rather than localised computation, consistent with the three-stage processing pipeline identified in Chapter 4. Limitations include task specificity, the ablation method potentially not reflecting natural layer importance, and interaction effects between layers.

7.3.4 Investigation of the 29th Node Effect

The SP-Random Walk model exhibits a puzzling representational transition at the 29th node position in context walks, where node representations collapse from clean arrival-direction clustering into entangled clusters with no obvious organisational principle. To investigate whether this representational change corresponds to functional limitations, we designed three targeted memory tests that probe the model’s ability to use information from different temporal positions in the context walk.

We tested three conditions: (1) *Both Nodes Late*: Both start and goal nodes appear after position 29 in the context walk, (2) *Start Early Goal Late*: Start node appears before position 29, goal node after, and (3) *Constrained Expansion*: Context walk is constrained to a 3×3 inner grid for the first 29 positions, then expands to the full 4×4 grid. Each condition was tested on 1000 evaluation cases using the SP-RW model.

The results demonstrate that the 29th node representational transition does not correspond to functional limitations. All three memory tests show high accuracy (92-98%), with the constrained expansion test achieving the highest performance (97.60%). This suggests that the representational collapse observed in PCA analysis represents a change in how information is encoded rather than a loss of functional capability. The model maintains robust spatial reasoning performance regardless of whether critical nodes appear before or after the 29th position threshold, indicating that the transition is likely a representational artifact rather than a fundamental computational limitation. This finding supports the interpretation that the 29th node effect reflects the model’s adaptation to variable-length training data rather than a breakdown in spatial reasoning capabilities.

7.3.5 Extended Direction Ablation Analysis

To further investigate the SP-Hamiltonian model’s directional processing, we conducted extended ablation experiments that selectively ablate different subsets of directional tokens. This analysis tests whether the model treats all four cardinal directions equivalently or shows asymmetric processing patterns.

We tested three ablation conditions: (1) ablating all directional tokens (NORTH, SOUTH, EAST, WEST), (2) ablating only NORTH/SOUTH tokens, and (3) ablating only EAST/WEST tokens. For each condition, we systematically ablated directional tokens at each layer and measured shortest path accuracy on 500 test cases.

The results demonstrate that the SP-Hamiltonian model treats all four cardinal directions equivalently. Ablating NORTH/SOUTH tokens produces identical performance degradation to ablating EAST/WEST tokens across all layers, with both conditions showing the same gradual recovery pattern. This symmetric processing supports the claim that the model’s functional use of directional information is globally symmetric, even though PCA analysis showed subtle representational differences along one axis. The finding contradicts any interpretation that the model has asymmetric directional processing capabilities, confirming that the horizontal mirroring observed in PCA represents a representational artifact rather than functional asymmetry.

Table 7.1: Loop completion task templates for different hop counts. Each template uses placeholders ({}) for node names and creates geometrically valid paths that return to the starting node. The final placeholder represents the target node that completes the loop. Direction abbreviations: E=EAST, W=WEST, N=NORTH, S=SOUTH.

Hops	Template Examples
2	{ } E { } W { } { } N { } S { }
4	{ } E { } S { } W { } N { } { } N { } E { } S { } W { }
6	{ } E { } E { } N { } W { } W { } S { } { } N { } N { } W { } S { } S { } E { }
8	{ } E { } E { } E { } S { } W { } W { } W { } N { } { } S { } S { } S { } E { } N { } N { } N { } W { }
10	{ } E { } E { } E { } S { } S { } W { } W { } W { } N { } N { } { } S { } S { } S { } E { } E { } N { } N { } N { } W { } W { }
12	{ } E { } E { } E { } S { } S { } S { } W { } W { } W { } N { } N { } N { } { } S { } S { } S { } E { } E { } E { } N { } N { } N { } W { } W { } W { }

Table 7.2: **Performance on tests probing the 29th node transition effect.** All conditions show high accuracy (92-98%), indicating that the representational collapse at position 29 does not impair functional performance.

Test Condition	Accuracy
Both Nodes Late	93.20%
Start Early Goal Late	92.80%
Constrained Expansion	97.60%

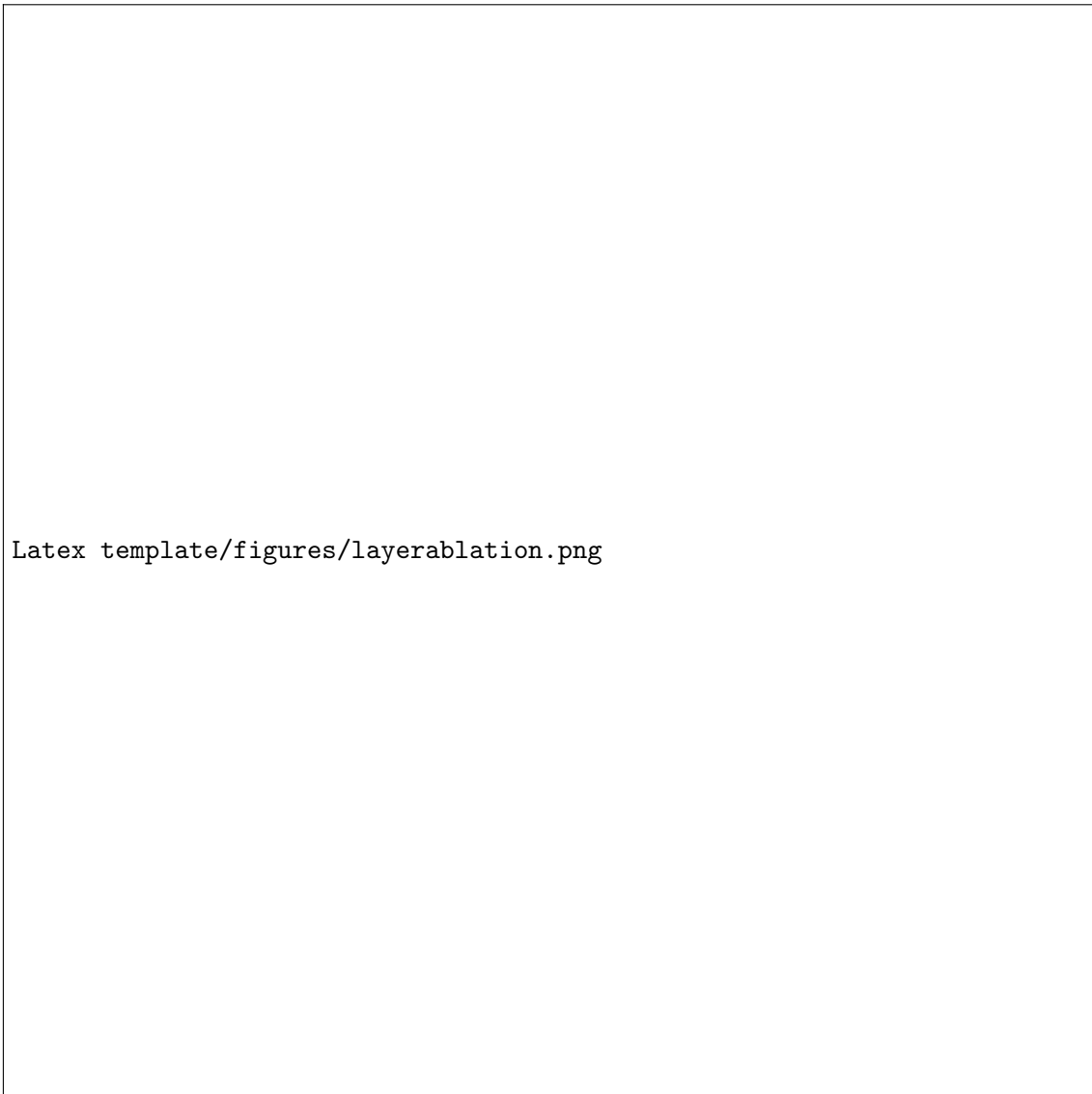
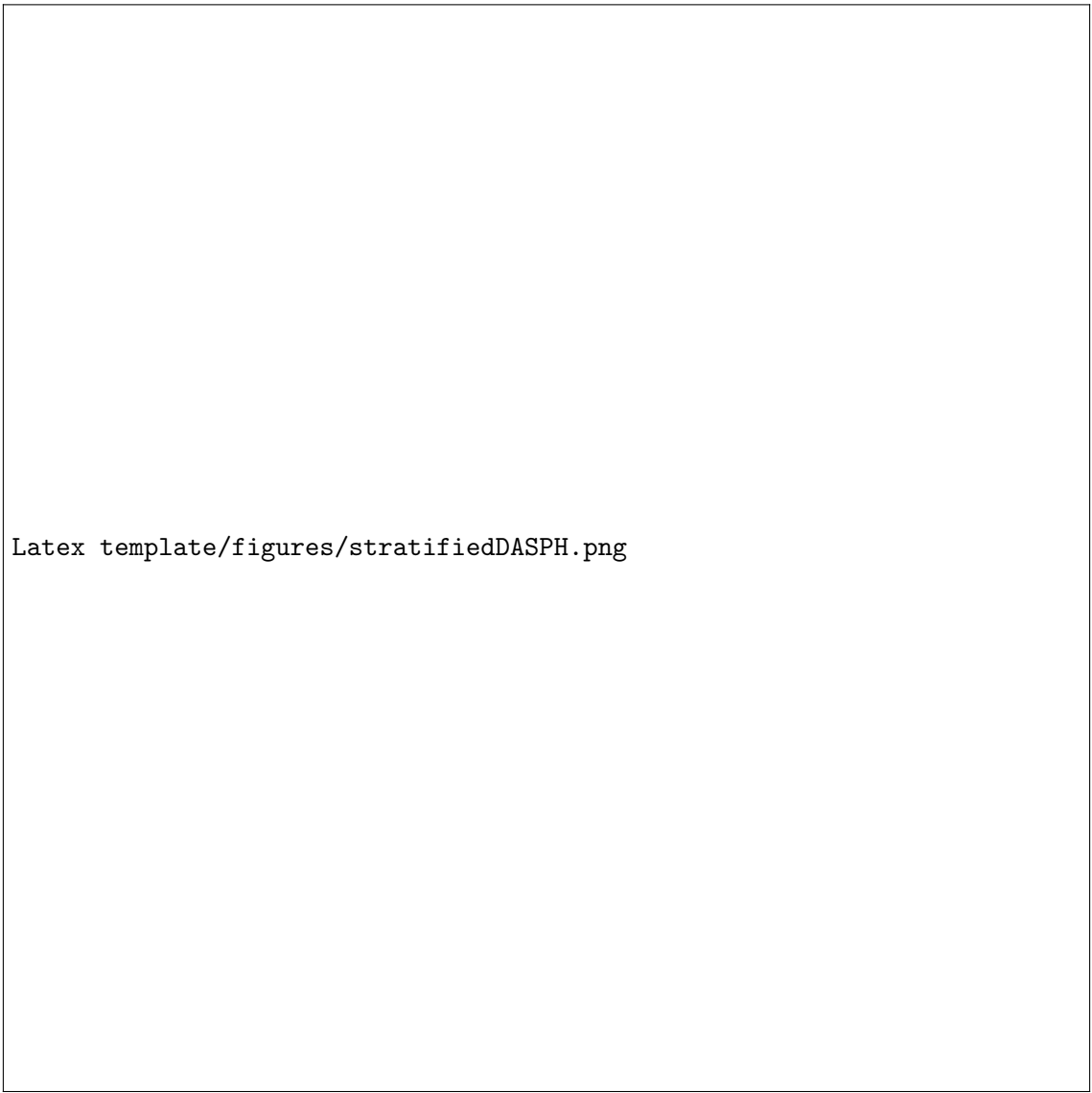


Figure 7.5: **Box plot showing the relationship between the number of ablated layers and model accuracy.** The plot demonstrates a strong inverse relationship: as more layers are ablated, performance degrades significantly. The model can tolerate ablating 1-2 layers with minimal impact, but ablating 5+ layers leads to severe performance collapse.



Latex template/figures/stratifiedDASPH.png

Figure 7.6: **Direction ablation analysis showing the impact of ablating different subsets of directional tokens across layers.** The identical performance curves for NORTH/SOUTH and EAST/WEST ablation demonstrate that the model treats all four cardinal directions equivalently, supporting claims about symmetric directional processing.