On Hierarchical Encoding and Reasoning in Deep Transformer-based Generative Models

by

Dean Lewis Slack

A thesis presented for the degree of

Doctor of Philosophy

in the

Department of Computer Science

Durham University

United Kingdom

May 2025

Declaration

The work in this thesis is based on research carried out in the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Abstract

Recent advances in generative Transformer-based foundation models have driven remarkable progress in artificial intelligence, yet their internal mechanisms for representing complex hierarchical structures remain largely unknown, posing significant challenges for interpretability, safety, and robust generalisation. This thesis aims to progress on these issues by systematically investigating how such models internalise hierarchical structures, the relationship between this learning and behaviours like generalisation versus memorisation, and how hierarchical principles can inform the development of safer, more accurate, generative models. To this end, we first introduce novel probing techniques to map the layer-wise emergence of linguistic hierarchies in language models and extend this analysis to the visual domain by developing PSViT: a pixel-space Transformer with hierarchical decompositions of video image patches, shown to learn and generalise hierarchical physical dynamics from raw video data. We investigate memorisation during fine-tuning, establishing an n-gram based early warning signal for verbatim leakage and proposing scalable defences that promote structural generalisation over verbatim memorisation. Building on these insights, we further demonstrate that a unified next-frame prediction framework enables a single model to process text, images, audio, and video without modality-specific encoders, thereby learning shared hierarchical patterns across these diverse inputs. Collectively, our findings underscore that the capacity to learn and represent hierarchical structure is a fundamental characteristic of Transformer models, and that a focused analysis of these underpinnings is crucial for advancing more capable, interpretable, and safer artificial intelligence.

Acknowledgements

This PhD has been a challenging and turbulent journey, and completing it would not have been possible without the support of many people.

I would like to express my deepest gratitude to my PhD supervisors, Dr. Noura Al Moubayed and Dr Mariann Hardey. Your guidance, encouragement, and belief in me through thick and thin - from the daunting unknowns to the light at the end of the tunnel - will always be remembered.

To Ugnė, for your boundless love, patience, and support.

To my family – Dad, Mum, Sean, and James – thank you for your constant, unwavering support throughout my life, and to my Grandma, whose memory remains a cherished source of inspiration.

My sincere thanks go to everyone at Wordnerds, particularly Hugh, Steve, and Pete, who took a chance on me and offered such a welcoming environment. The insights, experience, and warmth you have all shown will never be forgotten.

To everyone in Noura's research group, past and present, and to all my fellow Durham PhD students and the friends I've made along the way: your company, friendship, insightful discussions, and shared laughter have been lifelines that helped me overcome the doubts I had of reaching this point.

You have all, in so many ways, defined this journey for me.

Contents

Declaration i

Abstract ii

Acknowledgements iii

List of Figures ix

List of Tables xi

Nomenclature xii

1 Introduction 1

1.1 Motivation . . . 3

1.2 Research Objectives and Contributions . . . 6

1.3 Publications . . . 7

1.4 Thesis Structure . . . 7

2 Background 14

2.1 Machine Learning and Probability Theory . . . 15

2.1.1 Foundations . . . 15

2.1.2 Machine Learning in Natural Language Processing . . . 18

2.1.3 Machine Learning in Computer Vision . . . 19

2.1.4 Model Capacity, Overfitting, and Generalisation . . . 20

2.2 Language Modelling . . . 21

2.2.1 Causal Language Modelling . . . 22
2.2.2 Masked Language Modelling . . . 23
2.2.3 Self-Supervised Learning . . . 24
2.2.4 Text-to-Text and In-Context Learning . . . 26
2.2.5 Pretrained Language Models . . . 27
2.2.6 Fine-tuning and Feature Extraction . . . 28
2.2.7 Evaluating Language Models . . . 29
2.2.8 Relevance to Transformer Architectures . . . 30
2.2.9 Summary . . . 31
2.3 Transformer Architecture . . . 32
2.3.1 Model Overview . . . 33
2.3.2 Input and Positional Embeddings . . . 33
2.3.3 Self-Attention . . . 35
2.3.4 Feed-Forward Network . . . 38
2.3.5 Stacked Layers and Output Layer . . . 38
2.3.6 Masking Strategies . . . 39
2.3.7 Hierarchical Representations and In-Context Learning . . . 40
2.3.8 Representative Transformer Models . . . 41
2.3.9 Summary . . . 44
2.4 Hierarchical Reasoning . . . 45
2.4.1 Hierarchy in Language . . . 46
2.4.2 Hierarchical Representations in Language Models . . . 49
2.4.3 Hierarchy in Vision . . . 51
2.4.4 Integrating Language and Vision Hierarchies . . . 54
2.4.5 Conclusion and Outlook for Hierarchical Reasoning . . . 57
2.5 Epilogue . . . 58

3 Hierarchical Information in Contextual Representations 76

3.1 Introduction . . . 77
3.2 Pre-trained Transformer Contextualisers . . . 79
3.3 Methodology and Datasets . . . 81

3.3.1 Ancestor Sentiment Classification . . . 81
3.3.2 Ancestor Constituency Phrase Tagging . . . 83
3.3.3 Fine-tuned Layer Performances . . . 84
3.3.4 Non-linear Experiments . . . 85
3.4 Results and Discussion . . . 85
3.4.1 Layer Performance Distributions . . . 87
3.4.2 Fine-tuning Comparisons . . . 91
3.4.3 Non-linear Classifiers . . . 92
3.5 Conclusion . . . 92
3.6 Epilogue . . . 93

4 Hierarchy in Language Model Memorisation 98

4.1 Introduction . . . 99
4.2 Related Work . . . 102
4.2.1 Measuring Memorisation . . . 102
4.2.2 Memorisation in Pre-training Versus Fine-tuning . . . 102
4.2.3 Mitigation Strategies and Regularisation . . . 103
4.3 Methodology . . . 104
4.3.1 Memorisation Metrics . . . 104
4.3.2 n-gram Memorisation . . . 104
4.3.3 Datasets . . . 105
4.3.4 Pre-trained Models . . . 106
4.3.5 Fine-Tuning Approach . . . 106
4.4 Results and Discussion . . . 108
4.4.1 n-gram Memorisation Predicts Verbatim Memorisation . . . 108
4.4.2 Selection Criteria as Mitigation . . . 110
4.4.3 Comparing Mitigation Strategies . . . 114
4.5 Limitations . . . 115
4.6 Conclusion and Future Work . . . 116
4.6.1 Mitigation strategies . . . 117
4.7 Epilogue . . . 117

4.8 Appendix: Datasets . . . 119
4.9 Appendix: n-gram Regularisation Loss . . . 121

5 Spatiotemporal Reasoning in Video **126**

5.1 Introduction . . . 127
5.2 Related Works . . . 130
5.2.1 Video Models . . . 131
5.2.2 Autoregressive Video Models . . . 132
5.2.3 Dynamic Simulation Modelling . . . 133
5.3 Autoregressive Video Prediction . . . 134
5.3.1 Task Definition . . . 134
5.3.2 Autoregressive Learning Objective . . . 134
5.4 Model Architecture . . . 135
5.4.1 Input Patch Processing . . . 135
5.4.2 Patch Embedding Representation . . . 136
5.4.3 Spatiotemporal Attention Strategies . . . 136
5.4.4 Spatial Attention . . . 138
5.4.5 Causal Temporal Attention . . . 140
5.4.6 Layer Outputs . . . 140
5.4.7 U-Net Style Adaptation . . . 140
5.4.8 Register Tokens . . . 141
5.5 Methodology . . . 142
5.5.1 Datasets . . . 142
5.5.2 Object Divergence Metrics . . . 145
5.5.3 Positional Encodings . . . 146
5.5.4 Self-attention Strategies . . . 146
5.5.5 Training Setup . . . 147
5.5.6 Existing Approaches . . . 147
5.6 Results . . . 150
5.6.1 Video Prediction Results . . . 150
5.6.2 Comparing Input Context Sizes . . . 153

5.6.3 Comparison with Existing Approaches . . . 154
5.6.4 Qualitative Assessment . . . 156
5.7 Discussion and Structural Analysis . . . 157
5.7.1 Spatial and Temporal Reasoning . . . 158
5.7.2 Attention Head Mechanisms . . . 159
5.7.3 PDE Dynamics Information Probing . . . 160
5.7.4 Limitations . . . 162
5.8 Conclusion and Future Work . . . 162
5.9 Epilogue . . . 163

6 Multimodal Multi-Task Hierarchical Reasoning 174
6.1 Introduction . . . 175
6.2 Related work . . . 177
6.3 Method . . . 180
6.3.1 Reformulation . . . 180
6.3.2 Video Prediction Model . . . 184
6.3.3 Training Details . . . 185
6.4 Experiments . . . 186
6.5 Results . . . 187
6.5.1 Attention Maps . . . 191
6.5.2 Limitations and Future work . . . 193
6.6 Conclusion . . . 194
6.7 Epilogue . . . 195

7 Discussion & Concluding Remarks 200
7.1 Contributions . . . 200
7.2 Limitations and Future Work . . . 203
7.3 Epilogue . . . 204

List of Figures

2.1 Illustration of the Transformer Architecture 34

3.1 Constituency parse tree illustration 82
3.2 Linear probing classifier performance for BERT and GPT-2 (Raw) . . . 88
3.3 Linear probing classifier performance for Reformer 89
3.4 Linear probing classifier performance for XLNet 90

4.1 Rate of verbatim memorisation over fine-tuning 100
4.2 Partial memorisation over fine-tuning epochs for different datasets . . . 107
4.3 Partial memorisation over fine-tuning for different models 110
4.4 Partial memorisation over fine-tuning for various fine-tuning depths . . 111
4.5 Memorisation and performance for different optimisation criteria 112

5.1 Example video prediction model outputs 129
5.2 PSViT model illustration and space-time block 137
5.3 Spatiotemporal attention masking strategies 139
5.4 PSViT outputs on Fluid and CLEVRER datasets 151
5.5 PSViT model outputs for simulation datasets 152
5.6 PSViT video prediction performance over time 153
5.7 Impact of input context length on object divergence 154
5.8 Object divergence over time comparing approaches 155
5.9 PSViT outputs on Moving MNIST 157
5.10 Positional encodings and self-attention activations 158
5.11 Attention activation heatmaps on CLEVRER 160

5.12 PSViT outputs on CLEVRER (Appendix) . . . 165
5.13 PSViT outputs on Fluid (Appendix) . . . 166
5.14 PSViT outputs on simulation datasets (Appendix) . . . 167

6.1 Example of SST dataset as a video . . . 181
6.2 Example of CIFAR10 dataset as a video . . . 181
6.3 Example of TinyVIRAT dataset as a video . . . 182
6.4 Example of CLEVRER dataset as a video . . . 183
6.5 Example of colorization dataset as a video . . . 183
6.6 Example of LaSOT dataset as a video . . . 184
6.7 Example of AudioMNIST dataset as a video . . . 185
6.8 Sample outputs on multimodal task reformulations . . . 189
6.9 Attention activation heatmaps for different multimodal tasks . . . 192

List of Tables

2.1 Notable Transformer models and families 43

3.1 Probing results for best performing layers per contextualiser for Phrase Tagging Sentiment 86

3.2 Probing results for best performing layers per contextualiser for Phrase Tagging 87

3.3 Linear and non-linear probing results for SST-5 91

4.1 Summary and grouping of datasets used for fine-tuning. 103

4.2 Memorisation mitigation results for all models and datasets 113

5.1 Dataset Specifications 143

5.2 Parameter Estimation Details for Probing Hierarchical Knowledge . . . 143

5.3 Model Configurations. 148

5.4 Video Prediction Results and Divergence Scores. Divergence scores assess the model's ability to predict object trajectories accurately over time. 149

5.5 Video prediction performance on BAIR and Moving MNIST 156

5.6 Parameter estimation probing results 161

6.1 Training configurations for each task. Input/Target Lengths are in number of frames. 186

6.2 Multimodal task reformulation results 188

Nomenclature

AI Artificial Intelligence

CLM Causal Language Model

CNN Convolutional Neural Network

CV Computer Vision

FFN Feed Forward Network

GPT Generative Pretrained Transformer

LLM Large Language Model

LSTM Long Short-Term Memory Model

MLE Maximum Likelihood Estimation

MLM Masked Language Model

MSE Mean Squared Error

NLI Natural Language Inference

NLP Natural Language Processing

NLU Natural Language Understanding

PDE Partial Differential Equation

PLM Pretrained Language Model

PSNR Peak Signal-to-Noise Ratio

PSViT Pixel-Space Spatiotemporal Video Transformer

QA Question Answering

RNN Recurrent Neural Network

SSIM Structural Similarity Index Measure

SSL Self-Supervised Learning

ViT Vision Transformer

Chapter 1

Introduction

Human language and vision are among nature's most intricate information-bearing systems. Each offers a seemingly unbounded combinatorial space: language features sentences layered by syntax and discourse, and vision features scenes composed of objects in motion. Both have long served as benchmarks for artificial intelligence. Turing's "imitation game" (Turing, 1950) elevated conversational competence to a test of machine intelligence, and later decades added visual challenges such as scene understanding and autonomous navigation. Today, reliable algorithms for *understanding*, *reasoning*, and *generating* linguistic or visual data enable applications ranging from real-time translation and medical-image triage to open-domain question answering and self-driving cars.

From rules to data. Early systems depended on hand-crafted rules and feature templates. Symbolic grammars and finite-state transducers powered dialogue programs such as Winograd's *SHRDLU* (Winograd, 1972), while vision pipelines coupled *Scale-Invariant Feature Transform* or *Histogram of Oriented Gradients* with support-vector machines (Viola and Jones, 2001; Dalal and Triggs, 2005). Although effective in constrained settings, such methods faltered under the ambiguity and long-tailed variability of real-world data (Manning and Schutze, 1999).

The explosion of digital corpora in the 1990s prompted a decisive shift to *statistical learning*. In language technology, n-gram models and probabilistic parsers replaced rule lists; in vision, *bag-of-visual-words* classifiers learned object labels from code-

word histograms (Chen and Goodman, 1999; Sivic and Zisserman, 2003). These models moved feature discovery into the data, yet they still captured little beyond local co-occurrence patterns and often required task-specific heuristics.

Deep learning and self-supervision. A second revolution began when deep neural networks learned entire feature hierarchies end-to-end (Goodfellow, 2016). Convolutional Neural Network (CNN) models surpassed hand-engineered vision systems on ImageNet (Krizhevsky et al., 2012); sequence-to-sequence models with attention (Sutskever et al., 2014; Bahdanau et al., 2015) reshaped machine translation and speech recognition. Crucially, *self-supervised* objectives: predicting masked words (Mikolov et al., 2013; Devlin et al., 2019), next tokens (Radford et al., 2019), or contrastive image views (He et al., 2020), unlocked unlabelled corpora vastly larger than any curated dataset.

The Transformer era and foundation models. The Transformer architecture (Vaswani, 2017) replaced recurrence and convolution with fully parallel self-attention, enabling training on billions of tokens or image patches. Bidirectional masked-language models such as *BERT* (Devlin et al., 2019) advanced sentence understanding, whereas causal models like *GPT-2* and *GPT-3* (Radford et al., 2019; Brown et al., 2020) revealed *emergent* in-context learning, summarised by scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). Pure-Transformer backbones in vision: the Vision Transformer (*ViT*) (Dosovitskiy et al., 2021), Swin (Liu et al., 2022), and ViViT (Arnab et al., 2021), have likewise matched or surpassed CNN baselines. Collectively, these developments have produced *foundation models* (Bommasani et al., 2021) — single networks pre-trained on heterogeneous data and adapted to a wide range of machine learning tasks.

Yet increasing scale does not answer every scientific question. Training runs now consume gigawatt-hours (OpenAI et al., 2024), placing them beyond typical academic resources. More importantly, fundamental unknowns about their internal workings persist. This thesis aims to provide a deeper understanding of how these models *learn*, *represent*, and *generalise hierarchical structure*, an understanding that we believe is crucial for addressing these fundamental unknowns. Thus, we

pose three central research questions:

1. *Where do foundation models internalise hierarchical structures from data?*
2. *What is the relationship between this internalisation of hierarchy and model behaviours such as generalisation and memorisation?*
3. *How can insights into hierarchical reasoning guide the development of more unified, robust, and interpretable multimodal systems?*

In the subsequent section, we elaborate on the specific motivations for this thesis, which are shaped by the rapidly-moving trends and emerging topics of contemporary AI research.

1.1 Motivation

Over the lifecycle of this doctoral thesis, the rise of *foundation models* has reshaped Natural Language Processing (NLP). Autoregressive Transformers such as the GPT model family (Brown et al., 2020) and PaLM (Chowdhery et al., 2022) can write code, translate poetry, and draft policy briefs with minimal task-specific fine-tuning. Comparable trends appear in computer vision and speech. These achievements, however, rest on budgets affordable only by a few industrial laboratories*. Pursuing ever larger models is therefore beyond the scope of an individual researcher. Instead, this thesis focuses on the representational foundations that underpin the capabilities of such systems, particularly their learning and use of hierarchical structure, and examines this with the goal of improved interpretability, understanding, and safety.

Language is inherently hierarchical: morphemes form words, words combine into phrases, phrases into clauses, and clauses into discourse (Chomsky, 1957, 1965). Video shows a similar structure: optical flow aggregates into object trajectories, short interactions build into activities, and sequences of scenes compose narrative arcs (Arnab et al., 2021). Throughout this thesis, we define such hierarchical

*OpenAI estimates single-digit gigawatt-hours for GPT-4 pre-training (OpenAI et al., 2024)

structure as the *principle by which simple components are recursively composed into larger, meaningful units, which in turn serve as building blocks for higher levels of abstraction.* Although autoregressive Transformers achieve state-of-the-art performance, they are trained only to minimise next-token prediction loss. Remarkably, this simple objective appears to enable them to uncover aspects of syntax and semantics (Hewitt and Manning, 2019; Tenney et al., 2019); yet the depth, localisation, and robustness of these *internalised hierarchical representations* remain unclear. Seemingly benign paraphrases or controlled syntactic rearrangements can still trigger dramatic failures (McCoy et al., 2020; Bastings and Baroni, 2021). Understanding which layers encode which hierarchical abstractions, and how stable those encodings are, is essential if models are to be trusted. This directly motivates our first research question regarding *how and where* such structures are encoded.

This investigation into hierarchical reasoning is aligned with our own understanding of human cognition, as we naturally perceive the world not as a flat stream of data, but through nested layers of objects, events, and narratives. Therefore, by pursuing AI models with a similar capacity, we are not only building more robust and efficient systems but also creating a framework to better understand the principles of intelligence itself. Ultimately, hierarchical reasoning is a prerequisite for tackling the next frontier of AI challenges, such as long-term planning, complex problem-solving, and genuine creativity, which are all defined by their deeply nested structures.

Closely linked to hierarchical abstraction is the problem of *memorisation.* With billions of parameters, Transformers can store entire passages verbatim; extraction attacks have revealed private e-mails and unpublished fiction within supposedly general models (Carlini et al., 2021). Such leakage threatens copyright and confidentiality. However, the ability to remember and recall facts is crucial for model transferability and performance. While memorisation is often undesirable, it can be beneficial when models must retain canonical facts that support downstream accuracy such as information retrieval. The goal is to promote structured abstraction while constraining verbatim recall to appropriate contexts. The challenge, related

to our second research question, is to distinguish beneficial *generalisation of hierarchical structures* from harmful verbatim memorisation and recall of superficial n-grams or specific instances, particularly during model adaptation. This requires developing methods to detect and mitigate excessive memorisation without discarding valuable, generalisable, structurally learned knowledge (Zhang et al., 2021).

The same issues of hierarchical representation and the balance between generalisation and memorisation arise in vision. Pure-Transformer architectures such as ViT and Swin have replaced CNN backbones for images, and extensions like TimeSformer and ViViT rival CNNs for video (Arnab et al., 2021; Bertasius et al., 2021; Liu et al., 2022). Yet how these models fuse spatial and temporal cues, track identities through occlusion, and encode physical causality over long horizons (all inherently hierarchical tasks) remains underexplored. Adapting Transformer architectures and probing tools from NLP to investigate *spatiotemporal hierarchies* in video, as pursued in this work, promises deeper insight into whether current models genuinely learn multi-scale physical abstractions or merely interpolate.

Finally, real intelligence is multimodal. Humans read text, watch demonstrations, and manipulate objects in a unified cognitive space. Most current systems still rely on separate modality-specific encoders and decoders. Inspired by task reformulation in NLP (Raffel et al., 2020), our third research question leads us to explore whether a single, unified predictive objective can encourage the learning of *shared hierarchical representations* across text, images, audio, and video. Such unification would not only simplify architecture design but also provide a common framework for studying how hierarchical reasoning principles apply and transfer across different modalities, potentially leading to more robust and versatile AI.

This thesis addresses the overarching goal of understanding and leveraging hierarchical reasoning in modern Transformers by investigating: (1) the nature and location of learned hierarchical structures; (2) the interplay between hierarchical generalisation and memorisation; and (3) the application of hierarchical principles to build unified multimodal models. Through probing studies, controlled fine-tuning, and cross-modal reformulations, we aim to move the discussion beyond

benchmark scores towards a principled understanding of these complex AI systems.

1.2 Research Objectives and Contributions

To address the central research questions concerning the internalisation, generalisation, and application of hierarchical structures in Transformer models, this thesis makes six principal contributions:

- We develop probing techniques revealing *where and how Transformer language models encode syntactic and semantic hierarchy* in contextual embeddings.
- We construct ancestor classification tasks to *expose layer-wise hierarchical representation quality* across diverse architectures, comparing their internalisation of such structures.
- We investigate memorisation dynamics during fine-tuning, identifying *when hierarchical learning degrades into verbatim memorisation*, and introduce scalable defences to promote structural generalisation over verbatim recall.
- We design PSVIT, a video Transformer with explicit hierarchical priors (e.g., U-Net structure, tailored spatiotemporal attention), showing its enhanced capability for pixel-space modelling of complex physical dynamics.
- We extend hierarchical probing to video, using PSVIT to show how its spatiotemporal attention *encodes hierarchical physical dynamics and long-range dependencies*, advancing understanding of visual hierarchy internalisation.
- We propose a unified next-frame prediction framework, using an adapted PSVIT, enabling a single Transformer to process text, image, audio, and video, thereby fostering the *learning of shared hierarchical representations for unified multimodal systems.*

1.3 Publications

Research contributing to this thesis has been submitted for publication or published in journals and conference proceedings. The publications and corresponding chapters are listed below:

- Chapter 3 contains work partially presented in Slack, D. L., Hardey, M., & Al Moubayed, N. (2020). **On the Hierarchical Information in a Single Contextualised Word Representation**. Proceedings of the AAAI Conference on Artificial Intelligence, 34(10), 13917-13918.
- Chapter 4 contains work presented in Slack, D. L., & Al Moubayed, N., (2024) **Early Detection and Reduction of Memorisation for Domain Adaptation and Instruction Tuning**. Transactions of the Association for Computational Linguistics (TACL) (Under Revision Review).
- Chapter 5 contains work presented in Slack, D. L, Hudson, G. T., Winterbottom, T., & Moubayed, N. A. (2024). **Video Prediction of Dynamic Physical Simulations with Pixel-Space Spatiotemporal Transformers**. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) (Accepted pending minor revision).
- Chapter 6 contains work presented in Hudson, G. T., Slack, D. L, Winterbottom, T., Sterling, J., Xiao, C., Shentu, J., & Moubayed, N. A. (2024). **Everything is a Video: Unifying Modalities through Next-Frame Prediction**. IEEE/CVF International Conference on Computer Vision (ICCV) (Under Review).

1.4 Thesis Structure

This thesis sets out *to discover how hierarchical information is learned, encoded, and represented inside Transformer networks*, addressing the central questions of where these structures reside, how their learning relates to generalisation versus

memorisation, and how these insights can inform the development of unified multimodal models. We begin with causal language models, tracing how internal hierarchies emerge, how they sometimes surface as undesirable memorisation, and how similar structural priors in attention extend these findings to video and wider multimodal settings.

Chapter 2 lays the theoretical groundwork. We review probabilistic learning, deep-network optimisation, and self-supervision; formalise syntactic, semantic, and spatiotemporal hierarchies; and catalogue current methods for probing, memorisation analysis, and multimodal fusion. This chapter equips the reader with the concepts and tools employed in later chapters.

Chapter 3 directly addresses our first research question by probing the emergence of *hierarchical linguistic structure* inside contemporary Transformer language models. We introduce token-level "ancestor" tasks: sentiment and syntactic labels for a constituent word's parent, grand-parent, great-grand-parent, and sentence root, and apply them across BERT, GPT-2, XLNet, and Reformer in both base and large configurations. Layer-wise results show that bidirectional models concentrate hierarchy near the top of the stack, whereas causal or permutation-masked models disperse it more evenly. Fine-tuning magnifies mid-layer abstraction for XLNet and Reformer without erasing low-level cues, suggesting that architectural bias matters at least as much as raw parameter count. The probing framework established here supplies the analytical lens for investigating *where and how hierarchy is encoded*.

Chapter 4 turns that lens on our second research question, concerning the practical risk of memorisation during fine-tuning and its relationship to hierarchical learning. Using Pythia, Llama 2 and Llama 3, and Mistral language models covering a wide range of model sizes, we track verbatim leakage across domain-adaptation and instruction-tuning regimes. A simple n-gram "partial memorisation" score proves a reliable early warning signal, rising sharply one epoch before verbatim memorisation occurs. Exploiting this insight, we devise (i) an n-gram-threshold early-stopping rule and (ii) an n-gram-aware loss regulariser; together, they cut memorisation by up to 40% with only marginal cost to downstream accuracy, offer-

ing a scalable defence for promoting *generalised hierarchical learning* over verbatim memorisation.

Chapter 5 extends the hierarchical analysis from the linguistic domain to dynamic video modelling, further exploring our first and third research questions. We present PSViT, an end-to-end U-Net-style Transformer that predicts future video frames directly in pixel space. Evaluated on partial-differential-equation simulators and standard Moving-MNIST/BAIR benchmarks, PSViT outperforms latent-space baselines in long-horizon object tracking whilst remaining architecturally minimal. Probing of hidden states recovers latent simulation parameters and localises motion-specific attention heads, indicating that the model internalises *hierarchical physical abstractions* akin to those observed in language, and demonstrating how architectural priors can facilitate this.

Chapter 6 generalises the thesis theme to *multimodal reasoning*, directly addressing our third research question. Building on task-reformulation ideas from NLP, we recast text, image, audio, and video problems as a unified next-frame-prediction objective and train a single causal Transformer without modality-specific encoders. The resulting model performs competitively on captioning, visual question answering, and audio-to-text alignment, while sharing parameters and attention mechanisms across modalities, and offering a simpler end-to-end training regime. This task reformulation serves as a step towards universal models capable of learning *shared hierarchical representations across modalities*.

Chapter 7 summarises the contributions in light of the initial research questions, acknowledges limitations, and outlines future directions, ranging from adaptive attention mechanisms and causal-graph priors to privacy-preserving training regimes. We close by arguing that a hierarchy-centred analysis is essential for building safe, transparent, and physically-accurate foundation models.

Bibliography

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Bastings, J. and Baroni, M. (2021). Will you find these shortcuts? a test suite for shortcut learning in natural language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1005–1020.

Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 813–824.

Bommasani, R., Hudson, D. A., Adeli, E., and et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., and Kaplan, J. D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. (2021). Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium*, pages 2633–2650.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Chomsky, N. (1957). *Syntactic Structures*. Mouton & Co., The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). PaLM: Scaling language modeling with pathways. In *Proceedings of the International Conference on Machine Learning*.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Goodfellow, I. (2016). Deep learning.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Hoffmann, J., Borgeaud, S., Mensch, A., et al. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Kaplan, J., McCandlish, S., Henighan, T., et al. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Wang, Z., Li, Z., Zhang, X., Wang, Y., and Guo, B. (2022). Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12009–12019.

Manning, C. and Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

McCoy, R. T., Pavlick, E., and Linzen, T. (2020). Syntactic heuristics in natural language inference are derived from positional encoding. *arXiv preprint arXiv:2006.14635*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR) – Workshop Track*.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., and Akkaya, I. (2024). Gpt-4 technical report.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Technical Report. arXiv:1901.04537 (unpublished).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1477.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3104–3112.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, LIX(236):433–460.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518.

Winograd, T. (1972). Understanding natural language. *Cognitive psychology*, 3(1):1–191.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramer, F., and Carlini, N. (2021). Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*.

Chapter 2

Background

The research presented in Chapters 3 to 6 of this thesis investigates hierarchical reasoning within Transformer models across language, vision, and multimodal applications. This background chapter provides the essential theoretical, conceptual, and methodological foundations for these empirical studies. It is designed to support the thesis's central inquiry (outlined in Chapter 1) into how foundation models internalise hierarchical structures, the relationship between this internalisation and behaviours such as generalisation versus memorisation, and the application of hierarchical principles in developing unified multimodal systems. While this chapter will cover specific concepts, a basic familiarity with the principles of machine learning, Natural Language Processing (NLP), and Computer Vision (CV) is presumed. For a comprehensive introduction to these foundational fields, readers are directed to key texts such as Bishop (2006a); Jurafsky and Martin (2009); Szeliski (2010); Goodfellow et al. (2016).

This Chapter is organised into four sections:

1. Machine Learning and Probability Theory (§2.1);
2. Language Modelling (§2.2);
3. Transformer Architecture (§2.3);
4. Hierarchical Reasoning (§2.4).

2.1 Machine Learning and Probability Theory

Contemporary language and vision models (and more broadly, all systems explored in later Chapters) in large part derive their capabilities from probabilistic reasoning and data-driven optimisation. Probability theory allows us to quantify uncertainty inherent in linguistic or visual inputs, while machine learning supplies flexible approaches to make predictions or uncover underlying patterns (Bishop, 2006b; Murphy, 2012). This section briefly covers the fundamentals of key concepts that recur throughout the thesis, providing a backbone for understanding the Transformer-based models and the hierarchical reasoning principles investigated.

2.1.1 Foundations

Probabilistic reasoning is fundamental to contemporary machine learning, providing the mathematical scaffolding to manage uncertainty in real-world data (Bishop, 2006b). Whether handling ambiguity in language interpretation, variability in visual perception, or complexity in multimodal interactions, models must quantify their confidence in predictions through clearly defined probabilistic frameworks. This subsection introduces foundational concepts and notation for probability theory, essential for subsequent discussions of autoregressive models (Chapters 3 and 4), hierarchical probing, and later analyses of model behaviour.

Uncertainty and Probability Distributions. Natural data (whether text, images, or audio) rarely conform to deterministic patterns. Language, for example, is inherently ambiguous, context-sensitive, and richly variable (Shannon, 1951; Manning and Schütze, 1999; Chen and Goodman, 1999). Visual data also present considerable variability due to factors like viewpoint, lighting, and occlusion. Probability theory formalises such uncertainty by assigning likelihoods to outcomes rather than making categorical predictions. Instead of asserting definitively the identity of the next token or the content of an image region, a probabilistic model defines a

distribution over potential outcomes. Such distributions are fundamental to generative tasks (e.g., language modelling, Section 2.2, or video generation, Chapter 5) and scenarios requiring explicit confidence estimates (e.g., regression with uncertainty quantification (Lakshminarayanan et al., 2017)). For hierarchical reasoning, understanding how uncertainty propagates across nested structures or latent variables is crucial for developing robust and interpretable models, a key aim of this thesis (Pearl, 2009; Murphy, 2012).

Random Variables: Discrete vs. Continuous. A *random variable* X maps outcomes in a sample space to numerical values. In NLP, word tokens are often modelled by discrete random variables (taking values in a finite vocabulary $\mathcal{V}$). Continuous random variables arise for real-valued embeddings, physical parameters (as explored in Chapter 5 for simulation-based tasks), or pixel intensities in images. Formally:

- **Discrete Random Variables:** Characterised by a probability mass function $P(X = x)$, with $\sum_{x \in \mathcal{V}} P(X = x) = 1$.
- **Continuous Random Variables:** Characterised by a probability density function $p(x)$, which integrates to 1 over the domain. For real-valued vectors, e.g., embeddings, $p(\mathbf{x})$ might assume parametric forms like a Gaussian (Bishop, 2006a).

Joint, Marginal, and Conditional Probability. Data in language and vision often exhibit dependencies: the probability of one word depends on preceding words, and the interpretation of an image region can depend on its surroundings. A *joint distribution* $P(X_1, \ldots, X_T)$ describes the likelihood of entire sequences or sets of variables. We often focus on *conditional* probabilities, such as $P(X_t \mid X_1, \ldots, X_{t-1})$ in autoregressive language models (Bengio et al., 2003; Radford et al., 2019), or $P(\text{object class} \mid \text{image features})$ in image classification (Krizhevsky et al., 2012). *Marginal* distributions (for instance $P(X_1, X_2)$) emerge by summing or

integrating out the other variables. This is useful in tasks such as partial likelihood estimation or certain hierarchical Bayesian setups.

Chain Rule of Probability. The chain rule breaks down a joint distribution into a product of simpler conditionals:

$$P(X_1, \ldots, X_T) = \prod_{t=1}^{T} P(X_t \mid X_1, \ldots, X_{t-1}). \tag{2.1}$$

This factorisation is the cornerstone of autoregressive language modelling (Section 2.2), where each token's probability is conditioned on all prior tokens (Bengio et al., 2003; Brown et al., 2020). Similarly, it can be applied to model sequences of frames in video prediction (Chapter 5). We often train models to maximise $\sum_{t=1}^{T} \log P(x_t \mid x_{1:t-1})$, a strategy central to large-scale language models.

Maximum Likelihood Estimation. A natural follow-on to the chain rule is the practical approach for estimating these conditional probabilities. The most common method employed is *Maximum Likelihood Estimation (MLE)* (Bishop, 2006b). Given a set of observed training sequences $\mathcal{D} = \{x_{1:T}^{(n)}\}_{n=1}^{N}$, MLE seeks parameters θ that maximise the likelihood of observing the training data under the model P_θ:

$$\theta_{\text{MLE}} = \arg\max_{\theta} \prod_{n=1}^{N} P_\theta(x_{1:T}^{(n)}). \tag{2.2}$$

Typically, we optimise the logarithm of the likelihood (log-likelihood), which is computationally more convenient and numerically stable for gradient-based methods:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \sum_{n=1}^{N} \sum_{t=1}^{T} \log P_\theta(x_t^{(n)} \mid x_{1:t-1}^{(n)}). \tag{2.3}$$

MLE thus provides the foundational principle underpinning the training of nearly all language and generative vision models discussed in this thesis: from autoregressive Transformers (Chapters 3 and 4) to spatiotemporal video prediction models (Chapter 5). Understanding the properties of MLE estimators, such as consistency and potential biases, is crucial when diagnosing model behaviour, particularly the distinction between memorisation versus generalisation of learned structures (Hastie et al., 2009).

Notation Conventions. Throughout this thesis, we adopt consistent notation:

- $X_{1:T}$ denotes sequences of length T.
- x_t and $\mathbf{x}$ denote scalar and vector variables, respectively.
- θ represents model parameters.
- $P_\theta(x)$ indicates probability distributions parameterised by θ.

2.1.2 Machine Learning in Natural Language Processing

Natural Language Processing (NLP) is a field of artificial intelligence and linguistics concerned with the interactions between computers and human language; in particular, how to program computers to process and analyse large amounts of natural language data (Jurafsky and Martin, 2009). Machine learning has become the dominant paradigm for tackling complex NLP tasks, moving beyond earlier rule-based and statistical approaches (Manning and Schütze, 1999; Goldberg, 2017). Tasks such as text classification (e.g., sentiment analysis, topic categorisation), sequence labelling (e.g., part-of-speech tagging, named entity recognition), machine translation, and question answering are commonly framed as supervised or semi-supervised learning problems (Goldberg, 2017; Deng and Liu, 2018).

Initially, machine learning in NLP often relied on hand-crafted features extracted from text (e.g., bag-of-words, TF-IDF representations) fed into algorithms like Naive Bayes, Support Vector Machines (SVMs), or Logistic Regression (Manning and Schütze, 1999). While effective for certain tasks, these methods struggled with the inherent ambiguity, richness, and long-range dependencies present in human language. The advent of deep learning, particularly Recurrent Neural Networks (RNNs) (Mikolov et al., 2010; Sherstinsky, 2020) and later Transformers (Section 2.3), revolutionised the field by enabling models to learn relevant features directly from raw text data, leading to significant improvements in performance (Goodfellow et al., 2016; Goldberg, 2017).

Evaluation of NLP models depends on the specific task. For classification tasks, such as the probing tasks used in Chapter 3 to assess hierarchical information,

metrics like accuracy, precision, recall, and F1-score are standard (Manning et al., 2008; Deng and Liu, 2018). For language modelling itself, perplexity is a common intrinsic measure (discussed further in Section 2.2). The challenges in NLP, including capturing nuanced semantic meaning, understanding context, and generating coherent and grammatically correct text, continue to drive research into more sophisticated models and learning techniques, forming the backdrop for the investigations into Transformer capabilities in this thesis.

2.1.3 Machine Learning in Computer Vision

Computer Vision (CV) aims to enable machines to interpret and understand visual information from the world, such as images and videos. Similar to NLP, machine learning, especially deep learning, has become the cornerstone of modern CV, supplanting many classical techniques that relied on manually designed filters and features (Bishop, 2006b; Goodfellow et al., 2016). Tasks central to CV include image classification (assigning a label to an entire image), object detection (locating and classifying multiple objects within an image), image segmentation (partitioning an image into meaningful regions), and video analysis (understanding motion, activities, and events over time), which is particularly relevant to Chapters 5 and 6.

Classical CV often involved extracting local features like Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) or Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005), followed by machine learning classifiers. However, CNN models brought a paradigm shift by learning hierarchical feature representations directly from pixel data, from simple edges and textures in early layers to more complex object parts and entire objects in deeper layers (LeCun et al., 1998; Krizhevsky et al., 2012; Goodfellow et al., 2016). This ability to learn features end-to-end has led to breakthroughs across numerous vision benchmarks. More recently, Transformer architectures, initially designed for NLP, have also demonstrated remarkable success in vision tasks (Dosovitskiy et al., 2021), as discussed in Section 2.3 and explored in Chapters 5 and 6.

Evaluation metrics in CV are task-dependent. Image classification often uses accuracy or top-k accuracy. Object detection is commonly evaluated using mean Average Precision (mAP), a metric popularised through challenges like the PASCAL VOC challenge (Everingham et al., 2010). For video analysis and prediction tasks like those in Chapter 5, metrics can include pixel-level measures like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) (Wang et al., 2004), or task-specific metrics like object tracking accuracy. The challenges in CV are vast, including handling variations in scale, viewpoint, illumination, deformation, and occlusion, as well as understanding complex spatiotemporal dynamics in videos. These challenges motivate the development of robust hierarchical models capable of learning invariant and discriminative representations.

2.1.4 Model Capacity, Overfitting, and Generalisation

Understanding the principles of probabilistic modelling and machine learning also requires acknowledging the practical challenges encountered when fitting flexible models to finite datasets. These considerations are particularly salient for the high-capacity Transformer architectures central to this thesis. Key concepts include model capacity, the risk of overfitting, and the ultimate goal of achieving good generalisation. Each concept critically influences how effectively and reliably models can encode and generalise hierarchical structures across language, vision, and multimodal domains.

Model Capacity and Complexity. Model *capacity* refers to the expressive power of a model class, typically related to the number of parameters or degrees of freedom it possesses (Goodfellow et al., 2016). Higher-capacity models, such as large-scale Transformers (e.g., GPT-3), can represent increasingly sophisticated patterns and hierarchical structures, capturing subtle dependencies within complex data (Kaplan et al., 2020; Chowdhery et al., 2022). However, this flexibility comes at a cost: without sufficient data or appropriate regularisation, models with high capacity are more prone to fitting noise or idiosyncratic patterns specific to

the training set, potentially hindering their performance on unseen data (Bishop, 2006b; Zhang et al., 2017).

Overfitting and Generalisation. *Overfitting* occurs when a model learns the training data too well, including its noise and specific quirks, to the detriment of its performance on new, unseen data (Bishop, 2006b; Murphy, 2012). This is typically observed as a low error rate on the training data but a significantly higher error rate on a separate validation or test dataset. Formally, overfitting manifests as a divergence between training and test performance metrics, often measured by differences in likelihood (perplexity), accuracy, or task-specific evaluation scores. Classical statistical learning theory often describes this through the bias-variance trade-off: overly complex models (high capacity) may have low bias as they can fit the training data well, but they might suffer from high variance, meaning they are very sensitive to the specific training data and thus generalise poorly to new data (Hastie et al., 2009). Conversely, overly simple models (low capacity) may exhibit high bias (failing to capture the underlying structure) but low variance. Achieving good generalisation by balancing bias and variance is a central goal in machine learning, often addressed through techniques such as regularisation, early stopping, or careful model selection (Goodfellow et al., 2016).

2.2 Language Modelling

Language modelling has emerged as a foundational task within NLP, profoundly shaping research directions and methodologies over recent decades (Goldberg, 2017; Jurafsky and Martin, 2023). Originally framed as a probabilistic task focused on predicting the next word in a sentence (Shannon, 1951), it has expanded into a versatile paradigm encompassing text generation, representation learning, and transfer learning (Radford et al., 2019; Brown et al., 2020; Bommasani et al., 2021). Advances driven by language modelling have yielded powerful tools for diverse applications such as machine translation (Sutskever et al., 2014; Bahdanau et al., 2015), dialogue systems (Serban et al., 2016), and information retrieval (Nogueira

and Cho, 2020). Crucially, breakthroughs in language modelling underpin the rise of modern neural NLP systems, including the Transformer-based models investigated in this thesis. Understanding how these models learn linguistic phenomena, particularly hierarchical structures, is essential for their effective and safe application, aligning with the core goals of this thesis (Ruder et al., 2019; Rogers et al., 2020).

Language modelling is the task of learning a probability distribution over sequences of tokens in a language. Formally, given a sequence $x_1, x_2, \ldots, x_T$ of tokens, a language model assigns a probability $P(x_1, x_2, \ldots, x_T)$ to the sequence. Using the chain rule of probability introduced previously (Equation 2.1), this joint distribution factorises into conditional probabilities:

$$P(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} P(x_t \mid x_1, \ldots, x_{t-1}) \ , \tag{2.4}$$

where $P(x_1 \mid x_{<1}) \equiv P(x_1)$ is the marginal probability of the initial token. Historically, early statistical language models relied on Markov assumptions to simplify this estimation by conditioning each token only on the preceding $n-1$ tokens (forming an n-gram model). Despite their simplicity, these models suffered from data sparsity and limited contextualisation, restricting their ability to capture long-range dependencies inherent in language (Manning and Schütze, 1999; Chen and Goodman, 1999). Neural language models address these limitations by learning distributed representations (embeddings) of tokens, enabling generalisation beyond observed n-grams and capturing more complex linguistic patterns (Bengio et al., 2003; Mikolov et al., 2010). By using neural networks to estimate conditional probabilities, such models can learn $P(x_t \mid x_{<t})$ from data without explicit Markov constraints, forming the basis for the powerful sequential models discussed in subsequent chapters.

2.2.1 Causal Language Modelling

Causal Language Modelling (CLM), also known as autoregressive language modelling, explicitly models the sequential generation of tokens (Goldberg, 2017; Jurafsky

and Martin, 2023). Given a sequence, the model predicts each token based only on the preceding tokens, strictly adhering to the temporal or sequential order. Formally, the training objective is to maximise the log-likelihood of the observed sequences under the model's parameters θ:

$$\mathcal{L}_{\text{CLM}}(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log P_\theta(x_t^{(i)} \mid x_1^{(i)}, \ldots, x_{t-1}^{(i)}), \tag{2.5}$$

for a corpus of N sequences, where $x_t^{(i)}$ is the t-th token of the i-th sequence. This approach directly aligns with the chain rule of probability, making it naturally suited for tasks involving sequential generation, such as text completion, dialogue generation (Radford et al., 2019), and the predictive coding mechanisms hypothesised in human language processing (Shannon, 1951). Prominent examples include Transformer-based autoregressive models like GPT-2 and GPT-3 (Radford et al., 2019; Brown et al., 2020), which leverage masked self-attention (detailed in Section 2.3) to preserve causal ordering during training and inference. Understanding how these models, despite their simple next-token prediction objective, manage to learn and represent complex hierarchical structures (as investigated in Chapter 3) is central to one of the core research questions of this thesis.

2.2.2 Masked Language Modelling

Masked Language Modelling (MLM) (often referred to as autoencoding language models), introduced prominently by BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), diverges from the strictly causal formulation. Instead of predicting the next token in a sequence, MLM involves predicting randomly masked tokens using their surrounding unmasked context, thus leveraging both preceding (left) and succeeding (right) tokens. Formally, given a sequence $X = (x_1, \ldots, x_T)$ where a subset of tokens X_M is masked, and $X_{\setminus M}$ are the observed tokens, MLM aims to reconstruct the original tokens in X_M:

$$\mathcal{L}_{\text{MLM}}(\theta) = \sum_{x_m \in X_M} \log P_\theta(x_m \mid X_{\setminus M}). \tag{2.6}$$

Typically, around 15% of the input tokens are randomly selected for masking during pre-training, with specific strategies like replacing the token with '[MASK]',

a random token, or keeping it unchanged (Devlin et al., 2019). The model then learns by minimising the prediction error at these masked positions. This bidirectional approach encourages the learning of rich contextual embeddings that capture deep linguistic relationships. Unlike causal modelling, MLM models do not directly define a sequential generative process for entire sequences from scratch but excel at learning general-purpose linguistic representations suitable for a wide array of downstream tasks, such as classification, question-answering, and textual entailment, often via fine-tuning (Devlin et al., 2019; Rogers et al., 2020). Chapter 3 explores how the bidirectionality of MLM influences the encoding of hierarchical information compared to causal models.

2.2.3 Self-Supervised Learning

Both causal and masked language modelling are prime examples of Self-Supervised Learning (SSL), a learning paradigm that has become a cornerstone of modern AI. In SSL, models learn to represent data by solving "pretext" tasks where the supervision signal is generated automatically from the input data itself, rather than relying on costly and often scarce human-provided labels (Bengio et al., 2013; Goodfellow et al., 2016). For language modelling, the pretext task involves predicting parts of the input text (e.g., a masked word in MLM or the next word in CLM) based on other parts of the same text (Mikolov et al., 2013; Devlin et al., 2019). This intrinsic supervision allows models to learn from the abundant unlabelled text available on an unprecedented scale.

The fundamental strength of SSL lies in its ability to unlock the potential of massive unlabelled datasets. By defining pretext tasks, researchers have enabled models to learn rich, hierarchical, and transferable representations that capture underlying data structures, which is central to understanding *how hierarchical structures are internalised* by these models. For instance, in NLP, models like BERT (Devlin et al., 2019) learn contextual word embeddings by predicting masked words, while models like GPT (Radford et al., 2019) learn by predicting the next word. These representations have proven highly effective for a wide range of downstream tasks.

The success of these approaches has drastically reduced the dependency on large labelled datasets for specific tasks, shifting the paradigm towards pre-training on general domain data and then fine-tuning on smaller, task-specific datasets (Howard and Ruder, 2018; Ruder et al., 2019).

While language modelling provides prominent examples, the SSL paradigm extends across various modalities and has resulted in numerous innovative pretraining tasks. In computer vision, influential SSL approaches include contrastive learning methods like SimCLR (Chen et al., 2020) and MoCo (He et al., 2020), which learn representations by discriminating between similar and dissimilar augmented views of images. Masked image modelling, analogous to MLM in text, has also proven highly effective, with models like Masked Autoencoders (MAE) (He et al., 2022) reconstructing randomly masked patches of an image. Other approaches include Bootstrap Your Own Latent (BYOL) (Grill et al., 2020) and Contrastive Predictive Coding (CPC) (van den Oord et al., 2018), which explores predictability in latent spaces. Similarly, in speech processing, models like HuBERT learn representations by predicting masked acoustic units (Hsu et al., 2021). This broad applicability underscores the power of self-supervision to extract meaningful patterns irrespective of the data modality.

The representations learned through SSL often implicitly capture complex semantic and syntactic properties in language (Tenney et al., 2019; Hewitt and Manning, 2019), or compositional visual features (Caron et al., 2021), even though these are not explicitly supervised. This emergent understanding is crucial for the generalisation capabilities observed in large models. Indeed, SSL is the driving force behind the development of *foundation models* – large models pre-trained on broad data that can be adapted to a wide range of downstream applications, often exhibiting impressive few-shot or zero-shot learning capabilities (Brown et al., 2020; Bommasani et al., 2021). The self-supervised framework not only circumvents the data-labelling bottleneck but also encourages the development of models that learn more robust and broadly useful internal representations, significantly mitigating overfitting when applied to specific tasks with limited labelled data. Con-

sequently, SSL has become integral to modern AI pipelines, facilitating efficient transfer learning and domain adaptation, properties that are particularly relevant to the adaptation studies conducted in Chapter 4 of this thesis. The very ability of Transformers to learn hierarchical structures is largely enabled through these powerful SSL objectives during their pre-training phase.

2.2.4 Text-to-Text and In-Context Learning

The advancements in SSL and the scaling of Transformer models have given rise to a powerful and flexible approach for tackling diverse NLP tasks: the text-to-text paradigm. Popularised by models like T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020), this framework unifies various tasks by casting them all as problems of generating a textual output given a textual input. For example, translation, summarisation, question answering, and even classification can be reframed such that the model reads a prompt describing the task (and potentially including input data) and generates the desired output as a string of text. This approach simplifies the need for task-specific architectures and loss functions, promoting a more general method of transfer learning.

Perhaps one of the most striking emergent abilities of large-scale causal language models, such as GPT-3 (Brown et al., 2020), is *in-context learning* or *few-shot prompting*. Unlike traditional fine-tuning, where model weights are updated on a task-specific dataset, in-context learning allows these models to perform new tasks simply by conditioning on a textual prompt that includes a natural language description of the task and a few examples (or "shots") demonstrating the input-output pattern. The model then generates a completion that, for a well-crafted prompt, often correctly solves the task for a new input provided in the prompt, all without any direct modification of its parameters (Brown et al., 2020). This capability significantly lowers the barrier for adapting these powerful models to novel applications and has driven extensive research into prompt engineering and understanding the mechanisms behind such rapid, example-driven adaptation. More advanced prompting techniques, such as chain-of-thought prompting, which en-

courage the model to generate intermediate reasoning steps, can further enhance performance on complex reasoning tasks (Wei et al., 2022; Ouyang et al., 2022). The capacity for in-context learning highlights a sophisticated level of pattern recognition and analogical reasoning learned during pre-training, and its relationship with how models represent and utilise knowledge, including hierarchical structures, is an active area of investigation.

2.2.5 Pretrained Language Models

While earlier approaches to learning word representations, such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), provided static, non-contextual embeddings, the field witnessed a significant shift with the advent of Pretrained Language Models (PLMs) capable of generating *contextual word representations*. These models are typically deep neural networks trained on vast amounts of text data using self-supervised language modelling objectives, as discussed previously.

The feasibility of pretraining such large models increased with advancements in computational resources. Early explorations by Dai and Le (2015) demonstrated pretraining language models on large in-domain document collections, and subsequent work began to apply pretrained LLMs to specific downstream tasks like sequence labelling (Peters et al., 2017) and machine translation (Ramachandran et al., 2017). A pivotal moment came with Peters et al. (2018) who demonstrated that representations from a bidirectionally trained Long Short-Term Memory (LSTM) language model (ELMo) could significantly improve performance across a wide array of NLP tasks. These ELMo representations were 'deep' in that they were a function of all internal layers of the language model and contextual because the representation for each word depended on the entire input sentence.

Following this, Radford et al. (2018) scaled this approach by training a deeper Transformer-based language model, the Generative Pretrained Transformer (GPT), on more extensive data, showcasing strong performance on several benchmarks through fine-tuning. A further major development was the introduction of the MLM objective by Devlin et al. (2019) in BERT, which allowed for deep bid-

irectional pretraining. These PLMs have demonstrated remarkable capabilities, achieving substantial improvements on many established NLP tasks (Ruder et al., 2019; Qiu et al., 2020). Their success stems from their ability to learn rich linguistic knowledge from large corpora, which can then be transferred to downstream tasks. This transfer learning capability has been particularly impactful for domain adaptation and for enabling few-shot or even zero-shot learning scenarios (Howard and Ruder, 2018; Radford et al., 2018; Brown et al., 2020).

2.2.6 Fine-tuning and Feature Extraction

Once a language model has been pretrained on a large general-domain corpus, it needs to be adapted to specific downstream tasks. The process of leveraging pretrained knowledge for a new task is a form of transfer learning. There are two primary strategies for adapting PLMs: fine-tuning and feature extraction (Howard and Ruder, 2018; Peters et al., 2018; Ruder et al., 2019).

Feature Extraction involves using the pretrained model as a fixed feature extractor. The PLM processes the input text, and its hidden states (often from one or more layers) are taken as contextual embeddings (Peters et al., 2018). These embeddings are then used as input features for a separate, often simpler, task-specific model (e.g., a linear classifier or a shallow neural network) which is trained from scratch on the target task data. The weights of the pretrained model itself are 'frozen' and not updated during this process. This approach is beneficial as existing task-specific architectures can be reused, and if features are needed repeatedly, they only need to be extracted once, which can be computationally cheaper for multiple training iterations or tasks (Salman et al., 2023).

Fine-tuning, in contrast, involves unfreezing some or all of the pretrained model's parameters and continuing to train them on the target task data, typically with a task-specific output layer appended to the PLM (Howard and Ruder, 2018; Devlin et al., 2019). The pretrained parameters serve as a sophisticated initialisation, and gradients from the task-specific loss are propagated back through the PLM, allowing its representations to adapt to the nuances of the target task. Fine-tuning is

often convenient as it allows a single general-purpose model to be adapted with minimal modifications and has generally been found to yield better performance than using static (frozen) embeddings, especially when the pretrained model architecture is well-suited to the task (Howard and Ruder, 2018; Devlin et al., 2019) (though early work like Kim (2014) already showed benefits of fine-tuning word vectors). Howard and Ruder (2018) proposed ULMFiT, which introduced several effective techniques for fine-tuning language models, such as discriminative fine-tuning (different learning rates for different layers) and gradual unfreezing, significantly boosting the effectiveness of transfer learning in NLP.

However, fine-tuning is not without its challenges. For instance, if the target task training set is very small, fine-tuning all parameters can lead to overfitting (Howard and Ruder, 2018). Moreover, for word embeddings, only the parameters of words seen during fine-tuning are updated, potentially making embeddings of out-of-vocabulary or rare words 'stale' if they are not encountered frequently in the fine-tuning data or if subword units are not sufficiently representative in the new domain (Balde et al., 2024). The choice between feature extraction and fine-tuning can depend on factors like the similarity between the pretraining and target tasks, the size of the target dataset, and computational constraints. Some studies suggest that fine-tuning performs better when source and target tasks are similar, while feature extraction may be more robust when they are distant (Peters et al., 2019). Understanding the dynamics of fine-tuning, including the risk of memorising task-specific data (a key aspect of the second research question in this thesis) is explored in Chapter 4.

2.2.7 Evaluating Language Models

Evaluating language models quantitatively is essential for assessing their quality and comparing different architectures or training strategies. The primary intrinsic evaluation metric for language models, particularly causal ones, is *perplexity*. Perplexity measures how well a probability model predicts a sample. For a test set sequence $X_{1:T} = (x_1, x_2, \ldots, x_T)$, the perplexity is defined as the exponential of

the average negative log-likelihood per token:

$$\text{Perplexity}(X_{1:T}) = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log P_\theta(x_t \mid x_{<t})\right), \tag{2.7}$$

where $\log P_\theta(x_t \mid x_{<t})$ is the log-likelihood of the t-th token conditioned on the preceding tokens. We can think of this as measuring a model's capacity to make uniform predictions across all tokens in a training corpus.

A lower perplexity score indicates that the model is better at predicting the sample sequence, meaning the distribution learned by the model is closer to the empirical distribution of the language (Chen and Goodman, 1999). While perplexity is a useful measure of a model's fluency and ability to capture statistical regularities, it may not always directly correlate with performance on downstream NLP tasks (Gadre et al., 2024) or capture deeper aspects of linguistic understanding, such as syntactic correctness or semantic coherence, particularly in complex, long-context scenarios (Chi et al., 2024) or when probing the generalisation of learned hierarchical structures versus surface statistics. Other evaluation methods often involve assessing performance on specific downstream tasks (extrinsic evaluation), such as text classification (using accuracy, F1-score) or question answering, often through comprehensive benchmarks (Wang et al., 2018; Liang et al., 2022). However, these classical metrics alone may obscure deeper issues such as overconfidence (Guo et al., 2017), the risk of memorising training data (explored in Chapter 4), and vulnerability to subtle distribution shifts (Yang et al., 2023).

2.2.8 Relevance to Transformer Architectures

The Transformer architecture, which will be detailed in Section 2.3, has become the de facto standard for the state-of-the-art PLMs discussed above, largely supplanting previous recurrent architectures like LSTMs for these large-scale endeavours (Vaswani et al., 2017). Its parallelisable self-attention mechanism is highly effective at capturing dependencies across long sequences of text and scales efficiently with computational resources and vast datasets, making it particularly well-suited for the objectives of both CLM and MLM training.

Specifically, Transformer models implement the CLM objective (e.g., in GPT-style autoregressive architectures) by employing a specific type of attention mask (often an upper-triangular mask) that restricts each token from attending to subsequent tokens, thereby strictly preserving the autoregressive, left-to-right generation process (Vaswani et al., 2017; Radford et al., 2019). This is crucial for tasks like text generation and forms the basis for the emergent in-context learning abilities observed in models like GPT-3 (Brown et al., 2020). Conversely, for the MLM learning objective (e.g., in BERT-style architectures), Transformers are typically configured to allow tokens to attend to all other tokens in the input sequence bidirectionally (or non-directionally) when constructing representations to predict the masked tokens (Devlin et al., 2019). This rich bidirectional context is a key reason for BERT's strong performance on various natural language understanding tasks. The inherent flexibility of the Transformer in accommodating these different pretraining objectives, coupled with its scalability, has cemented its role as the foundational architecture for exploring the capabilities of PLMs. This includes investigating the emergence and interpretability of hierarchical linguistic structures (a central theme of Chapter 3) and how these structures relate to model generalisation and memorisation (Chapter 4), key questions posed in this thesis. The capacity of these Transformer-based PLMs to be adapted via fine-tuning or used in few-shot settings further underscores their impact on the field.

2.2.9 Summary

In summary, language modelling has evolved from a probabilistic task of sequence prediction into a cornerstone of modern NLP, driving significant advancements in how machines understand and generate human language. Key developments include the shift from n-gram models to neural approaches, the refinement of causal (autoregressive) and masked (autoencoding) objectives, and the transformative impact of self-supervised learning. This has culminated in the era of large PLMs which produce rich contextual representations and can be adapted effectively to a multitude of downstream tasks through fine-tuning or leveraged directly via in-

context learning within the text-to-text paradigm. These PLMs, predominantly built upon the Transformer architecture, have not only pushed the boundaries of performance but also opened new avenues for research into their internal workings, generalisation properties, and the nature of the linguistic knowledge they acquire, particularly regarding hierarchical structures.

The principles and models discussed in this section, from the foundational chain rule of probability to sophisticated PLMs and their adaptation methods, provide the essential background for the empirical investigations undertaken in this thesis. We build on this with investigations into hierarchical representation learning within Transformers (Chapter 3), the challenges of mitigating memorisation during fine-tuning (Chapter 4), and the extension of these ideas to vision and multimodal learning (Chapters 5 and 6), all of which contribute to addressing the overarching research questions of this thesis. We will next look into the specifics of the Transformer architecture itself.

2.3 Transformer Architecture

The Transformer architecture, introduced by Vaswani et al. (2017), represents a pivotal shift in sequence modelling, particularly within NLP, by moving away from recurrent and convolutional mechanisms towards a design based entirely on attention. This section provides a detailed exposition of the Transformer model, its core components, and its operational principles. While we focus on the architecture itself, a comprehensive understanding of neural networks, their training via back-propagation, and optimisation algorithms is presumed. For foundational knowledge on these broader deep learning topics, readers are referred to standard texts such as Bishop (2006b) and Goodfellow et al. (2016). Our aim here is to understand how the Transformer's specific design choices, from multi-head self-attention to positional encodings, enable the learning of complex dependencies and hierarchical features in sequential data, features that are central to this thesis's investigation into how models internalise, generalise, and apply hierarchical structure. Under-

standing these architectural details is crucial, as our later analyses in Chapters 3, 5, and 6 rely on interpreting the structures and representations learned within these networks.

2.3.1 Model Overview

The original Transformer architecture (Vaswani et al., 2017), designed for machine translation, comprises two main blocks (Figure 2.1):

- **Encoder Block:** A stack of N identical layers, each processing input embeddings to produce context-aware representations.
- **Decoder Block:** Another stack of N layers that receives both the encoder outputs and its own previous layer outputs to generate predictions autoregressively.

For unidirectional language modelling tasks, many influential implementations (e.g., GPT-2, GPT-3) utilise only a *decoder-like* stack with causal masking applied to the self-attention mechanism (Radford et al., 2019; Brown et al., 2020). Conversely, models like BERT employ an *encoder-only* stack for tasks requiring bidirectional context (Devlin et al., 2019). Despite these variations, the fundamental components discussed below are common across most Transformer-based systems.

2.3.2 Input and Positional Embeddings

Let $X = (x_1, x_2, \ldots, x_n)$ be an input sequence of n tokens. Each token x_i is first mapped to a d-dimensional embedding vector $\mathbf{e}_i \in \mathbb{R}^d$ using a learned embedding matrix. Since the Transformer architecture does not inherently process sequential order due to the parallel nature of its attention mechanisms (unlike RNNs), explicit information about the position of tokens in the sequence must be injected. This is achieved through *positional encodings* (**PE**). The original Transformer paper proposed using sinusoidal functions (Vaswani et al., 2017):

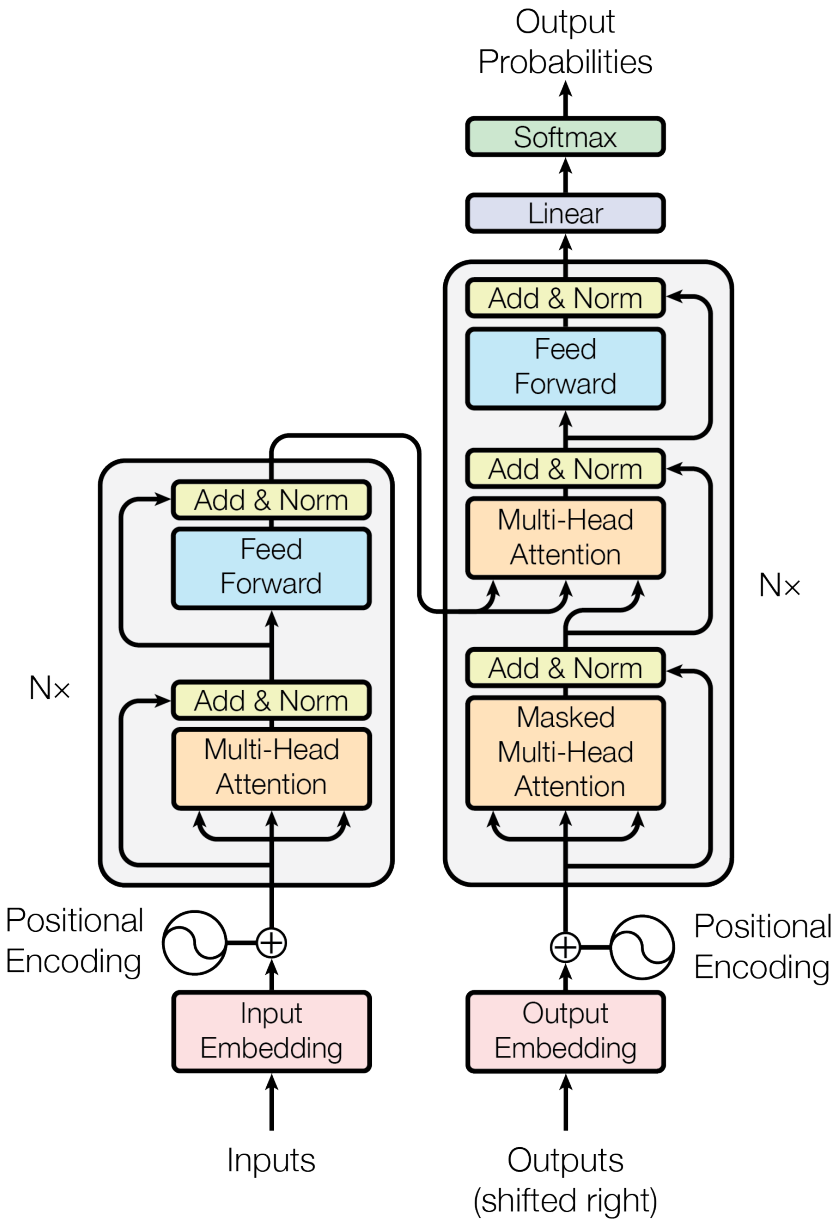


Figure 2.1: The Transformer architecture, illustrating the encoder (left) and decoder (right) stacks. Adapted from Vaswani et al. (2017).

$$\mathbf{PE}(pos, 2k) = \sin\left(\frac{pos}{10000^{2k/d_{\text{model}}}}\right), \tag{2.8}$$

$$\mathbf{PE}(pos, 2k+1) = \cos\left(\frac{pos}{10000^{2k/d_{\text{model}}}}\right), \tag{2.9}$$

where pos is the position index in the sequence ($0 \leq pos < n$), k is an index into the dimensions of the embedding ($0 \leq k < d_{\text{model}}/2$), and d_{model} is the dimensionality of the embeddings (denoted as d elsewhere in this text for simplicity). These positional

encodings $\mathbf{p}_{pos} \in \mathbb{R}^{d_{\text{model}}}$ are then added element-wise to the corresponding token embeddings:

$$\mathbf{z}_{pos}^{(0)} = \mathbf{e}_{pos} + \mathbf{p}_{pos} \tag{2.10}$$

The resulting vectors $\mathbf{z}_{pos}^{(0)}$ form the input to the first Transformer layer. The sinusoidal scheme was chosen because it could allow the model to learn relative positions and potentially generalise to sequence lengths not seen during training (Vaswani et al., 2017), although robust length generalisation remains an active research area (Press et al., 2022). Other forms of positional encodings, including learned absolute or relative positional embeddings, have also been explored and are used in various Transformer variants (Shaw et al., 2018; Devlin et al., 2019; Raffel et al., 2020). The choice of positional encoding can be particularly important for tasks requiring precise spatial or temporal understanding, as explored in Chapter 5, and may influence how hierarchical spatial or temporal relationships are encoded.

Positional encodings provide the model with a sense of order or position for each token. Since every token in a Transformer layer is processed in parallel through self-attention, without these encodings, the model would be permutation-invariant, treating the input as an unordered set of tokens. The positional signals allow the model to learn dependencies based on relative or absolute positions.

2.3.3 Self-Attention

At the heart of the Transformer is the *self-attention* mechanism (Vaswani et al., 2017). Self-attention allows the model to weigh the importance of different tokens in a sequence when computing the representation for each token. Instead of relying on fixed-window convolutions or sequential processing, self-attention enables each token to interact directly with all other tokens in its receptive field (which can be the entire sequence, or a masked portion thereof).

For a given layer l, let the input be a sequence of n vectors represented as a matrix $\mathbf{Z}^{(l-1)} \in \mathbb{R}^{n \times d}$, where each row $\mathbf{z}_i^{(l-1)}$ is the d-dimensional representation of token i from the previous layer.

Key, Query, and Value Projections. From the input representations $\mathbf{Z}^{(l-1)}$, three matrices are generated by multiplying with learned weight matrices: a *Query* matrix $\mathbf{Q}$, a *Key* matrix $\mathbf{K}$, and a *Value* matrix $\mathbf{V}$.

$$\mathbf{Q} = \mathbf{Z}^{(l-1)}\mathbf{W}_Q, \tag{2.11}$$

$$\mathbf{K} = \mathbf{Z}^{(l-1)}\mathbf{W}_K, \tag{2.12}$$

$$\mathbf{V} = \mathbf{Z}^{(l-1)}\mathbf{W}_V, \tag{2.13}$$

where $\mathbf{W}_Q \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_K \in \mathbb{R}^{d\times d_k}$, and $\mathbf{W}_V \in \mathbb{R}^{d\times d_v}$ are learnable parameter matrices. Typically, $d_k = d_v$.

The query $\mathbf{q}_i$ (a row in $\mathbf{Q}$) for a token i can be thought of as asking: *"What information do I need from other tokens to better represent myself?"*. The key $\mathbf{k}_j$ (a row in $\mathbf{K}$) from another token j represents what kind of information token j offers. The value $\mathbf{v}_j$ (a row in $\mathbf{V}$) is the actual content or representation of token j that will be aggregated if token i attends to token j.

Scaled Dot-Product Attention. The attention scores are computed by taking the dot product of each query with all keys. These scores are scaled by $\frac{1}{\sqrt{d_k}}$ to prevent overly large values which could lead to vanishing gradients in the softmax function. A softmax function is then applied to these scaled scores to obtain attention weights α_{ij}, which represent how much token i should attend to token j. The output for token i, $\mathbf{h}_i$, is a weighted sum of all value vectors. The complete operation for all tokens can be expressed in matrix form:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.14}$$

where the matrix of attention weights α_{ij} is often denoted $\mathbf{A} \in \mathbb{R}^{n\times n}$. The output $\mathbf{H} \in \mathbb{R}^{n\times d_v}$ is a sequence of context-aware representations.

Multi-Head Attention. Instead of performing a single attention function, Transformers employ *multi-head attention*. This involves projecting the queries, keys, and values H times with different, learned linear projections $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)}$ for each head $h = 1, \ldots, H$. Scaled dot-product attention is then performed in parallel

for each of these "heads", yielding H output matrices $\text{head}_h \in \mathbb{R}^{n \times d_v}$. These are concatenated and once again projected with a final weight matrix $\mathbf{W}_O \in \mathbb{R}^{Hd_v \times d}$ to produce the final output of the multi-head attention sub-layer:

$$\text{MultiHead}(\mathbf{Z}^{(l-1)}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}_O \tag{2.15}$$

where,

$$\text{head}_h = \text{Attention}(\mathbf{Z}^{(l-1)}\mathbf{W}_Q^{(h)}, \mathbf{Z}^{(l-1)}\mathbf{W}_K^{(h)}, \mathbf{Z}^{(l-1)}\mathbf{W}_V^{(h)}) \tag{2.16}$$

Typically, $d_k = d_v = d/H$, so the dimensionality of each head's output is d/H.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With multiple heads, each head can learn to focus on different aspects of the sequence, such as different types of syntactic dependencies or semantic relationships (Clark et al., 2019; Voita et al., 2019). This enhances the representational power of the model. The specific roles and specialisations of attention heads are a subject of interpretability research, relevant to Chapter 5.

Residual Connections and Layer Normalization. Each sub-layer in a Transformer (i.e., the multi-head self-attention mechanism and the position-wise feed-forward network) is followed by a residual connection (He et al., 2016) and then layer normalization (Ba et al., 2016). That is, if Sublayer($\mathbf{X}$) is the function implemented by the sub-layer itself acting on input $\mathbf{X}$, the output is LayerNorm($\mathbf{X}$ + Sublayer($\mathbf{X}$)). Residual connections help mitigate the vanishing gradient problem in deep networks, allowing gradients to propagate more easily through the layers during training. Layer normalization helps to stabilize the activations and improve training speed and performance by normalizing the inputs to each sub-layer independently across the feature dimension for each example in the batch.

2.3.4 Feed-Forward Network

In addition to the attention sub-layers, each layer in the Transformer encoder and decoder contains a fully connected position-wise Feed-Forward Network (FFN). This FFN is applied to each position (i.e., each token representation) separately and identically. It typically consists of two linear transformations with a non-linear activation function in between. Common activation functions include ReLU (Rectified Linear Unit) (Nair and Hinton, 2010) or GELU (Gaussian Error Linear Unit) (Hendrycks and Gimpel, 2016). For an input $\mathbf{z} \in \mathbb{R}^d$ from a specific position, the FFN is:

$$\text{FFN}(\mathbf{z}) = \text{ActivationFunction}(\mathbf{z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{2.17}$$

If using ReLU, this becomes:

$$\text{FFN}_{\text{ReLU}}(\mathbf{z}) = \max(0, \mathbf{z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{2.18}$$

And using GELU:

$$\text{FFN}_{\text{GELU}}(\mathbf{z}) = \text{GELU}(\mathbf{z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{2.19}$$

Here, $\mathbf{W}_1 \in \mathbb{R}^{d \times d_{ff}}$, $\mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d}$, and $\mathbf{b}_2 \in \mathbb{R}^d$. The dimensionality of the input and output of the FFN is d, and the inner-layer typically has a larger dimensionality d_{ff} (e.g., $d_{ff} = 4d$).

While the self-attention layers are responsible for capturing dependencies between different tokens in the sequence, the FFN sub-layer provides additional non-linear transformations to each token's representation independently. This can be seen as further processing or enriching the information aggregated by the attention mechanism for each position. It is hypothesised that FFNs play a role in storing factual knowledge or acting as key-value memories (Geva et al., 2021; Meng et al., 2022).

2.3.5 Stacked Layers and Output Layer

The Transformer stacks N identical layers (e.g., $N = 6$ or $N = 12$ for "base" models, and $N = 24$ or more for "large" models) in both the encoder and decoder

sections (Vaswani et al., 2017). This deep stacking allows the model to learn increasingly complex representations and features. This layer-wise structure and its role in encoding hierarchical information is a key area of investigation in Chapter 3 and Chapter 5.

For a decoder-only autoregressive Transformer used in CLM training, the output from the final Transformer layer, $\mathbf{Z}^{(N)} \in \mathbb{R}^{n \times d}$, where $\mathbf{z}_t^{(N)}$ is the representation for token t, is typically passed through a final linear layer (often called the language model head). This layer maps the d-dimensional hidden states to logits over the vocabulary $\mathcal{V}$. A softmax function is then applied to these logits to obtain a probability distribution over the next token:

$$p(x_{t+1} \mid x_1, \ldots, x_t; \theta) = \text{softmax}(\mathbf{z}_t^{(N)} \mathbf{W}_{\text{lm}} + \mathbf{b}_{\text{lm}}) \tag{2.20}$$

where $\mathbf{W}_{\text{lm}} \in \mathbb{R}^{d \times |\mathcal{V}|}$ and $\mathbf{b}_{\text{lm}} \in \mathbb{R}^{|\mathcal{V}|}$ are the parameters of the output layer, and θ represents all model parameters.

2.3.6 Masking Strategies

Causal (Autoregressive) Masking. For autoregressive tasks like next-token prediction in CLM training, Transformers (specifically decoder or decoder-only architectures) apply a causal mask to the self-attention mechanism (Vaswani et al., 2017). This mask ensures that when computing the representation for token i, the model can only attend to tokens at positions $j \leq i$. This is typically achieved by adding a mask matrix (with $-\infty$ for disallowed positions and 0 for allowed positions) to the scaled dot products $\mathbf{Q}\mathbf{K}^T/\sqrt{d_k}$ before the softmax operation. This masking strategy enforces the left-to-right factorization of the joint probability of the sequence, $p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_1, \ldots, x_{i-1})$, directly mirroring the chain rule and enabling generative capabilities (Radford et al., 2019). Despite this unidirectional constraint, multi-head attention still allows the model to capture complex hierarchical dependencies from the preceding context (Warstadt et al., 2024); investigating how such models learn hierarchical dependencies from purely sequential signals is a key aspect of this thesis.

Autoencoding (Bidirectional) Representations. In contrast, autoencoding models like BERT (Devlin et al., 2019) are designed to learn bidirectional representations. During pretraining with the MLM objective, a certain percentage of input tokens are randomly masked, and the model's task is to predict these masked tokens based on the *unmasked* surrounding context (both left and right). To achieve this, the self-attention mechanism in BERT's encoder stack is allowed to attend to all tokens in the input sequence without a causal mask. This bidirectional context enables the model to learn rich, deep representations that are highly effective for various natural language understanding tasks (Rogers et al., 2020). However, this approach means BERT is not directly suited for autoregressive text generation in the same way that autoregressive CLMs are.

The choice of masking strategy fundamentally influences the type of representations learned and the tasks for which the model is best suited. Chapter 3 explores how these different approaches impact the encoding of linguistic hierarchies.

2.3.7 Hierarchical Representations and In-Context Learning

In-Context Learning (Autoregressive Setting). One of the most remarkable emergent capabilities of Transformer-based Large Language Models (LLMs) is *in-context learning*, or few-shot learning (e.g., GPT-3 (Brown et al., 2020)). LLMs are typically defined and distinguished from earlier models by their vast scale: typically involving billions of parameters and web-scale training data. These models can perform new tasks or adapt their behaviour based solely on a few examples or instructions provided in their input prompt, without any updates to their parameters. The model processes the prompt, which includes task demonstrations, and then generates a completion that often correctly applies the demonstrated pattern to a new query instance. Mechanistic interpretability studies suggest that specific components within Transformers, such as "induction heads", play a role in detecting and replicating patterns from the prompt, effectively enabling this rapid, on-the-fly adaptation (Olsson et al., 2022; Elhage et al., 2021). The ability to guide these models using hierarchically structured prompts (e.g., chain-of-thought

reasoning (Wei et al., 2022; Ouyang et al., 2022)) further highlights their capacity to process and leverage nested information. Conversely, *jailbreak* attacks demonstrate that pretrained LLM human alignment fine-tuning can be bypassed with automatically discovered adversarial prompts and are now covered by comprehensive taxonomies of attacks and defences (Zou et al., 2023; Yi et al., 2024).

Relevance to Thesis Goals. The capacity of Transformers to learn hierarchical representations and perform in-context learning is central to the investigations in this thesis. Understanding how different architectural choices (e.g., causal vs. bidirectional attention, specific attention mechanisms for spatiotemporal data as in Chapter 5) and training objectives influence the emergence and nature of these hierarchical encodings is vital for building more interpretable, robust, and generalisable models. The probing methodologies developed and applied in Chapter 3 are designed to shed light on the internalisation of hierarchy (research question 1), while analyses of in-context learning and memorisation (Chapter 4) relate to the balance between hierarchical generalisation and verbatim memorisation and recall (research question 2). Furthermore, applying these insights to guide the development of unified multimodal systems (Chapter 6) addresses research question 3.

2.3.8 Representative Transformer Models

Although the original architecture depicted in Fig. 2.1 serves as a foundational blueprint, practical systems and research explorations invariably incorporate modifications tailored to specific domains, tasks, or to address computational and efficiency constraints. Table 2.1 highlights several canonical Transformer variants, including those reflecting increasing scale and diverse design choices, that are referenced or relevant to the discussions throughout this thesis.

Hierarchy-aware design trends. Many advancements in Transformer architectures, particularly for vision and long-sequence processing, explicitly incorporate

notions of hierarchy or efficiency mechanisms that interact with hierarchical processing:

- **Efficient Attention for Long Sequences:** Models like Longformer (Beltagy et al., 2020) and other sparse attention mechanisms (Child et al., 2019) aim to reduce the quadratic complexity of full self-attention, often by restricting attention to local windows combined with some global attention, which can influence how long-range hierarchical dependencies are captured.
- **Hierarchical Architectures for Vision:** Vision Transformers like Swin Transformer (Liu et al., 2021) build explicit hierarchical feature maps by merging image patches at deeper layers, mimicking the multi-scale processing of CNNs and facilitating transfer to dense prediction tasks.
- **Factorised or Decomposed Attention:** For modalities like video, models such as TimeSformer (Bertasius et al., 2021) factorise attention into spatial and temporal components, or process dimensions separately, which can be seen as a structured approach to handling different aspects of spatiotemporal hierarchy. This is relevant to the PSViT model developed in Chapter 5.
- **Cross-modal Alignment at Multiple Granularities:** Multimodal models like CLIP (Radford et al., 2021) and others often learn to align representations from different modalities (e.g., image regions with text phrases) at various levels of detail, implicitly or explicitly capturing cross-modal hierarchical correspondences, a theme explored in Chapter 6.

By grounding later experiments and discussions in the context of these concrete architectures and design trends, we can more effectively trace how theoretical claims about hierarchical encoding manifest in practice, for instance, when probing BERT layers for constituency information (Chapter 3), analysing spatiotemporal attention in custom video Transformers (Chapter 5), or considering unified multimodal frameworks (Chapter 6).

Model Family	Example Params	Domain	Notable Design Choices or Contributions
$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019)	110M	Text	Bidirectional (masked) pretraining; segment + position embeddings; strong for Natural Language Understanding (NLU) tasks.
GPT (Series) (Radford et al., 2019; Brown et al., 2020; OpenAI, 2023)	1.5B (GPT-2) to 175B (GPT-3)	Text	Causal decoder-only; large scale; emergent few-shot/in-context learning; instruction following. GPT-4 details remain partially undisclosed but build on this paradigm.
T5 (Raffel et al., 2020)	Up to 11B	Text	Unified text-to-text framework; encoder-decoder; explores transfer learning limits.
Pythia Suite (Biderman et al., 2023)	70M to 12B	Text	Fully open-source suite of decoder-only LLMs trained on public data, with all intermediate checkpoints available; enables research on LLM development and scaling.
Llama (Series) (Touvron et al., 2023a,b; AI at Meta, 2024)	7B to 70B+ (Llama 2), up to 405B (Llama 3)	Text	High-performance, open-access decoder-only LMs; focus on efficient training and strong performance across benchmarks; variants include instruction-tuned models.
Longformer (Beltagy et al., 2020)	149M	Long Text	Efficient attention (sliding window + global) for processing long sequences (e.g., 4k–8k+ tokens).
Vision Transformer (Dosovitskiy et al., 2021)	86M (ViT-B/16)	Images	Applies Transformer directly to image patches; encoder-only; minimal vision-specific inductive bias.
TimeSformer (Bertasius et al., 2021)	121M	Videos	Factorised spatial and temporal attention for video understanding; processes video clips as sequences of patch embeddings.
CLIP (Radford et al., 2021)	Up to 400M	Image+Text	Dual-encoder contrastive pretraining on image-text pairs; learns aligned vision and language representations.

Table 2.1: Illustrative Transformer models and families, highlighting their characteristics relevant to discussions in later chapters. Parameter counts are approximate and refer to specific model sizes or ranges.

2.3.9 Summary

In summary, the Transformer architecture has revolutionised sequence processing by leveraging self-attention mechanisms to capture complex dependencies within data, without relying on recurrence or explicit convolutions. Its core components: multi-head self-attention, positional encodings, position-wise FFNs, residual connections, and layer normalization, all combine to form powerful models capable of learning rich, hierarchical representations from vast amounts of data. Different configurations, such as causal masking for autoregressive generation or bidirectional context for understanding tasks, tailor the Transformer to diverse objectives.

The ability of Transformers to learn hierarchical structures implicitly, and the emergent phenomenon of in-context learning in large-scale autoregressive versions, are particularly significant. These capabilities underpin much of their success and are central to the investigations of this thesis. Understanding the interplay between the architectural design of Transformers and the nature of the representations they learn (specifically how they internalise, generalise, and apply hierarchical information) is crucial for advancing the field towards more capable, interpretable, and reliable AI systems. The following chapters will build upon this architectural foundation to:

- Investigate how different Transformer designs (CLM vs MLM) encode linguistic hierarchies (Chapter 3), addressing *how and where hierarchy is internalised*.
- Analyse how modifications to attention mechanisms, such as spatiotemporal attention in video models, can better exploit multi-level sequential information (Chapter 5), further exploring the *internalisation and application of hierarchical principles*.
- Assess how these design choices and learned representations impact phenomena like in-context learning, memorisation (Chapter 4), and generalisation across different modalities (Chapter 6), probing the *relationship between hier-*

archical generalisation and memorisation, and guiding the development of *unified multimodal systems*.

2.4 Hierarchical Reasoning

Hierarchical structures are fundamental to understanding complex information in both language and vision. In linguistics, hierarchy governs how words combine into phrases and sentences, and how meaning is composed across discourse levels (Chomsky, 1957, 1965; Jackendoff, 1977; Pollard and Sag, 1994). In vision, it describes how pixels form edges, parts, objects, and scenes (Marr, 1982; Zeiler and Fergus, 2014), and how these elements interact over time in dynamic events (Oprea et al., 2020). A robust grasp of these hierarchical properties is critical for designing, training, and interpreting modern deep learning models, particularly Transformers, which often implicitly learn to encode and exploit such structures (Tenney et al., 2019; Hewitt and Manning, 2019; Clark et al., 2019). Empirical and psycholinguistic evidence further underscores the importance of hierarchical processing in human cognition for both language comprehension (Frazier and Rayner, 1979; Stowe, 1986; Gibson, 1998) and visual perception (Palmer, 1977; Biederman, 1987; Maniglia and Öttl, 2024). This section reviews theories of hierarchy in language and vision, discusses how these concepts manifest in computational models, and explicitly catalogues probing methods and interpretability tools (ranging from linear probes to causal tracing and attention analysis) that are employed in the subsequent chapters to investigate *how and where* foundation models internalise hierarchical structures and the implications thereof.

Recent developments and open questions. The study of how neural networks, especially Transformers, acquire and represent hierarchical knowledge is a vibrant research area, directly relevant to the core questions of this thesis. Large-scale experiments now indicate that CLM Transformers can develop a preference for tree-consistent generalisations when the training signal contains sufficient semantic content (Yedetore and Kim, 2024). Conversely, synthetic studies have revealed a

striking "grokking" phenomenon, where models may initially memorise surface patterns for many epochs before abruptly uncovering and generalising based on latent hierarchical rules (Power et al., 2022; Murty et al., 2023b,a). This highlights the critical distinction between superficial memorisation and genuine hierarchical understanding, a key focus of Chapter 4. Mechanistic interpretability work has begun to pinpoint how such learning occurs, tracing hierarchical processing to specialised components like *induction heads* within attention mechanisms (Elhage et al., 2021; Olsson et al., 2022) and to the role of FFNs as key-value memories (Geva et al., 2021). Additionally, recent work on model editing shows that specific factual associations in Transformer LLMs can be precisely located and rewritten (Meng et al., 2022; Fang et al., 2025). These findings motivate the diverse probing and attention analysis methods introduced later in this section and deployed extensively in later Chapters.

2.4.1 Hierarchy in Language

Human language is widely recognised as inherently hierarchical. Words combine to form syntactic constituents like noun phrases (NPs) and verb phrases (VPs), which in turn form larger phrasal or clausal units, often recursively (Chomsky, 1957, 1965; Pollard and Sag, 1994). This hierarchical organisation is not confined to syntax; it also extends to semantic composition and discourse structure (Mann and Thompson, 1988; Kayne, 1994; Pesetsky, 1995; Sag et al., 2012). The systematic arrangement of linguistic elements in these layered structures underpins the expressive power, generativity, and flexibility of human communication (Frazier and Rayner, 1979; Pinker, 1994; Gibson, 1998).

Constituency and recursion. A central concept in modern linguistics is that language is not merely a linear sequence of tokens but is structured into larger, meaningful units called *constituents* (Chomsky, 1957, 1965; Radford, 2016). These constituents can be recursively nested, forming hierarchical parse trees that represent the grammatical structure of sentences. For example, in the sentence, *"The*

book that John read was fascinating", the NP *"The book that John read"* contains a relative clause (*"that John read"*), which itself has an internal syntactic structure. This capacity for recursion allows for the generation of an unbounded number of novel and complex sentences from a finite set of rules and lexical items (Hauser et al., 2002). Such nested structures are crucial for explaining phenomena like:

- **Long-distance dependencies:** Agreement or case marking between words that are distant in the linear string but syntactically related (e.g., subject-verb agreement across an embedded clause) (Chomsky, 1981).
- **Structural ambiguity:** Sentences that can have multiple interpretations based on different underlying constituent structures (e.g., "I saw the man with the telescope") (Jurafsky and Martin, 2009).
- **Coordination and subordination:** How phrases and clauses are linked at the same or different hierarchical levels to build complex sentences (Huddleston and Pullum, 2002).

Understanding these phenomena requires models that can process hierarchical relationships, not just linear contiguity.

Syntactic vs. semantic hierarchies. Hierarchies operate at multiple linguistic levels. *Syntactic* hierarchies pertain to the formal rules of composition that govern how words form phrases and clauses, as described by grammars (Radford, 2004; Jurafsky and Martin, 2009). *Semantic* hierarchies, on the other hand, concern how meaning is composed from smaller units to larger ones, from individual words to phrases, sentences, and entire discourses (Partee, 1995; Goldberg, 2006). For instance, the meaning of a complex phrase like *"the old red car that sped down the highway"* is built compositionally from the semantics of its constituents and their structural relationships. Modern computational models, particularly LLMs, often learn representations where syntactic and semantic information is intertwined (Linzen et al., 2016; Hewitt and Manning, 2019; Tenney et al., 2019). For example,

resolving subject-verb agreement (a syntactic phenomenon) may require understanding semantic roles, and anaphora resolution (a discourse-level semantic task) depends on identifying syntactically valid antecedents. Research has shown that data-driven models can learn *emergent* hierarchical behaviours, implicitly encoding bracketed structures or latent phrase boundaries without explicit grammatical supervision (Clark et al., 2019; Kudugunta et al., 2019; Manning, 2020). Furthermore, providing richer semantic supervision appears to guide these models towards more tree-consistent syntactic generalisation (Yedetore and Kim, 2024).

- **From morphology to discourse:** Hierarchical composition is evident from the lowest levels (morphemes combining to form words) through lexical and phrasal levels (words forming phrases expressing tense, mood, or argument structure) up to the discourse level (sentences combining to form coherent narratives or arguments) (Grosz et al., 1995; Goldberg, 2006; Linzen et al., 2016).
- **Coreference and topic flow:** Discourse-level phenomena such as pronoun resolution and topic continuity rely on analysing reference chains and thematic structures that span multiple clauses or sentences, often reflecting hierarchical discourse organisation (Tenney et al., 2019; Manning, 2020).
- **Real-time comprehension:** Psycholinguistic studies indicate that humans parse sentences incrementally, using cues like phrase boundaries and syntactic expectations to build hierarchical structures on the fly (Frazier and Rayner, 1979; Stowe, 1986; Tanenhaus et al., 1995), underscoring the cognitive reality and efficiency of hierarchical processing.

By incorporating or learning these hierarchical cues, computational models can achieve more robust ambiguity resolution, maintain better coherence, and generalise more effectively to novel linguistic inputs. The extent to which current models successfully do this, and how such internalised hierarchies are structured, is a primary focus of this thesis.

2.4.2 Hierarchical Representations in Language Models

Although Transformer models process input sequences in parallel via self-attention, causal variants like GPT still generate text token by token, conditioning strictly on past context. Remarkably, this unidirectional constraint does not prevent them from modelling long-range syntactic and semantic dependencies that are inherently hierarchical (Warstadt et al., 2024); indeed, the left-to-right factorisation may even facilitate the analysis of the hierarchical structures they learn (Rogers et al., 2020). While much foundational work on probing linguistic structure has focused on bidirectional models like BERT (Goldberg, 2019; Tenney et al., 2019; Jawahar et al., 2019), research increasingly explores these phenomena in autoregressive settings, which is particularly relevant to this thesis's focus on how such models internalise and generalise hierarchical information.

Implicit hierarchy in autoregressive attention. With a causal mask, each token attends to all preceding tokens, allowing global patterns and dependencies to emerge:

- **Syntactic dependencies:** Autoregressive models have been shown to learn and represent syntactic dependencies, including phenomena like subject-verb agreement, though their performance can vary based on context and model scale (Lakretz et al., 2021; Warstadt et al., 2024). Probing studies (discussed later) on bidirectional models show that information about such dependencies often appears in specific layers, with lower layers tracking more local agreements and higher layers encoding broader clausal grammar (Clark et al., 2019; Tenney et al., 2019); understanding if and how similar layer-wise specialisation occurs in causal models is pertinent to this thesis.
- **Recursive nesting:** On synthetic tasks designed to test for recursive structure (e.g., bracket completion or list processing), causal Transformers can infer approximate tree structures without explicit tree-based supervision, though their ability to generalise deeply recursive patterns robustly is an

area of active research (Arora et al., 2024; Fernando, 2024). Earlier work on RNNs (Shen et al., 2019) and BERT (McCoy et al., 2020) also explored similar capabilities. Performance can degrade as the required depth of recursion increases beyond that seen in training, sometimes linked to the "grokking" phase transition (Murty et al., 2023b).

- **Tree emergence at scale with semantic guidance:** Recent work suggests that when the training corpus provides rich semantic signals, CLMs tend to favour tree-consistent generalisations, sometimes outperforming bidirectional models on out-of-distribution syntactic tests (Yedetore and Kim, 2024; Ahuja et al., 2024).

These findings indicate that causal self-attention can foster internal hierarchical representations rather than merely shallow n-gram statistics, particularly when the training objective and data implicitly reward such structure. The precise mechanisms and robustness of these learned hierarchies in causal models, and how they are internalised (research question 1), remain an active area of investigation (López-Otal et al., 2024).

Few-shot prompts as hierarchical scaffolds. A hallmark of text-generative CLMs (e.g., GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022)) is *in-context learning*: the model adapts to new tasks by conditioning on a few examples provided in the prompt, without weight updates. Mechanistic interpretability research has identified specialised "induction heads" in attention layers that detect and replicate token subsequences from the prompt, effectively performing pattern completion over variable-length chunks that can be hierarchically structured (Elhage et al., 2021; Olsson et al., 2022). Prompting strategies that explicitly provide intermediate reasoning steps (chain-of-thought prompting (Wei et al., 2022)) can further improve performance on complex tasks (Ouyang et al., 2022), underscoring the model's ability to leverage explicitly structured hierarchical information in its input.

Memorisation, chunking, and abstraction. The capacity of LLMs to store and process information also leads to challenges like *verbatim memorisation* of training data (Carlini et al., 2021, 2022), a topic explored in detail in Chapter 4. The interplay between learning useful hierarchical abstractions and rote memorisation is complex, forming the core of this thesis's second research question, and is the focus of the contributions discussed in Chapter 4.

- **Memorisation vs. generalisation dynamics:** Models might initially memorise surface n-grams before later "grokking" underlying hierarchical rules (Feldman, 2020; Power et al., 2022; Murty et al., 2023b). Understanding this transition is key to distinguishing robust generalisation from superficial pattern matching.
- **Role of feed-forward networks:** FFNs in Transformers have been characterised as key-value memories that can store factual knowledge and multi-word expressions as somewhat atomic units (Geva et al., 2021; Meng et al., 2022; Dai et al., 2022). While this aids fluency and knowledge recall, it can also contribute to generating memorised text, potentially at the expense of generalising hierarchical patterns.

A key goal of interpretability research, therefore, is to disentangle beneficial hierarchical abstraction (e.g., learning syntactic rules, semantic roles) from potentially harmful verbatim memorisation and recall, or superficial pattern matching.

2.4.3 Hierarchy in Vision

Hierarchical organisation is as fundamental to visual perception as it is to language. At the lowest level, pixel intensities form local features like edges and textures; these combine into parts, which assemble into objects and scenes; and in dynamic visual data like video, objects and scenes evolve and interact over time to form events (Marr, 1982; Biederman, 1987; Oprea et al., 2020). Classical CV systems, particularly CNNs, explicitly encoded such multi-scale processing through stacked

layers of convolutions and pooling operations, where early layers learn low-level features and deeper layers learn more abstract, complex structures (LeCun et al., 1998; Zeiler and Fergus, 2014). More recent Vision Transformer (ViT) models learn visual hierarchies end-to-end by applying self-attention mechanisms to sequences of image patches or video tubelets (Dosovitskiy et al., 2021; Bertasius et al., 2021; Arnab et al., 2021). This thesis explores whether similar principles of hierarchical internalisation observed in language models extend to the visual temporal domain.

Illustrative cases and challenges in video. Understanding dynamic scenes in video requires processing information hierarchically across space and time:

- **Object permanence and tracking:** Identifying and tracking an object (e.g., a ball in motion) as it becomes occluded and reappears requires binding fine-grained appearance cues to a longer-range spatiotemporal trajectory (Kuehne et al., 2018; Yun et al., 2022).
- **Action recognition and compositionality:** Recognising a complex action (e.g., "overtaking on a motorway") often involves identifying a sequence of simpler sub-actions ("checking mirrors", "signalling intent", "accelerating into the adjacent lane", "clearing the vehicle"), each composed of even finer-grained movements. This mirrors clausal embedding in language (Feichtenhofer et al., 2019; Sener et al., 2020).
- **Scene understanding and transitions:** Interpreting events across scene changes (e.g., physical objects entering a scene and interacting together, or performing human action recognition (both explored in Chapter 6)) requires models to build abstract scene representations that persist or evolve coherently across abrupt visual shifts (Tapaswi et al., 2016; Oprea et al., 2020).

Effectively modelling such phenomena necessitates architectures that can integrate information across multiple spatiotemporal scales, a form of hierarchical processing.

Hierarchical spatiotemporal attention in Transformers. Transformer variants for video understanding often incorporate explicit hierarchical design choices or factorised attention mechanisms to manage the complexity of spatiotemporal data:

- **Factorised space-time attention:** Models like TimeSformer (Bertasius et al., 2021) and ViViT (Arnab et al., 2021) often separate spatial attention (within frames) from temporal attention (across frames), allowing local spatial features to be aggregated before reasoning about temporal dynamics.
- **Progressive patch merging and hierarchical feature pyramids:** Architectures such as the Swin Transformer (Liu et al., 2021) and its video extensions (e.g., Video Swin Transformer (Liu et al., 2022)) create hierarchical representations by progressively merging tokens corresponding to adjacent spatial or spatiotemporal regions, effectively creating multi-scale feature pyramids similar to CNNs. Multiscale Vision Transformers (MViT) also employs pooling to create hierarchical feature representations (Fan et al., 2021).
- **Slow-Fast pathways inspiration:** Though originally developed for CNNs, the SlowFast concept (Feichtenhofer et al., 2019): using a high-frame-rate "fast" pathway to capture rapid motion and a low-frame-rate "slow" pathway for semantics, has inspired hybrid or analogous designs in video Transformers, enabling them to process information at different temporal resolutions.

These architectural adaptations aim to provide inductive biases that facilitate the learning of spatiotemporal hierarchies. SSL pretraining strategies for video, such as VideoMAE (Tong et al., 2022), also leverage these backbones to learn powerful representations from unlabelled video data. The PSViT model developed in Chapter 5 builds on these ideas, using causal attention in pixel space to model physical dynamics and investigate the internalisation of hierarchical physical rules.

Hierarchy, interpretability, and transfer in vision models. Similar to language models, probing studies on Vision Transformers suggest a layer-wise emer-

gence of hierarchical features. Early layers tend to focus on low-level features like edges and textures within patches, middle layers may learn to group parts of objects, and final layers often represent more global scene context or object categories (Dosovitskiy et al., 2021; Caron et al., 2021). Such structured representations can aid transfer learning, where features learned for one task (e.g., image classification) are adapted for others (e.g., object detection, video action recognition (Bao et al., 2022; Yoo et al., 2023)). A lack of appropriate temporal hierarchy can lead to models making brittle, short-horizon predictions in video tasks, motivating the exploration of spatiotemporal inductive biases in Chapters 5 and 6 as part of understanding how to guide more robust hierarchical reasoning.

2.4.4 Integrating Language and Vision Hierarchies

Many real-world AI applications require *joint* reasoning over linguistic and visual hierarchies. For instance, an image captioning system must align noun phrases with object regions while mapping clausal semantics to global scene properties (Anderson et al., 2018; Wang and et al., 2022). In video description or question answering, a multi-clause linguistic narrative often corresponds to a sequence of complex spatiotemporal interactions unfolding across frames (Yan et al., 2021; Lei et al., 2021; Alayrac et al., 2022). Successfully bridging these modalities demands architectures that can represent and align hierarchical structures on both sides, learning correspondences at multiple levels of granularity. This directly informs the third research question of the thesis concerning unified multimodal systems.

Tasks requiring spatiotemporal–linguistic coherence. For video-language tasks, models must achieve coherence between unfolding events and their linguistic descriptions. Below are two relevant tasks:

- **Video Question Answering:** Models like Flamingo (Alayrac et al., 2022) process sequences of video frames and a textual question, often using cross-attention from the text (being generated or processed) to relevant video

frames to ground the answer. Hierarchical prompting can guide these models to consider both recent visual input and broader narrative context.

- **Video Summarisation/Description:** Generating textual summaries or detailed descriptions of video content requires mapping complex, temporally extended actions and events to appropriate linguistic structures (clauses, sentences, paragraphs) (Zhang et al., 2020; Bertasius et al., 2021; Yan et al., 2021).

Chapter 6 explores a unified next-frame prediction approach to tackle such multimodal tasks, aiming to learn these alignments implicitly through shared hierarchical representations.

Cross-modal alignment strategies for multimodal tasks. Various approaches have been developed to learn joint language-vision representations:

- **Multi-granularity alignment:** Models like X-VLM (Wang and et al., 2022) and Oscar (Li et al., 2020) explicitly aim to align visual tokens (e.g., image patches or regions) with textual tokens (e.g., words or phrases) at different scales, from local object-noun correspondences to global image-sentence mappings.
- **Contrastive learning for joint embeddings:** Approaches such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) learn a shared embedding space where corresponding image-text pairs have high similarity. While often operating at a global image-text level, extensions explore finer-grained alignments (Li et al., 2021).
- **Unified vision-language pretraining:** Models like SimVLM (Wang et al., 2022) and CoCa (Yu et al., 2022) employ large-scale pretraining on image-text data using objectives that encourage both image understanding and text generation conditioned on visual input, implicitly learning to map linguistic structures to visual content.

- **Iterative attention and fusion for complex inputs:** Architectures like Perceiver IO (Jaegle et al., 2021) and PaLI (Chen et al., 2022) use cross-attention mechanisms to flexibly integrate information from very large visual inputs (many patches) with textual queries or prompts, enabling reasoning over detailed multimodal contexts.

Interpretability and Probing in Multimodal Models. Understanding how these complex models make decisions and what representations they learn is crucial for trust and development, particularly concerning their grasp of compositional and hierarchical multimodal information. Interpretability techniques are adapted and extended to the multimodal setting:

- **Attention Analysis:** Visualising attention maps in Transformers can offer insights into which parts of an image or video a model focuses on when processing related text, or vice-versa (Clark et al., 2019; Hila et al., 2021). However, directly equating attention with explanation can be misleading (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019).
- **Probing Multimodal Representations:** Similar to unimodal probing (Chapter 3), linear classifiers or other simple models can be trained on intermediate representations from multimodal models to test if they encode specific unimodal or cross-modal properties (e.g., object categories from text-conditioned image features, or syntactic roles from visually-grounded language representations) (Hewitt and Manning, 2019; Belinkov and Glass, 2019; Hendricks and Pinter, 2021).
- **Causal Tracing and Mediation Analysis:** More advanced techniques aim to understand the causal effect of specific model components (e.g., individual neurons or attention heads) on model outputs. For example, causal tracing can identify and even edit factual knowledge stored in language models (Meng et al., 2022; Dai et al., 2022). Applying such methods to multimodal models could reveal how information from different modalities is integrated and transformed (Vig et al., 2020). Circuit-based analysis, inspired by work like Olah

et al. (2020) and Elhage et al. (2021), seeks to understand the "algorithms" learned by specific pathways within the network, including those involved in multimodal fusion or cross-modal reasoning like induction heads (Olsson et al., 2022).

These interpretability tools are vital for dissecting whether models are truly performing hierarchical, compositional reasoning across modalities or relying on superficial correlations, thereby informing the development of more robust multimodal systems as per this thesis's third research question.

2.4.5 Conclusion and Outlook for Hierarchical Reasoning

This section has underscored the pervasive nature of hierarchy across language and vision, and the increasing capacity of Transformer-based models to learn and leverage such structures. In language, hierarchical representations manifest from syntactic constituency to discourse coherence. In vision, they span from local features to complex spatiotemporal events. Modern deep learning models, especially large-scale Transformers, demonstrate an emergent ability to capture these layers, even when trained on ostensibly flat sequence prediction objectives (Tenney et al., 2019; Hewitt and Manning, 2019; Dosovitskiy et al., 2021).

Recent research highlights that the conditions of training, including the nature of the data and the specifics of the learning objective, significantly influence whether models generalise via hierarchical rules or resort to simpler heuristics or memorisation (Power et al., 2022; Murty et al., 2023b; Yedetore and Kim, 2024). This distinction is central to the research questions posed in this thesis. For instance, semantically rich training signals appear to promote tree-consistent generalisations in language models, while in vision, architectures with explicit multi-scale processing or factorised attention mechanisms often achieve superior performance on complex video tasks (Bertasius et al., 2021; Liu et al., 2022).

The integration of language and vision further complicates but also enriches the study of hierarchical reasoning, requiring models to build and align hierarchies

across modalities. The interpretability tools discussed, from probing to causal analysis, are becoming increasingly important for understanding these complex systems and ensuring that their impressive performance is built upon robust and generalisable internal representations rather than shallow pattern matching.

The overarching lessons from the study of hierarchical reasoning in these domains suggest that:

1. Hierarchy is a fundamental organising principle for complex data, and models that capture it tend to generalise better and are often more interpretable.
2. Scale and appropriate inductive biases (architectural or data-driven) play crucial roles in the emergence of hierarchical understanding in models.
3. Multimodal reasoning presents new frontiers for understanding how different types of hierarchical structures can be learned, aligned, and composed, offering a path towards more unified AI systems.

2.5 Epilogue

This chapter has laid the essential theoretical and methodological groundwork necessary to support the empirical investigations presented in the remainder of this thesis, which aim to address our central research questions on hierarchical reasoning in Transformers. We began by covering foundational concepts in machine learning and probability theory, establishing the basis for understanding model training and evaluation. We then looked into the principles of language modelling, discussing various paradigms such as CLM (autoregressive) and MLM (autoencoding) training objectives, the transformative role of SSL within NLP, the rise of pretrained LLMs with contextual representations, methods for their adaptation, and the shift towards the text-to-text paradigm with emergent in-context learning capabilities.

We then detailed the Transformer architecture, from its core components like self-attention and positional encodings to its configurations in powerful autoregressive and autoencoding models. Following this, we explored the critical concept

of hierarchical reasoning, surveying its manifestations in language and vision, its emergence in computational models, and tools for its analysis. Together, these discussions provide the theoretical foundation for our thesis's empirical contributions: understanding how models internalise hierarchical structures, the interplay between generalisation and memorisation, and the development of unified multimodal systems.

The subsequent chapters of this thesis will now turn these foundational observations and concepts into concrete experiments, each addressing specific aspects of hierarchical reasoning in generative Transformer models:

- Chapter 3 explores *where* and *how* hierarchical linguistic signals are encoded within contemporary LLMs, directly addressing the first research question. This will involve employing techniques such as linear probes, alongside analysis of attention mechanisms including induction-head tracing, and potentially causal mediation tests to understand the internal representations.
- Chapter 4 tackles the critical trade-off between hierarchical abstraction and undesirable memorisation, particularly during model adaptation, thus investigating the second research question. It proposes early-warning diagnostics and an n-gram-aware regularisation technique designed to mitigate verbatim leakage while preserving valuable learned abstractions.
- Chapter 5 extends the hierarchical analysis from the linguistic domain to dynamic video modelling. It introduces PSViT: a pixel-space spatiotemporal Transformer, and investigates whether causal attention, when equipped with appropriate inductive biases, can effectively recover and represent latent physical dynamics from raw video, contributing to research questions 1 and 3.
- Chapter 6 proposes and demonstrates a unified next-frame prediction formulation to advance research question 3. This approach aims to allow a single Transformer model to operate across diverse modalities including text, images, audio, and video, positioning hierarchical alignment as a key prin-

ciple for achieving truly general perception and learning shared hierarchical representations.

Taken together, these studies argue that a *hierarchy-centred analysis* is not merely descriptive but can be prescriptive: it offers insights that can guide architectural choices, help safeguard against issues like memorisation, and illuminate a path towards more transparent, robust, and versatile multimodal foundation models.

Bibliography

Ahuja, A., Van Dyke, B., and Bender, E. M. (2024). Learning syntax without planting trees: Understanding hierarchical generalization in transformers. *Transactions of the Association for Computational Linguistics*, 12:568–588. Preprint arXiv:2310.17026.

AI at Meta (2024). The Llama 3 technical report. Technical report, Meta AI.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., and et al. (2022). Flamingo: A visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846.

Arora, G., Singh, C., Anand, A., Anand, R., A., Jain, S., Zhang, C., and Risteski, A. (2024). Mechanistic evaluation of transformers and state space models. *arXiv preprint arXiv:2405.15105 (Note: Search result showed 2505.15105, corrected to 2405.15105)*.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Balde, G., Roy, S., Mondal, M., and Ganguly, N. (2024). MEDVOC: Vocabulary adaptation for fine-tuning pre-trained language models on medical text summarization. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*.

Bao, H., Dong, L., Piao, S., and Wei, F. (2022). Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations (ICLR)*.

Belinkov, Y. and Glass, J. (2019). Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. In *Findings of EMNLP*, page 152–162.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*.

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. (2023). Pythia: A suite of fully open-source large language models trained on public data. *arXiv preprint arXiv:2304.01373*.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115.

Bishop, C. M. (2006a). *Pattern Recognition and Machine Learning*. Springer.

Bishop, C. M. (2006b). *Pattern Recognition and Machine Learning*. Springer, Berlin, Heidelberg.

Bommasani, R., Hudson, D. A., Adeli, E., and et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pages 1877–1901. Curran Associates, Inc.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., and Lee, K. (2021). Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. (2022). Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*. Appeared at IEEE Symposium on Security and Privacy 2023.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. pages 9650–9660.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 1597–1607.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al. (2022). Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.

Chi, C., Liu, B. A., Yin, H., Li, A., Li, P., Liu, J., Yu, L., Wei, F., and Kong, L. (2024). What is wrong with perplexity for long-context language modeling? In *The Twelfth International Conference on Learning Representations (ICLR)*.

Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Chomsky, N. (1957). *Syntactic Structures*. Mouton de Gruyter, The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Chomsky, N. (1981). *Lectures on government and binding*. Foris Publications.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., and et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Dai, A. M. and Le, Q. V. (2015). Semi-supervised sequence learning. *arXiv preprint arXiv:1511.01432*.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. (2022). Knowledge neurons in pretrained transformers. *Transactions of the Association for Computational Linguistics*, 10:571–585.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1.

Deng, L. and Liu, Y. (2018). *Deep Learning in Natural Language Processing*. Springer Publishing Company, Incorporated, 1st edition.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Elhage, N., Nanda, N., Olsson, C., and et al. (2021). A mathematical framework for transformer circuits. Transformer Circuits Thread, Distill. doi:10.23915/distill.00030.

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Fan, H., Xiong, B., Mangalam, K., et al. (2021). Multiscale vision transformers. In *Proc. ICCV*.

Fang, J., Jiang, H., Wang, K., Ma, Y., Shi, J., Wang, X., He, X., and Chua, T.-S. (2025). Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

Feldman, V. (2020). Does learning require memorization? a short tale about a long tail. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pages 20200–20211. Curran Associates, Inc.

Fernando, C. (2024). Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *Computational Linguistics*, pages 1–35.

Frazier, L. and Rayner, K. (1979). Parsing temporarily ambiguous sentences: Eye movements in the resolution of syntactic category ambiguities. *Journal of Memory and Language*, 18(1):37–68.

Gadre, S. Y., Smyrnis, G., Shankar, V., Gururangan, S., Wortsman, M., Shao, R., Mercat, J., Fang, A., Li, J., Keh, S., Xin, R., Nezhurina, M., Vasiljevic, I., Jitsev, J., Soldaini, L., Dimakis, A. G., Ilharco, G., Koh, P. W., Song, S., Kollar, T., Carmon, Y., Dave, A., Heckel, R., Muennighoff, N., and Schmidt, L. (2024). Language models scale reliably with over-training and on downstream tasks.

Geva, M., Schuster, R., Berant, J., and Levy, O. (2021). Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5484–5495. Association for Computational Linguistics.

Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Goldberg, A. E. (2006). *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, Oxford.

Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Goldberg, Y. (2019). Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z. D., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Grosz, B. J., Joshi, A. K., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330. PMLR.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, New York, 2 edition.

Hauser, M. D., Chomsky, N., and Fitch, W. T. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298(5598):1569–1579.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16010.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hendricks, J. and Pinter, Y. (2021). Probing multimodal machine translation with a new benchmark: Ambiguous expression checking (aec). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3300–3309.

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 4129–4138.

Hila, C., Gur, S., and Wolf, L. (2021). Visualizing and understanding attention in vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3196–3205.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. In *Proceedings of Interspeech*, pages 1474–1478.

Huddleston, R. and Pullum, G. K. (2002). *The Cambridge grammar of the English language*. Cambridge University Press.

Jackendoff, R. S. (1977). *X-bar syntax: A study of phrase structure*, volume 2 of *Linguistic Inquiry Monographs*. MIT Press.

Jaegle, A., Gimeno, F., Brock, A., et al. (2021). Perceiver: General perception with iterative attention. In *Proc. ICML*.

Jain, S. and Wallace, B. C. (2019). Attention is not explanation. In *Proceedings of NAACL-HLT*, pages 3543–3556.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3651–3657.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. pages 4904–4916.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing*. Pearson Prentice Hall, Upper Saddle River, NJ.

Jurafsky, D. and Martin, J. H. (2023). *Speech and Language Processing (3rd ed. draft)*.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*. v4.

Kayne, R. S. (1994). The antisymmetry of syntax. *Linguistic Inquiry Monographs*, 25.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA. Curran Associates Inc.

Kudugunta, S., Khot, T., Sabharwal, A., and Kalyan, A. (2019). Investigating multiscale representation learned by bert. *ArXiv preprint arXiv:1908.11164*.

Kuehne, H., Richard, A., and Gall, J. (2018). Hypermotion: Recurrent multi-scale model for feature extraction in video data. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 535–544. IEEE.

Lakretz, Y., Hupkes, D., Verga, P., Cotterell, R., Kruszewski, G., and Baroni, M. (2021). Mechanisms of syntactic generalization in human and artificial intelligence. *Cognitive Science*, 45(10):e13042.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 6402–6413. Curran Associates, Inc.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324.

Lei, J., Li, L., Wang, L., Shen, Y., trycatch, P., Verma, V., Wang, L., Liu, Z., and Wang, L. (2021). Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7331–7341.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. (2021). Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 9694–9705.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European conference on computer vision (ECCV)*, pages 121–137. Springer.

Liang, P., Hashimoto, T., et al. (2022). Holistic evaluation of language models. Technical report, Stanford CRFM.

Linzen, T., Dupoux, E., and Goldberg, Y. (2016). Assessing the ability of LSTMs to learn syntax-sensitive dependencies. In *ACL 2016*, pages 1–11.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. ICCV*.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. (2022). Video Swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

López-Otal, M., Gracia-Bernad, A., Bobed, C., Pitarch-Ballesteros, J., and Anglés-Herrero, J. (2024). Linguistic interpretability of transformer-based language models: a systematic review. *arXiv preprint arXiv:2404.08001*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Maniglia, T. and Öttl, J. (2024). A computational framework to study hierarchical processing in visual narratives. *Cognitive Science*, 48(1):e13396.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. In *Text-Interdisciplinary Journal for the Study of Discourse*, volume 8, pages 243–281.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Manning, C. D. (2020). Emergent linguistic structure in transformer-based models. *Philosophical Transactions of the Royal Society B*, 375(1791):20190536.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

McCoy, R. T., Min, Y., and Pavlick, E. (2020). BERT's decoding of syntactic structure: Is tree travel necessary? In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*.

Meng, K., Bau, D., Andor, D., Belinkov, Y., Zheng, T., Yao, Z., and Weiss, P. (2022). Locating and editing factual knowledge in GPT. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Workshop Rrepresentativeations at ICLR*.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of INTERSPEECH*, pages 1045–1048.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, MA.

Murty, S., Sharma, P., Andreas, J., and Manning, C. (2023a). Grokking of hierarchical structure in vanilla transformers. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 439–448, Toronto, Canada. Association for Computational Linguistics.

Murty, S., Sharma, P., Andreas, J., and Manning, C. D. (2023b). Grokking of hierarchical structure in vanilla transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 439–448.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Nogueira, R. and Cho, K. (2020). Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. (2020). Zoom in: An introduction to circuits. *Distill*, 5(3):e00024.001.

Olsson, C., Nanda, N., Lieberum, T., et al. (2022). In-context learning and induction heads. *arXiv: 2209.11895*.

OpenAI (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., and Argyros, A. (2020). A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2806–2826.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., and et al. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Palmer, S. E. (1977). Hierarchical structure in perceptual representation. *Cognitive psychology*, 9(4):441–474.

Partee, B. H. (1995). Lexical semantics and compositionality. *An invitation to cognitive science*, 1:311–360.

Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pesetsky, D. (1995). Zero syntax: Experiencers and cascades. MIT Press.

Peters, M. E., Ammar, W., Bhagavatula, C., and Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 3rd Workshop on Representation Learning for NLP (RepL4NLP@ACL)*, pages 7–14.

Pinker, S. (1994). *The Language Instinct: How the Mind Creates Language*. William Morrow and Company, New York.

Pollard, C. and Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. (2022). Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Press, O., Smith, N. A., and Lewis, M. (2022). Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations (ICLR)*. arXiv preprint arXiv:2108.12409.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *AI Open*, 1:38–62.

Radford, A. (2004). *Minimalist Syntax: Exploring the Structure of English*. Cambridge University Press, Cambridge.

Radford, A. (2016). *Analysing English Sentences: A Minimalist Approach*. Cambridge University Press.

Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. In *Proc. ICML*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI Blog*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1:9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ramachandran, P., Liu, P. J., and Le, Q. V. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 136–146.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T. (2019). Transfer learning in natural language processing. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 15–18.

Sag, I. A., Wasow, T., and Bender, E. M. (2012). *Sign-Based Construction Grammar*. CSLI Publications, Stanford, CA.

Salman, S., Kadhm, S. A., and Sahib, M. A. (2023). Transfer learning and its role in machine learning. *EasyChair Preprint no. 11137*.

Sener, F., Singhania, D., Mahdisoltani, F., and Yao, A. (2020). Temporal action segmentation from timestamp supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1009–1018.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Shannon, C. E. (1951). Prediction and entropy of printed english. *Bell System Technical Journal*, 30(1):50–64.

Shaw, P., Uszkoreit, J., and Vaswani, A. (2018). Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

Shen, Y., Tan, S., Sordoni, A., and Courville, A. (2019). Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations (ICLR)*.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306.

Stowe, L. A. (1986). Sentence processing and the processing of empty categories. In *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, pages 495–500.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *NeurIPS 2014*, pages 3104–3112.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K. M., and Sedivy, J. C. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science*, 268(5217):1632–1634.

Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., and Fidler, S. (2016). Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4593–4601.

Tong, Z., Song, Y., Wang, J., Zhu, Y., Shen, Y., Yan, Y., and Yang, J. (2022). VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023a). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *NeurIPS 2017*, pages 5998–6008.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. (2020). Causal mediation analysis for interpreting neural nlp: The case of gender bias. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 12729–12740.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting. In *ACL 2019*, pages 5797–5808.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Wang, P. and et al. (2022). Learning multi-granular vision language alignments for vision-language pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Wang, Z., Gan, Z., Liu, Y., et al. (2022). Simvlm: Simple visual language model pretraining with weak supervision. In *Proc. ICLR*.

Warstadt, A., Choshen, L., and Cotterell, R. (2024). Incremental sentence processing mechanisms in autoregressive transformer language models. *arXiv preprint arXiv:2404.16014 (To appear NAACL 2025)*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pages 24824–24837. The author list reflects the NeurIPS publication; Brian Ichter was on the arXiv version.

Wiegreffe, S. and Pinter, Y. (2019). Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11–20.

Yan, W., Zhang, Y., Abbeel, P., and Srinivas, A. (2021). Videogpt: Video generation using vq-vae and transformers. In *arXiv preprint arXiv:2104.10157*.

Yang, L., Song, Y., Ren, X., Lyu, C., Wang, Y., Zhuo, J., Liu, L., Wang, J., Foster, J., and Zhang, Y. (2023). Out-of-distribution generalization in natural language processing: Past, present, and future. pages 7849–7865.

Yedetore, A. and Kim, N. (2024). Semantic training signals promote hierarchical syntactic generalization in transformers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Yi, S., Liu, Y., Sun, Z., Cong, T., He, X., Song, J., Xu, K., and Li, Q. (2024). Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Yoo, J.-S., Lee, H., and Jung, S.-W. (2023). Hierarchical spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*.

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. (2022). CoCa: Contrastive captioners are image–text foundation models. *arXiv preprint arXiv:2205.01917*.

Yun, H., Kim, J., Cho, H.-C., and Noh, H. (2022). Look at adjacent frames: Video representation learning by modeling temporal relations.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision (ECCV)*, pages 818–833. Springer.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *Proc. ICLR*.

Zhang, C., Gan, Z., Wang, L., Chen, M., Liu, Z., and Liu, J. (2020). Vist-probing: A new benchmark for visually grounded story generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2597–2607.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models.

Chapter 3

Hierarchical Information in Contextual Representations

While Transformer LLMs demonstrate impressive benchmark performance, a clear understanding of *how* and *where* they internalise linguistic hierarchy remains elusive. This chapter aims to shed light on this by treating every contextualised word embedding as a diagnostic probe into the model's latent structure. Concretely, we address three related questions:

1. At which layers do syntactic and semantic hierarchies first become linearly recoverable?
2. How do architectural choices, autoregressive vs autoencoding, permutation language modelling, or locality-sensitive hashing, shape the depth-wise distribution of that recoverable hierarchical information?
3. What footprint does task-specific fine-tuning leave on these distributions, and what does this reveal about the interplay between hierarchical learning and potential memorisation?

To answer these questions, we design two parallel suites of *ancestor probing tasks*. Focusing on sentiment and syntactic hierarchies, these tasks are applied across diverse Transformer architectures (BERT, GPT-2, XLNet, and Reformer). Our

analysis reveals that distinct architectural paradigms and fine-tuning regimes systematically shape how these models encode such structural knowledge layer-by-layer within individual token representations. The probing framework and the understanding of internalised linguistic hierarchy developed here are foundational, providing the analytical lens for our subsequent investigations into memorisation (Chapter 4), spatiotemporal reasoning (Chapter 5), and multimodal unification (Chapter 6).

The remainder of this chapter is organised as follows. Section 3.1 provides a detailed introduction to the challenge of understanding linguistic features in pre-trained language models and further elaborates on this chapter's specific research aims and contributions. Section 3.2 describes the pre-trained Transformer models chosen for our comparative analysis. Section 3.3 then details the ancestor probing tasks, the construction of datasets from the Stanford Sentiment Treebank, our fine-tuning procedures, and non-linear experimental setups. The core empirical findings from our extensive probing experiments are presented and discussed in Section 3.4, covering layer-wise performance distributions and the impact of fine-tuning.

3.1 Introduction

Pre-trained Transformer-based LLMs have enabled widespread transferability and performance gains across many NLP tasks, yet detailed knowledge about the linguistic features they produce is still developing. The pre-training of neural language models has become ubiquitous in NLP, driven largely by efforts to improve the quality of linguistic features contained within word embeddings. This has resulted in *contextual word embeddings*: continuous representations conditioned on the entire input context. Notable examples of this approach are ELMo (an autoregressive LSTM) (Peters et al., 2018b), BERT (an autoencoding Transformer model) (Devlin et al., 2019), and GPT-2 (an autoregressive Transformer model) (Radford et al., 2019). These techniques have led to significant state-of-the-art improvements on many downstream NLP tasks, highlighting the potential to advance transfer learn-

ing and language model pre-training (Howard and Ruder, 2018).

However, our understanding of the linguistic information contained within contextual representations produced by these models remains incomplete. Consequently, a thriving new area of research has emerged, employing novel analysis techniques to improve the interpretability and explainability of these models, specifically concerning how simple language-modelling pre-training leads to such effective, transferable features that appear to capture linguistic structure. Examples of such work include probing the behaviour of single neurons to identify the type of features encoded (Dalvi et al., 2019b). Analysis of single-neuron activation behaviour on test inputs suggests that specific features within a single representation are responsible for encoding attribute identifiers such as number, tense, position in sentence, and other taxonomic groups of language (Dalvi et al., 2019a). The work of Coenen et al. (2019) analyses the internal states of the BERT language model, specifically the geometry of the word representations produced in terms of syntactic and semantic subspaces. They find that linguistic features belonging to distinct linguistic taxonomic groups also appear to be represented by separate subspaces within BERT representations. This is part of a now large body of work studying the BERT language model, an area often coined *BERTology* (Rogers et al., 2021), reflecting the model's significant impact within NLP.

Despite the aforementioned work, much progress is still needed beyond task-specific performance metrics to determine how internal representations of hierarchical structure vary between architectures and how their learned features are distributed across internal layers. Works such as Peters et al. (2018a) and Liu et al. (2019) explore how the performance of probing classifiers trained on representations produced by language model architectures vary by layer depth across a range of traditional NLP tasks. They show that syntactic features tend to be encoded by shallower layers and semantic features by deeper layers of the model. Similarly, edge probing tasks were introduced in Tenney et al. (2019c) and Tenney et al. (2019a) to explore sentence-level knowledge within word representations by training classifiers limited to specific spans of the input sequence. This chapter builds

upon such probing methodologies to specifically investigate the encoding of explicit hierarchical relationships.

Our contributions in this chapter, addressing the specific questions posed at the beginning of this chapter, are threefold:

1. **Hierarchical probing tasks.** We introduce ancestor-level probes for both sentiment and syntax, requiring each token embedding to predict labels for its parent, grand-parent, great-grand-parent, and the sentence root, thereby directly testing for multi-level hierarchical information.

2. **Layer-wise analysis across architectures.** Using BERT, GPT-2, XL-Net, and Reformer (base and large), we chart how recoverable hierarchical information varies by depth, showing that autoregressive and permutation-masked variants disperse useful cues more evenly than their strictly bidirectional counterparts.

3. **Impact of fine-tuning and architectural bias.** We demonstrate that fine-tuning amplifies mid-layer abstraction of hierarchical information in XLNet and Reformer without erasing lower-level word information, whereas BERT and GPT-2 benefit less in this regard. Architectural innovations such as permutation masking or locality-sensitive hashing yield larger gains in representing hierarchy than merely scaling parameter count.

These findings reveal that hierarchical signals reside, layer by layer, in the very vectors treated as atomic word features by downstream models. The probing framework established here therefore becomes the empirical lens for Chapter 4, where we examine memorisation during fine-tuning, and for Chapters 5 and 6, where we extend the analysis to video and multimodal Transformers.

3.2 Pre-trained Transformer Contextualisers

To examine how architectural design choices influence the depth-wise encoding and internalisation of hierarchical information, we select four representative Trans-

former families: BERT, XLNet, GPT-2, and Reformer, each in *base* and *large* variants. All models are drawn from the HuggingFace `transformers` library (Wolf et al., 2019) and fine-tuned in PyTorch (Paszke et al., 2019). Parameter counts and pre-training corpora are matched as closely as possible so that observed differences may be attributed primarily to architecture rather than sheer scale. In addition, static GloVe embeddings (Pennington et al., 2014) serve as a non-contextual baseline.

BERT BERT (Devlin et al., 2019) stacks bidirectional Transformer encoders without modifying the original multi-head attention of Vaswani et al. (2017). Pre-training masks 15% of input tokens and tasks the model with reconstructing them, while an auxiliary *next-sentence prediction* objective encourages cross-sentential representations.

- **BERT (base)**: 12 layers, 768 hidden units, 12 heads, **110M** parameters.
- **BERT (large)**: 24 layers, 1024 hidden units, 16 heads, **340M** parameters.

XLNet XLNet (Yang et al., 2020) is autoregressive but mitigates the uni-directionality of standard language models via *permutation language modelling*: the model predicts each token given a random ordering of its predecessors, thereby accessing both left and right context during training. A segment-recurrence mechanism further lengthens its effective context window.

- **XLNet (base)**: 12 layers, 768 hidden units, 12 heads, **110M** parameters.
- **XLNet (large)**: 24 layers, 1024 hidden units, 16 heads, **340M** parameters.

GPT-2 GPT-2 (Radford et al., 2019) employs the Transformer *decoder* stack with causal masking, learning to predict the next token from left context only. Its strictly autoregressive training grants strong generative ability but limits immediate access to right-hand context.

- **GPT-2 (base)**: 12 layers, 768 hidden units, 12 heads, **117M** parameters.
- **GPT-2 (medium)**: 24 layers, 1024 hidden units, 16 heads, **345M** parameters.

Reformer Reformer (Kitaev et al., 2020) targets the quadratic memory cost of attention. It replaces exact dot-product attention with locality-sensitive hashing (LSH) and swaps sinusoidal positions for trainable *coordinate-wise* encodings, permitting far longer sequences on commodity hardware.

- **Reformer (base)**: 12 layers, 768 hidden units, 12 heads, **125M** parameters.
- **Reformer (large)**: 24 layers, 1024 hidden units, 16 heads, **355M** parameters.

Non-contextual reference. **GloVe 840B** (Pennington et al., 2014) (300-dim.) provides a static baseline whose performance marks the ceiling attainable without contextualisation. Any gains achieved by the probes must therefore arise from structure encoded *within* the Transformer rather than from token identity alone.

3.3 Methodology and Datasets

In this section, we outline the approach taken for dataset generation and processing. Both tasks are built upon the same underlying corpus, with the same hierarchical constituency parse trees. This allows us to perform a direct comparison for each word representation with respect to what kind of sentiment and part-of-speech information related to the hierarchy is encoded.

3.3.1 Ancestor Sentiment Classification

We leverage an existing dataset with sentiment classifications for each constituent phrase in a sentence: the Stanford Sentiment Treebank (SST) (Socher et al., 2013).

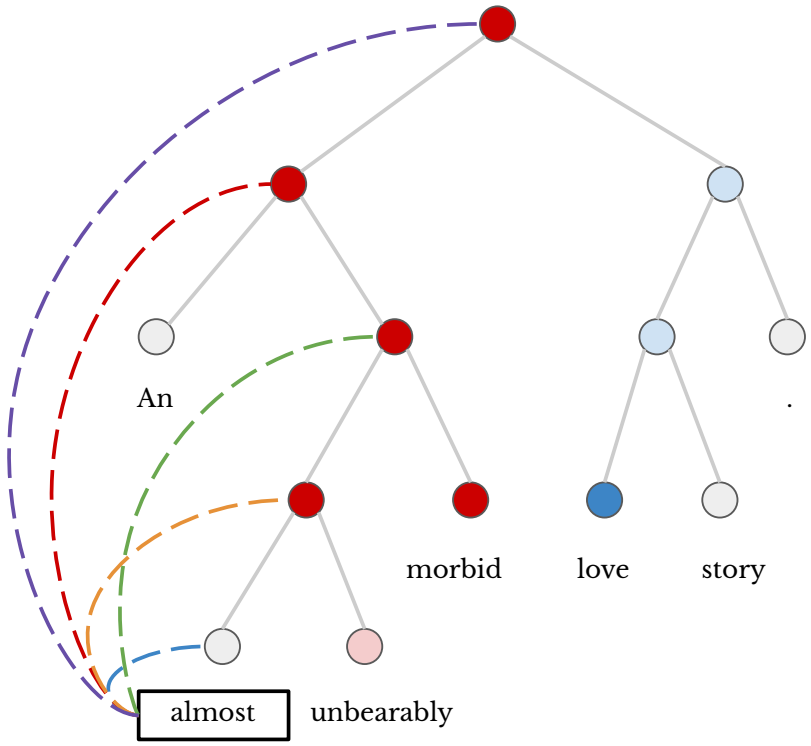


Figure 3.1: Illustration of the constituency tree structure and associated ancestor tagging levels. Sample taken from the SST dataset, where each leaf node corresponds to a token in the sentence, with the sentiment classification of each node ranging from negative (darker red) to positive (lighter blue). The dashed lines represent the nodes for the corresponding ancestor tagging levels of the highlighted token 'almost'. Purple is the root level tag, blue is the leaf tag, with red, green, and orange being Parents 3, 2, and 1, respectively.

SST contains 11,855 sentences from Rotten Tomatoes movie reviews, each parsed into a constituency tree using the Stanford Constituency Parser (Klein and Manning, 2003a) that yields 215,154 phrases annotated by three human judges. Labels were collected via an Amazon Mechanical Turk slider interface and then merged into five discrete sentiment classes (very negative to very positive) for phrase and sentence-level supervision. Each phrase/sentence received three scores on a 25-point slider; with the authors noting an average inter-rater variance of 9.7 and that human judges overwhelmingly used the five anchor positions—motivating the standard five-class discretisation of SST, as well as binary (no neutral class). Of the 11,855 sentences, the five classes are imbalanced: 1,510 very negative (12.7%), 3,140 negative (26.5%), 2,242 neutral (18.9%), 3,111 positive (26.3%), and 1,852

very positive (15.6%).

We use both the SST five-class and binary sentiment labels for each constituent phrase of a sentence. Using this hierarchically labelled sentiment data, we formulate a token-wise ancestor sentiment analysis task designed to assess how different levels of semantic (sentiment) hierarchy are encoded. For a given token in a sentence, its corresponding contextualised word representation is tasked with predicting the sentiment classification of its parent, grandparent, and great-grandparent constituent phrase. For token-level labelling, if a token is split into multiple subwords by the tokenizer, we use the embedding of the *first* subword as the token representation. Words split into a single subword are unchanged. For cases where the token does not have an ancestor phrase at a given level, the linear model is tasked to predict a 'None' classification label. As we are probing for semantic features useful for classifying sentiment at varying constituency levels, we perform all tasks using single word representations contextualised on the full root sentence.

Additionally, as each sentence has a sentence-level sentiment classification, we task each word representation to predict the overall *root* sentence sentiment. All classifiers are trained on words contained in full sentences only, with no sub-phrases included (8,544 sentences).

Figure 3.1 illustrates the constituency tree and associated ancestor tags for a given token. In the example shown, it can be clearly seen that to correctly predict the root-level sentiment for the token *'love'*, information from the left-hand side of the sentiment tree must be encoded within the representation of that token to correctly identify the sentiment negation.

3.3.2 Ancestor Constituency Phrase Tagging

We use the same SST corpus data, parsed using the Stanford Constituency Parser (Klein and Manning, 2003b), such that the constituency parse trees are identical to those found in SST, with the addition of having Constituency Part-of-Speech labels for each hierarchical phrase. The resulting task is formulated to probe syn-

tactic hierarchy: for each token in a sentence, its corresponding contextualised word representation is tasked with predicting the constituency phrase label of its parent, grandparent, and great-grandparent constituent phrases, as well as the leaf part-of-speech classification. For cases where the token does not have an ancestor phrase at a given level, the linear model is tasked to predict a 'None' classification label. The distribution is heavy-tailed: Noun Phrase (NP) is the most frequent constituent category by a wide margin, followed by Verb Phrase (VP) and Prepositional Phrase (PP), with subordinate clauses (SBAR), Adjective Phrases (ADJP), Adverb Phrases (ADVP) considerably rarer.

3.3.3 Fine-tuned Layer Performances

In addition to the linear probing tasks outlined above, we explore the impact of fine-tuning on the representation of hierarchical information. The precedent in the current state of transfer learning within NLP is to fine-tune a pre-trained Transformer model on each downstream task. Fine-tuning yields consistently better performances for the majority of tasks when compared to feature extraction, where a task-specific model processes the raw features extracted from pre-trained models.

In light of this, for comparison with raw feature extraction results, we fine-tune each Transformer model on each ancestor tagging task. The goal is to compare distributions of layer-performance for feature extraction and for fine-tuning, which could reveal how hierarchical information is modified throughout the model due to the fine-tuning process, and whether or not performance is retained at lower layers after fine-tuning.

Fine-tuning Procedure Fine-tuning of pre-trained Transformer models is far simpler and faster than the pre-training step, owing to the decision by many model designers to use additional special classification tokens during tokenisation and pre-training. These classification tokens, appended to the input context (position and value are specific to each Transformer model), are used to fine-tune the model by the addition of a linear classification layer (dimension of output is task-specific). The

benefit of using such classification tokens is that their representations are designed to aggregate information from the entire input sequence, making them suitable for sentence-level (or context-level) classification tasks. While the parameters of the appended linear classification layer are trained for the specific task, the fine-tuning process allows gradients to propagate back through the entire pre-trained model. This potentially adjusts many of its internal weights, enabling the model to adapt its representations more broadly to the nuances of the new task.

3.3.4 Non-linear Experiments

To determine how much performance is constrained by using linear-only classifiers, we train non-linear classifiers for each experiment described earlier. This will allow a comparison to non-linear models leveraging the entire input context. The non-linear classifier is a simple feed-forward ReLU (Nair and Hinton, 2010) layer matching the dimension of the embeddings for each model (768 for base, and 1024 for large).

All results reported henceforth are on the test splits of each dataset. Linear and non-linear classifiers are trained for 5 epochs using the Adam optimiser (Kingma and Ba, 2017), with a learning rate of 1×10^{-4}, and a batch size of 64. We report accuracy as the primary metric, consistent with prior sentence classification (Wang et al., 2018). To manage seed sensitivity and stay comparable to probing protocols, we run each experiment 5 times, each initialised with unique random seeds, and report the best run, mirroring multi-run reporting used to reduce noise in probing and addressing known variance across seeds (Tenney et al., 2019c).

3.4 Results and Discussion

Table 3.1 reports the best performing layer results for all linear classifier layers for each Transformer model, in addition to non-contextualised baseline comparisons using GloVe, as well as a state-of-the-art comparison (where all input tokens are used), for ancestor sentiment classification of the root (full sentence), leaf (input

Pre-trained Transformer	Leaf	Parent 1	Parent 2	Parent 3	Root
BERT (base)	92.74	64.25	58.15	54.16	39.04
GPT2 (base)	93.23	62.42	55.19	51.59	37.61
Reformer (base)	**93.46**	64.66	**59.28**	**55.96**	41.70
XLNet (base)	92.80	**65.11**	59.16	55.24	**42.45**
BERT (large)	93.13	64.24	58.27	54.15	39.06
GPT2 (large)	93.44	62.58	55.04	51.86	37.61
Reformer (large)	**94.57**	64.15	58.94	**55.10**	41.11
XLNet (large)	93.82	**65.48**	**59.20**	54.99	**43.73**
BERT Fine Tuned (base)	92.60	64.81	58.45	54.92	45.67
GPT2 Fine Tuned (base)	93.25	62.35	54.81	51.60	40.10
Reformer Fine Tuned (base)	**93.44**	64.70	60.08	**56.67**	50.08
XLNet Fine Tuned (base)	92.77	**65.86**	**60.30**	55.87	**51.58**
BERT Fine Tuned (large)	93.36	65.64	58.94	55.79	46.77
GPT2 Fine Tuned (large)	93.14	63.01	55.34	52.82	42.88
Reformer Fine Tuned (large)	**94.54**	65.05	**60.57**	55.97	51.94
XLNet Fine Tuned (large)	93.95	**66.68**	60.90	**56.58**	**53.51**
GloVe (840B.300d) (non-contextual)	90.27	60.28	47.53	39.96	28.81
State-of-the-art (all tokens)	-	-	-	-	54.70

Table 3.1: Best performing layer per contextualiser on Fine-grained Ancestor SST Classification. Entries report accuracy (%) of each pre-trained Transformer contextualiser on the 5-class ancestor sentiment analysis task. Sections are divided into models of comparable size; base and large versions of each architecture. Best performing contextualisers per task are in bold. All models and results reported are evaluated using the test split of the dataset (2,210 sentences) and a linear classifier layer.

token), and constituent parents 1, 2, and 3. Results for ancestor sentiment classification across each constituent task show that all pre-trained Transformers vastly outperform the non-contextualised baseline, confirming that a significant amount of global, hierarchical sentence-level sentiment information is contained within a single contextualised representation. Furthermore, we observe that the leaf-level sentiment classification is also improved relative to the non-contextual baseline, suggesting the quality of word-level information is not compromised by encoding contextual information into the embeddings. As for comparisons between the Transformer architectures, GPT-2 shows a drop in performance on the higher-level and root tasks relative to the other architectures. This is most likely due to GPT-2 embeddings being limited to prior context only as a result of the autoregressive pre-training procedure used, thereby having reduced access to global information relative to the other architectures.

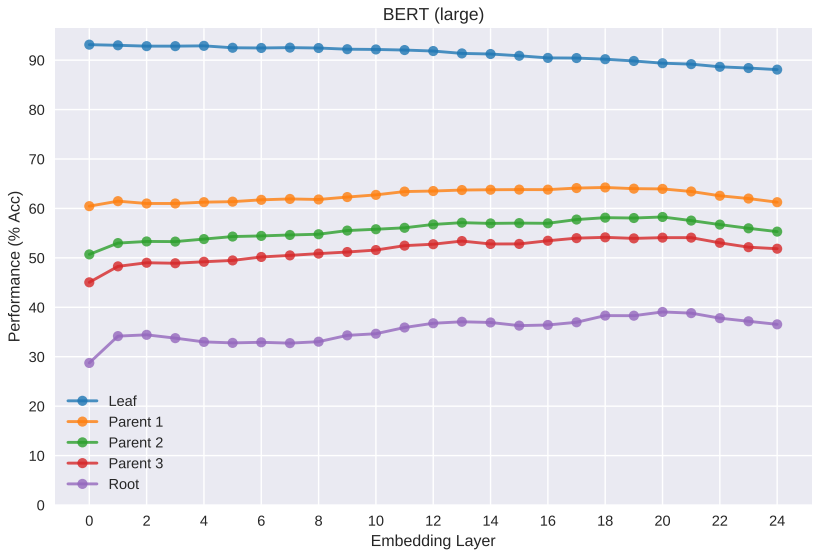
Pre-trained Transformer	Leaf	Parent 1	Parent 2	Parent 3	Parent 4
BERT (base)	94.31	88.39	74.08	59.89	60.04
GPT2 (base)	91.78	84.91	68.38	57.15	57.24
Reformer (base)	94.05	87.71	73.08	59.02	58.82
XLNet (base)	**95.58**	**88.62**	**74.42**	**60.74**	**60.72**
BERT (large)	92.78	87.53	**73.63**	**60.03**	**59.88**
GPT2 (large)	91.88	84.92	67.90	57.17	57.53
Reformer (large)	93.47	87.57	72.65	58.94	58.85
XLNet (large)	**95.64**	**88.77**	73.50	59.58	59.82
GloVe (non-contextual)	91.03	85.02	72.13	45.99	23.93
State-of-the-art (all tokens)	-	-	-	-	61.30

Table 3.2: Best performing layer per contextualiser on Ancestor Constituency Phrase Tagging. Entries report accuracy (%) of each pre-trained Transformer contextualiser on the 76-class ancestor constituency phrase tagging task. Sections are divided into models of comparable size; base and large versions of each architecture. Best performing contextualisers per task are in bold. All models and results reported are evaluated using the test split of the dataset.

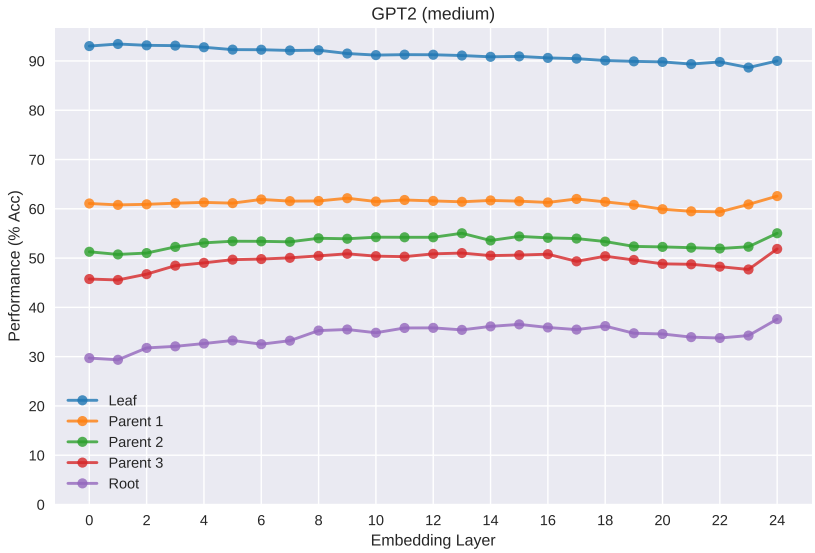
Table 3.2 reports the corresponding results for Ancestor Constituency Phrase Tagging. Similar to the results for sentiment classification, we see that all models significantly outperform the non-contextual baseline across all tasks, as well as at the leaf level where contextual information is not strictly required but can still be beneficial. GPT-2 suffers the same drop in performance on higher-level constituents as mentioned above, relative to the other architectures. Interestingly, the larger models do not show a consistent performance increase over the base models for these syntactic hierarchy tasks.

3.4.1 Layer Performance Distributions

When evaluating the base models on both tasks, we can see that the Reformer and XLNet architectures consistently outperform BERT and GPT-2. The increased gap in performance is most notable on the higher-level tasks, particularly for ancestor sentiment classification. When analysing the layer-wise performances for sentiment analysis shown in Figures 3.2, 3.3, and 3.4, we see that Reformer and XLNet have much flatter performance distributions than BERT and GPT-2, with the best performance for hierarchical sentiment often extracted from middle layer representations. This contrasts with BERT and GPT-2, where the best perform-



(a) BERT (large) - Raw Features

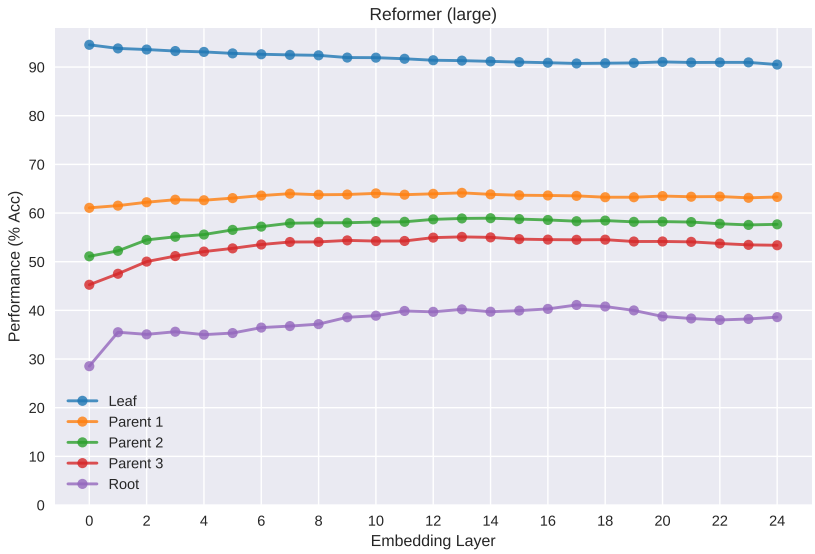


(b) GPT-2 (medium) - Raw Features

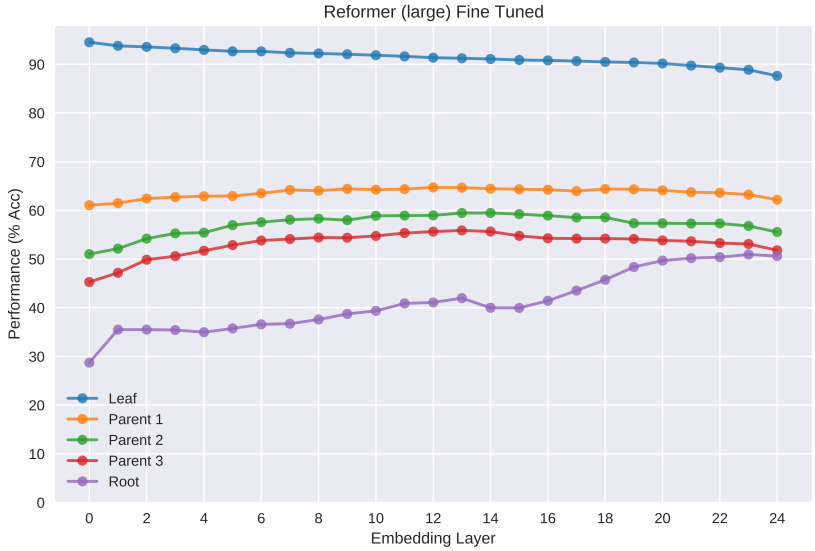
Figure 3.2: Linear classifier layer-wise performances for Ancestor Sentiment Classification tasks for (a) BERT (large) and (b) GPT-2 (medium) using raw, non-fine-tuned features.

ance for such tasks is often achieved in the later layers (for GPT-2, this is frequently the final layer).

The distribution of layer performances for constituency phrase tagging shows that Reformer, XLNet, and GPT-2 often show their best performance for syntactic hierarchy towards the early-to-mid layers of the network. Aside from leaf (word-level) classification, the figures for sentiment (e.g., Figure 3.3) suggest that the



(a) Reformer (large) - Raw Features

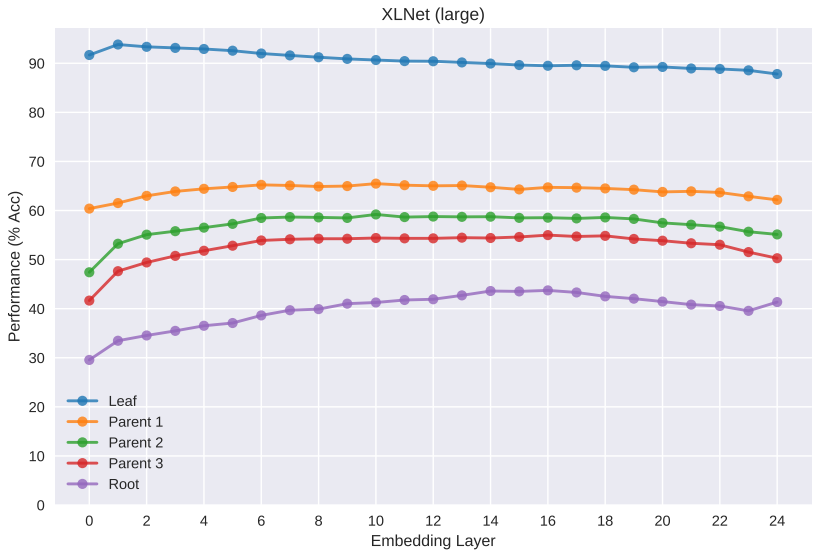


(b) Reformer (large) - Fine-Tuned

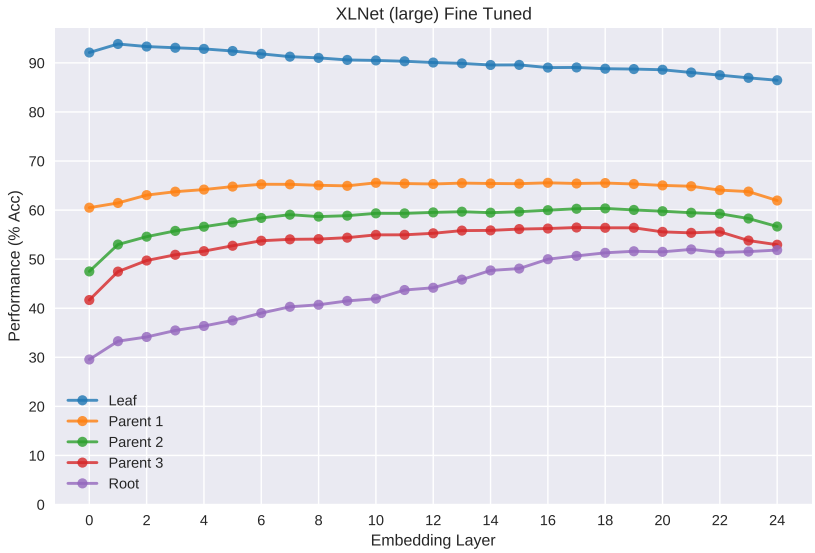
Figure 3.3: Linear classifier layer-wise performances for the Ancestor Sentiment Classification tasks for (a) Reformer (large) using raw features and (b) Reformer (large) after fine-tuning.

optimal layers for extracting hierarchical information for a given task (sentiment or syntax) do not vary drastically between different parent levels (e.g., Parent 1 vs Parent 3), indicating that hierarchical information related to a specific phenomenon is concentrated within particular layer ranges rather than being diffused uniquely for each level of the hierarchy.

Transformer encoders tend to *disperse* task-relevant signals across layers and heads,



(a) XLNet (large) - Raw Features



(b) XLNet (large) - Fine-Tuned

Figure 3.4: Linear classifier layer-wise performances for the Ancestor Sentiment Classification tasks for (a) XLNet (large) using raw features and (b) XLNet (large) after fine-tuning.

with progressively higher-level abstractions appearing in later layers (Tenney et al., 2019b). This dispersion brings robustness: many attention heads can be pruned with little or no loss, yet also redundancy (Michel et al., 2019). On the downside, it complicates analysis: a single layer's representations may not be linearly separable for a given property, so probe results depend on probe capacity. Practically, although lower layers encode broadly useful linguistic features, fine-tuning

Pre-trained Transformer embedder	SST-5 Root	
	Linear	Non-linear
BERT (base)	39.04	40.21
GPT2 (base)	37.61	37.80
Reformer (base)	41.70	43.09
XLNet (base)	42.45	**44.30**
BERT (large)	39.06	42.40
GPT2 (large)	37.61	38.33
Reformer (large)	41.11	44.48
XLNet (large)	43.73	**47.10**
GloVe (non-contextual)	28.81	29.22
State-of-the-art (all tokens)	-	54.70

Table 3.3: Performance (accuracy %) comparison of linear and non-linear probing classifiers on the SST-5 root classification task for all contextualisers, including a non-contextual baseline, and a state-of-the-art model utilising all input tokens as opposed to a single embedding. Best results in each group are in bold.

(or lightweight adapter tuning) is often required to make task-relevant information accessible to simple heads. Fine-tuning reconfigures upper layers for the target task, and adapter-style updates can tap information across the stack (Merchant et al., 2020; Peters et al., 2019).

3.4.2 Fine-tuning Comparisons

To assess how the fine-tuning process affects layer-wise representation of hierarchical sentiment, we fine-tune each model on root-level sentiment classification using the same SST dataset as described before. Results are shown in the bottom-half of Table 3.1. We observe similar relative performance trends as before, with Reformer and XLNet significantly outperforming BERT and GPT-2 on this hierarchical task when probed layer-wise post fine-tuning. Interestingly, we note that the accuracy achieved by the best performing layer of the fine-tuned XLNet model on root-level sentiment classification is competitive with a fine-tuned BERT model that utilises all input tokens.

Finally, we compare the layer-wise performances before and after fine-tuning, as shown in Figures 3.3(b) and 3.4(b) for the large Reformer and XLNet models. For XLNet and Reformer models, the fine-tuning process clearly improves the root-

level sentiment classification performances from the middle to end layers of the Transformer, showing that information gained from fine-tuning is propagated further down the layer stack when compared to the other architectures. This also comes at no cost to the performance on lower-level (leaf or shallow parent) tasks.

3.4.3 Non-linear Classifiers

Results for the best performing layers of the non-linear classifiers are reported in Table 3.3. The larger models appear to benefit more from including non-linearities in the classification layer for root-level sentiment prediction, relative to the linear results, suggesting that some hierarchical information might be encoded in a non-linearly separable manner. A likely reason is that, as model scale and paramaterisation grows, models increasingly represent information in distributed/superposed features, so the useful signals exist but are not necessarily aligned to a single linear direction in the embedding space, and non-linear heads can combine interacting features to recover it whereas linear probes cannot.

3.5 Conclusion

The experiments presented in this chapter show that a single contextualised word representation is capable of encoding significant information useful for classifying sentiment and constituent tags at multiple levels of the syntactic and semantic hierarchy of the sentence it is contextualised on. Linear probing results for ancestor sentiment analysis and constituency tagging using a range of pre-trained Transformer architectures show that hierarchical sentence information is often found within specific layer ranges for each task, with lower layers typically retaining more local word-level information. Additionally, we compare performance distributions of each Transformer and find that the XLNet and Reformer architectures often exhibit much flatter distributions when compared to BERT and GPT-2. These architectures also yield bigger gains from fine-tuning across all layers in terms of

representing hierarchical information, suggesting such information can more easily propagate to their earlier layers.

In Transformer encoders, task-relevant signals are often *dispersed* across layers and heads rather than localized in a single point, with different layers capturing complementary abstractions (Tenney et al., 2019b; Liu et al., 2019). This dispersion brings upsides: robustness and transferability, since redundant cues make models resilient to ablations (e.g., many attention heads can be pruned with little loss) and provide multiple access points for downstream tasks (Michel et al., 2019; Voita et al., 2019) However, it also complicates interpretation and intervention: useful information may not be linearly separable at any one layer, and higher-capacity probes risk learning the task themselves, motivating careful probe design and complexity controls.

Finally, we show there is enough global hierarchical information encoded in a single representation of a fine-tuned XLNet to achieve 53.51% accuracy on 5-class sentence-level sentiment analysis; comparable with a fine-tuned BERT model utilising the entire input context. These findings help further our understanding of how and where hierarchical structures are internalised within these models, helping to address research question 1.

Future work could aim to probe a wider range of pre-trained contextualisers and NLP classification tasks typically solved by sequential classifiers utilising the full sequence, where single tokens can be tasked to predict labels at varying constituent levels or hierarchy classes of the sequence. Additionally, we aim to explore how capable each layer is at capturing the presence of linguistic categories in the input sample such as age, gender, region, dialect, etc., and how this varies between architectures.

3.6 Epilogue

This chapter has demonstrated that rich hierarchical information, encompassing both syntactic and semantic structures, is robustly encoded and recoverable from

token-level representations within diverse Transformer architectures. Our core findings reveal that the internalisation of this hierarchy is a layer-dependent phenomenon, systematically shaped by architectural choices (such as bidirectionality versus autoregression) and further refined by task-specific fine-tuning. These results provide crucial insights into *how and where* foundation models learn to represent linguistic structure, directly addressing a key component of our first research question.

The developed ancestor probing framework and the specific patterns of hierarchical encoding uncovered here serve as an essential foundation for the remainder of this dissertation. Understanding that detailed structural information resides within individual embeddings informs our subsequent investigation in Chapter 4 into the interplay between such learning and memorisation (related to research question 2). Moreover, these insights into analysing internalised hierarchy are adapted and extended to explore spatiotemporal reasoning in video Transformers (Chapter 5) and to guide the development of a unified multimodal framework (Chapter 6, relevant to research question 3), marking this chapter as a first step in our broader inquiry into hierarchical reasoning in contemporary AI.

Bibliography

Coenen, A., Reif, E., Yuan, A., Kim, B., Pearce, A., Viégas, F., and Wattenberg, M. (2019). Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*.

Dalvi, F., Durrani, N., Sajjad, H., Belinkov, Y., Bau, A., and Glass, J. (2019a). What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proc. of AAAI*.

Dalvi, F., Nortonsmith, A., Bau, A., Belinkov, Y., Sajjad, H., Durrani, N., and Glass, J. (2019b). Neurox: A toolkit for analyzing individual neurons in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9851–9852.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, pages 4171–4186.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proc. of ACL*, pages 328–339.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kitaev, N., Łukasz Kaiser, and Levskaya, A. (2020). Reformer: The efficient transformer.

Klein, D. and Manning, C. D. (2003a). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

Klein, D. and Manning, C. D. (2003b). Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019). Linguistic knowledge and transferability of contextual representations. In *Proc. of NAACL*, pages 1073–1094.

Merchant, A., Rahimtoroghi, E., Pavlick, E., and Tenney, I. (2020). What happens to BERT embeddings during fine-tuning? In Alishahi, A., Belinkov, Y., Chrupała, G., Hupkes, D., Pinter, Y., and Sajjad, H., editors, *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? In *Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peters, M., Neumann, M., Zettlemoyer, L., and Yih, W.-t. (2018a). Dissecting contextual word embeddings: Architecture and representation. In *Proc. of EMNLP*, pages 1499–1509.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018b). Deep contextualized word representations. In *Proc. of NAACL*.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In Augenstein, I., Gella, S., Ruder, S., Kann, K., Can, B., Welbl, J., Conneau, A., Ren, X., and Rei, M., editors, *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2021). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*, pages 1631–1642.

Tenney, I., Das, D., and Pavlick, E. (2019a). Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Tenney, I., Das, D., and Pavlick, E. (2019b). BERT rediscovers the classical NLP pipeline. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., and Pavlick, E. (2019c). What do you learn from context? probing for sentence structure in contextualized word representations. In *Proc. of ICLR*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting. In *ACL 2019*, pages 5797–5808.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2020). Xlnet: Generalized autoregressive pretraining for language understanding.

Chapter 4

Hierarchy in Language Model Memorisation

While LLMs are designed and optimised to generalise to unseen data, they also exhibit a tendency to memorise training data, a behaviour that can be significantly amplified during fine-tuning and poses considerable risks regarding data privacy and model trustworthiness. Building upon the findings in Chapter 3 which revealed how Transformers internalise rich hierarchical structures, this chapter addresses the second research question of our thesis: investigating the critical interplay between this internalisation of hierarchy and the onset of unwanted verbatim memorisation. Specifically, we explore the dynamics of verbatim memorisation during common fine-tuning regimes and develop practical, scalable methods to detect and mitigate such undesirable storage, thereby promoting a better balance between useful structural learning and harmful data leakage. Our key findings and proposed mitigation techniques are detailed within.

The chapter is structured as follows. Section 4.1 further introduces the challenge of memorisation in fine-tuned LLMs and outlines this chapter's specific contributions to addressing it. We then review relevant literature not covered in Chapter 2 on measuring, characterising, and mitigating memorisation in Section 4.2. Section 4.3 describes our experimental methodology, including the metrics for verbatim and n-gram memorisation, the diverse datasets and models employed, and our fine-tuning

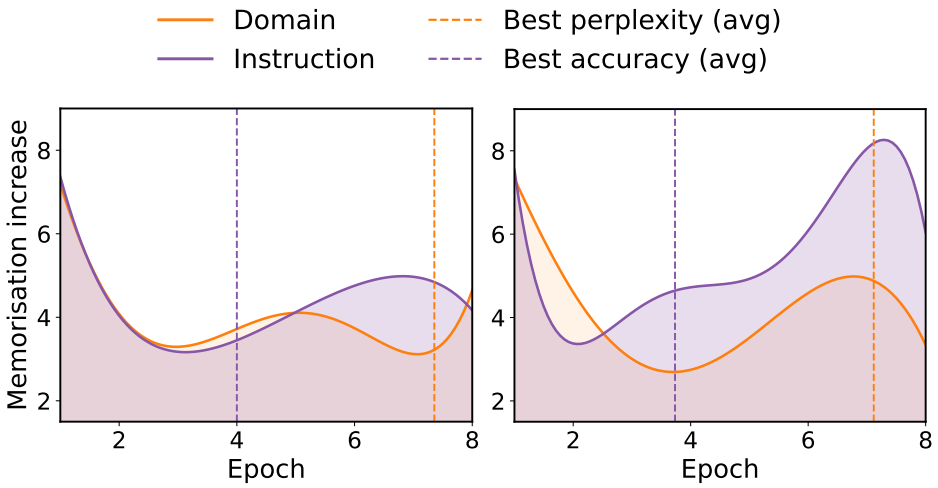
protocols. The core empirical findings, focusing on the predictive utility of n-gram overlap and a comparative analysis of our mitigation techniques, are presented and discussed in Section 4.4. Finally, Section 4.5 acknowledges the limitations of this study.

4.1 Introduction

LLMs have become increasingly powerful, achieving remarkable performance across diverse tasks and domains as they scale from millions to trillions of parameters (Brown et al., 2020b; Fedus et al., 2022). Transformer-based architectures have propelled significant advancements in NLP, setting new benchmarks in various applications (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020a). However, alongside these achievements, concerns have emerged about the extent to which these models memorise their training data rather than genuinely understanding and generalising underlying hierarchical linguistic patterns (Khandelwal et al., 2019; Tänzer et al., 2021).

Memorisation in LLMs poses serious privacy and security risks. Models have been shown to reproduce verbatim passages from their training data, including sensitive personal information and copyrighted material (Patil et al., 2024). This not only presents ethical challenges and potential legal issues but can also undermine user consent when deploying models in a generative environment. Training data extraction attacks (Carlini et al., 2021) demonstrate that adversaries can recover spans of pre-training sample data, highlighting the practical threat of generative model deployment.

Most existing mitigation efforts focus on unlearning strategies and regularisation techniques applied during pre-training (Cheng et al., 2021; Carlini et al., 2023). While valuable, these approaches often lack scalability and are not easily deployable in practice, especially given the immense computational resources required to retrain large models or apply differential privacy methods (Anil et al., 2022). Moreover, on large datasets, exhaustive extraction tests are infeasible, making it



(a) Pythia 1.4B (b) Pythia 12B

Figure 4.1: Memorisation increases (number of new samples memorised at a given epoch) at successive fine-tuning epochs, comparing fine-tuning for domain adaptation (orange) and instruction tuning (purple) on the same data. Dashed vertical lines mark the average epoch for which validation perplexity (orange) and task evaluation accuracy (purple) are achieved, showing high memorisation before for both (a) Pythia 1.4B and (b) Pythia 12B models.

challenging to assess and mitigate memorisation effectively. Fine-tuning pre-trained LLMs on domain-specific and instruction-specific data is a common practice to adapt models to new domains and tasks, often utilising datasets with private and sensitive information. Despite this widespread application, there is a gap in understanding how fine-tuning for domain adaptation or instruction tuning impacts memorisation dynamics.

Our preliminary observations, illustrated in Fig. 4.1, show significant memorisation occurring early during fine-tuning, often before the model achieves optimal validation perplexity or task evaluation performance. This suggests that LLMs rapidly memorise new information before reaching typical early stopping criteria, potentially exposing sensitive information. Owing to this, this chapter presents our empirical investigation into memorisation in LLMs during fine-tuning. We focus on fast, deployable mitigation strategies and insights applicable during both domain adaptation and instruction tuning, leveraging widely used memorisation metrics. We perform fine-tuning experiments using the *Pythia* model family (Biderman et al., 2023) across multiple parameter scales (1.4B - 12B), as well as *Llama*

2 7B (Touvron et al., 2023), *Llama 3* 8B and 70B (Grattafiori et al., 2024), and *Mistral* 7B (Jiang et al., 2023) models for both domain adaptation and instruction tuning across a range of common text-generative LLM evaluation datasets.

Our key contributions, which directly address the challenge of understanding and mitigating memorisation in the context of fine-tuning, are:

- **Understanding memorisation dynamics to safeguard structural learning**: We examine how verbatim memorisation manifests during common fine-tuning paradigms (domain adaptation and instruction tuning). This analysis offers crucial insights into the conditions under which the generalisation of learned hierarchical structures, a key focus of this thesis, may be undermined by verbatim memorisation.
- **Early detection of verbatim memorisation to preserve hierarchical understanding**: We establish that an n-gram based partial memorisation metric serves as a robust early indicator of verbatim data leakage. This provides a critical tool for interventions aimed at preserving the integrity of learned hierarchical patterns over superficial memorisation.
- **Optimal stopping criteria to favour structural generalisation**: Leveraging our n-gram metric, we identify optimal stopping criteria during fine-tuning that significantly reduce verbatim memorisation. This approach helps to preserve robust task performance by promoting the model's reliance on generalised hierarchical learning.
- **Scalable mitigation for enhanced structural generalisation**: We introduce and evaluate an n-gram aware loss regulariser, demonstrating its capacity to achieve scalable and generalisable reductions in verbatim memorisation. This technique supports the development of models that maintain strong task performance based on generalised hierarchical understanding, rather than reliance on memorised training instances, helping promote safer AI systems.

4.2 Related Work

Prior research on memorisation in LLMs spans three main areas: measurement, characterisation across pre-training versus fine-tuning, and mitigation, each covered in the following sub-sections.

4.2.1 Measuring Memorisation

Evaluating the extent of memorisation in LLMs necessitates robust metrics and evaluation techniques. Carlini et al. (2023) introduce the concept of *k-extractable memorisation*, which measures a model's tendency to reproduce training data when provided with specific input prefixes, representing a stringent test for data leakage. Complementary approaches include membership inference attacks aimed at classifying pre-training samples (Shokri et al., 2017). Memorisation and generalisation have been shown to carry interdependent relationships (Yeom et al., 2018; Khandelwal et al., 2019; Tänzer et al., 2021), with memorisation dynamics in large-scale LLMs studied in Tirumala et al. (2022); Carlini et al. (2023).

4.2.2 Memorisation in Pre-training Versus Fine-tuning

The dynamics of memorisation exhibit distinct characteristics during the pre-training and fine-tuning stages of LLM development. In the pre-training phase, models are exposed to extensive and often publicly available datasets, where factors such as data redundancy and model size play critical roles in determining the extent of memorisation (Khandelwal et al., 2019; Tänzer et al., 2021; Carlini et al., 2023). Research indicates that larger models are more prone to rapidly memorising training data (Tirumala et al., 2022; Nasr et al., 2023). Conversely, during fine-tuning on specialised or private datasets, different memorisation risks emerge. Studies have demonstrated that specific fine-tuning methodologies, like adapter-based techniques, can reduce the likelihood of memorising sensitive information (Raffel et al., 2020; Dodge et al., 2021; Mireshghallah et al., 2022). Additionally,

Category	Dataset	Domain	Input type	Output type
Classification	SST-5	Movie reviews	Single review sentence	5-way sentiment
	QQP	Quora community QA	Question_1, Question_2	Duplicate? (yes/no)
	RTE	News & Wikipedia	Premise, Hypothesis	Entail / Not-entail
	WANLI	MultiNLI-derived genres	Premise, Hypothesis	Entail / Neutral / Contradict
QA	SQuAD v2	Wikipedia articles	Paragraph context, Question	Answer span or `[NoAnswer]`
	HellaSwag	WikiHow narratives	Context + 4 candidate endings	Correct ending (MC-4)
	PubMedQA	Biomedical abstracts	Abstract (no conclusion), Question	Yes / No / Maybe
Summarisation	XSum	BBC news	Full news article	One-sentence summary
	CNN/DailyMail	CNN & Daily Mail news	Full news article	Multi-sentence highlights
Instruction	Alpaca	Mixed user prompts	Instruction (± optional input)	Free-form response
	FLAN v2	Multi-domain tasks	Instruction template	Free-form response

Table 4.1: Summary and grouping of datasets used for fine-tuning.

counterfactual memorisation assessments (Zhang et al., 2021) aid in distinguishing between memorisation arising from pre-training and that from fine-tuning, thereby informing targeted mitigation strategies tailored to each training phase.

4.2.3 Mitigation Strategies and Regularisation

During the training process, regularisation methods such as the addition of noise to input embeddings (Miyato et al., 2017) are employed to mitigate memorisation (Feldman and Zhang, 2020; Tirumala et al., 2022). Post-training techniques include fine-tuning and machine unlearning approaches (Maini et al., 2023), which aim to remove specific data from the model without necessitating a complete retraining. Despite these measures, achieving a balance between preserving model performance and ensuring data privacy remains a significant challenge. Mitigating memorisation in language models is critical for preserving privacy and preventing the leakage of sensitive information. Conventional regularisation techniques, such as weight decay and dropout, are designed to prevent overfitting and thereby reduce memorisation (Feldman and Zhang, 2020). However, these methods have proven inadequate in fully reducing memorisation within LLMs (Tirumala et al., 2022). Advanced regularisation approaches, including data-dependent token dropout (Hans et al., 2024) and targeted token masking (Jain et al., 2024), offer partial mitigation but often fail to eliminate the risk of memorising entire data passages, especially when dealing with highly duplicated datasets.

4.3 Methodology

We begin by defining how we measure memorisation, leveraging an existing approach and introducing a partial measure for more fine-grained analysis. We follow this by introducing the experimental setup for our study of fine-tuning for domain adaptation and instruction tuning.

4.3.1 Memorisation Metrics

For an exact and scalable measure of verbatim memorisation, we employ the widely used extraction metric introduced in Carlini et al. (2023).

Memorisation: Let f be a generative LLM trained on data D, with prefix-suffix pair (p, s) contained within a sample in D. A suffix s is said to be *k-extractable* (*memorised*) if f generates a string containing s exactly when prompted with a prefix of length k using greedy decoding.

Therefore we can compute the percentage of the fine-tuning data memorised at each fine-tuning epoch as:

$$\text{Mem} = \left(\frac{\text{number of extractable suffixes } s}{\text{total samples in data } D} \right) \times 100. \tag{4.1}$$

This definition provides a directly computable metric on the generated output from our fine-tuned models, allowing fast evaluation at each fine-tuning epoch. We use the above as the definition for *memorisation* throughout this chapter.

4.3.2 n-gram Memorisation

For a fine-grained measure of memorisation, we implement a partial memorisation metric based on n-gram overlap.

n-gram Memorisation: For a set of n-gram sizes $N = \{n_1, n_2, \ldots, n_k\}$, the n-gram memorisation score between the model's output $f(p)$ and the target sequence

s is defined as the proportion of matching n-grams of sizes in N. The matches are exact for each n-gram, but the score is invariant to the ordering of n-grams within the sequences.

Formally, given M_i as the fraction of matching n-grams from the set N between $f(p_i)$ and s_i for a given sample i, the n-gram memorisation score for the dataset D is then calculated as:

$$n\text{-gram Mem} = \left(\frac{\sum_{d \in D} M_d}{|D|} \right) \times 100. \tag{4.2}$$

This metric provides a finer-grained measure of partial memorisation that allows for different lengths and numbers of n-grams, which can be tuned for suitability for specific datasets, sequence lengths, and the granularity of sensitive information. Because the score sums exact n-gram matches over all positions, the same n-gram can be counted multiple times via overlapping windows; consequently, shorter n-grams, which occur far more frequently carry greater effective weight, and potentially provide an earlier, more sensitive memorisation signal.

4.3.3 Datasets

We leverage datasets taken from three open instruction pools: the Public Pool of Prompts (P3) (Sanh et al., 2021), the FLAN collection (Wei et al., 2023), and the Alpaca-52K corpus (Taori et al., 2023). We conduct both instruction tuning and domain adaptation experiments by choosing to include or remove the task-specific instruction prompt for each dataset. These datasets encompass a range of core NLP task types: classification, Natural Language Inference (NLI), coreference resolution, Question-Answering (QA), and free-form instruction following, and span diverse domains such as encyclopedic text, news, clinical notes, biomedical research, and social media content. A summary of all datasets used is outlined in Table 4.1, with further details in Appendix 4.8. We categorise them into the following:

- **Classification & NLI**: short, label-based prompts (sentiment, paraphrase, entailment).

- **Question-Answering**: a mix of extractive, multiple-choice, and yes/no items.
- **Summarisation**: single-sentence and multi-sentence summaries.
- **Instruction Following**: open-ended prompts from Alpaca and FLAN tasks.

These datasets are chosen to provide task and domain diversity for evaluating how this impacts memorisation, as well as providing datasets which can be used for both domain adaptation and instruction tuning.

4.3.4 Pre-trained Models

Experiments are run on the *Pythia* model family (Biderman et al., 2023) using sizes of 1.4B, 2.8B, 6.9B, and 12B parameters. The Pythia suite offers a controlled setting where pre-training hyper-parameters and dataset composition are kept fixed while model size is systematically varied, providing a clean scaling ladder for evaluation. Additionally, we use *Llama 2* 7B (Touvron et al., 2023), *Llama 3* 8B and 70B (Grattafiori et al., 2024), and the *Mistral 7B* model (Jiang et al., 2023). These models are chosen to enable comparisons between architectural variants at similar model sizes. All pre-trained model checkpoints are publicly accessible via HuggingFace (Wolf et al., 2019). Fine-tuning is performed using the Adam optimiser (Kingma and Ba, 2014). We perform full-parameter fine-tuning and, for comparison, conduct *partial fine-tuning* in which only the top n Transformer layers are updated while the rest remain frozen, enabling us to measure how restricting the trainable subset of parameters alters memorisation behaviour.

4.3.5 Fine-Tuning Approach

We employ domain adaptation and instruction tuning by fine-tuning each model for up to 8 epochs on a maximum of 5,000 samples from the target dataset. When performing domain adaptation, we simply remove the task-specific instructions

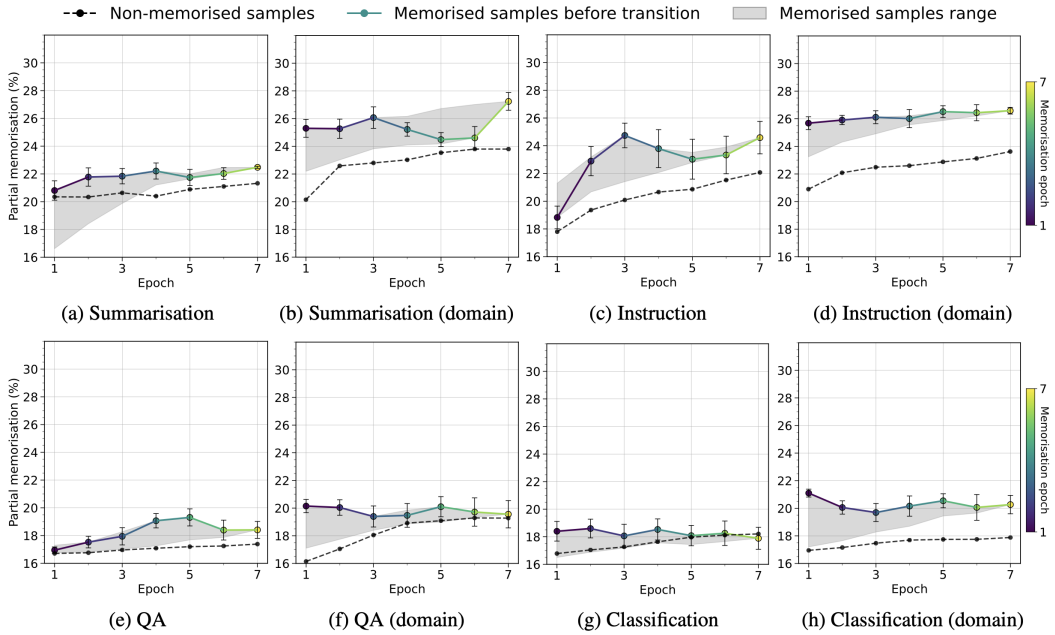


Figure 4.2: Partial n-gram memorisation across fine-tuning epochs for the four dataset categories, with *domain* indicating domain adaptation fine-tuning. In each panel, the coloured solid line reports, at epoch t, the *median* score of samples that become memorised at subsequent epoch $t+1$; the point colour encoding that memorisation epoch. The grey shaded region spans the full score range of *all* samples that ever become memorised, irrespective of when the transition occurs. Error bars show the standard deviation over five random seeds, while the black dashed line is the baseline for samples that are never memorised. Results are averages over Pythia model sizes from 1.4B to 12B parameters.

from the input. We evaluate on a held-out validation set for both *validation perplexity* and task-specific *evaluation performance*. For evaluation performance, we use the standard evaluation metrics for each task (details can be found in Appendix 4.8). For our memorisation and n-gram memorisation metrics, we evaluate on the 5,000 samples of training data used for fine-tuning. The small number of fine-tuning samples allows us to rapidly experiment over model scales and datasets while remaining relevant to typical scenarios involving small and potentially private fine-tuning datasets. Evaluations are performed at each epoch to monitor the progression of memorisation relative to validation performance and evaluation performance.

We test k-extractable memorisation with three prefix lengths, $k \in \{12, 16, 20\}$, and a fixed 20-token suffix. We diverge from Carlini et al. (2023) by using smaller prefix lengths due to the following rationale: lengths below 12 tokens collide frequently

across corpora, whereas prefixes longer than 20 tokens vastly limit the number of samples we can use from each of the datasets, as they are selected for fine-tuning and not language model pre-training. We empirically find that 4, 5, and 6-grams for our n-gram memorisation metric provide a good signal for highly memorised phrases without being computationally prohibitive. All results are averaged over 10 runs with random seed initialisations. For robustness, we use different randomly sampled prefix-suffix pairs for each of the 10 randomly initialised fine-tuning runs.

4.4 Results and Discussion

We begin by evaluating n-gram memorisation results over model scales and domains. Subsequently, we discuss epoch selection criteria for minimising memorisation and their performance trade-offs. Finally, we compare mitigation strategies across model scales.

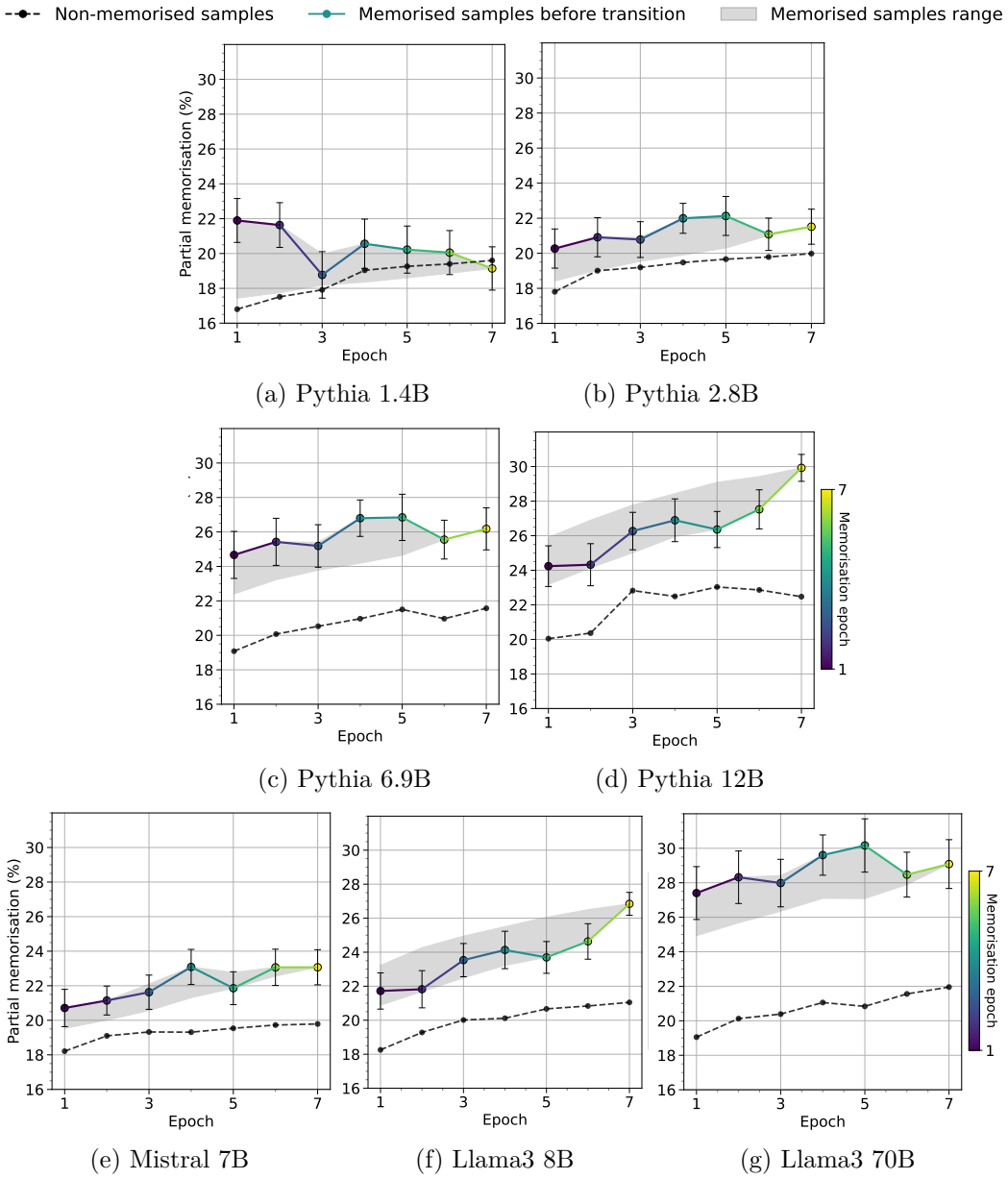
4.4.1 n-gram Memorisation Predicts Verbatim Memorisation

Driven by the observation shown in Fig. 4.1 that high-rate memorisation occurs in the early epochs preceding optimal stopping criteria for both validation perplexity and task evaluation performance, we investigate n-gram memorisation values as a proxy for fine-grained memorisation. To correctly identify early warning signs of samples at high risk of verbatim memorisation, we evaluate n-gram memorisation after each fine-tuning epoch. Fig. 4.2 shows our results for this evaluation on each of the dataset categories outlined in Table 4.1, with *domain* indicating domain-adaptation fine-tuning. For all samples that are identified as memorised during 8 fine-tuning epochs, we track their associated n-gram memorisation score on the epochs preceding the transition to verbatim memorisation. This allows us to understand if the partial memorisation score is higher in the epoch preceding a transition to verbatim memorisation, relative to non-memorised phrases. For this, we plot the average n-gram memorisation for non-memorised phrases throughout fine-tuning as a baseline.

For each of the dataset categories visualised, we observe a clear distinction in partial memorisation between the memorised and non-memorised samples, with the majority of epochs scoring markedly higher than the baseline for non-memorised samples. We see the degree to which this is higher varies significantly between domains, with the largest discrepancy observed in Instruction following and Summarisation. The News domains used in the summarisation tasks tend to include high-frequency stock phrases; as such, these datasets are known to encourage extractive copying (Tejaswin et al., 2021), with which our results concur. Most notably, we find that for all datasets, the domain adaptation version sees a significant increase in partial memorisation over the baseline, whereas the baseline scores do not change significantly. Interestingly, there is a large increase in partial memorisation scores of samples which are memorised in the early epochs when performing domain adaptation.

We perform the same evaluation but compare model size and architecture, shown in Fig. 4.3. We identify the expected trend that larger model sizes correlate to higher memorisation capacity, which is reflected in the partial memorisation score increase across the Pythia models. The partial memorisation score gap between memorised and non-memorised samples increases significantly with increasing model size, showing a strong indicator that this metric serves as a scalable precursor to verbatim memorisation. An unexpected result is that for the smaller 1.4B model, partial memorisation decreases for samples memorised in the latter epochs of fine-tuning; a trend which does not follow for the larger model sizes. Comparing different architectures, we find similar gaps to baseline and the same trend of increasing partial memorisation gap to baseline over fine-tuning epochs.

Figure 4.4 repeats the analysis for Llama3 8B when only the top n Transformer layers are updated. The non-memorised baseline is unaffected, but unfreezing more layers suppresses partial memorisation in the first few epochs and heightens it in later epochs. This is consistent with a capacity-bottleneck view in which extra trainable layers delay, yet ultimately amplify, overfitting during fine-tuning.



(a) Pythia 1.4B (b) Pythia 2.8B

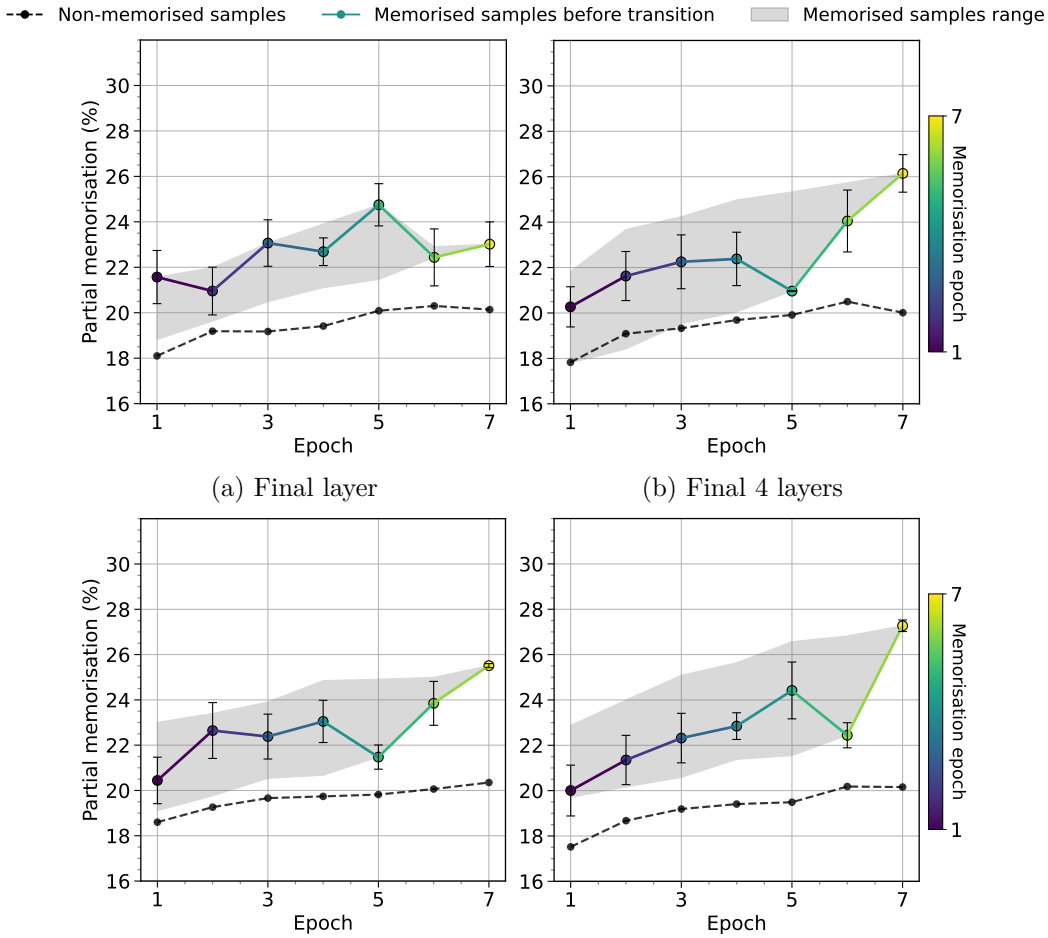
(c) Pythia 6.9B (d) Pythia 12B

(e) Mistral 7B (f) Llama3 8B (g) Llama3 70B

Figure 4.3: Partial memorisation across epochs for different models. The coloured solid line gives the median score of samples that will be memorised at epoch $t+1$; point colour marks that future epoch. The grey region shows the full range for all eventually memorised samples, while the black dashed line is the baseline for samples never memorised. Error bars denote the standard deviation across five random seeds.

4.4.2 Selection Criteria as Mitigation

Following our findings that high-rate memorisation occurs before optimal validation perplexity or task evaluation performance, and that partial memorisation serves as a potential precursor to memorisation, we now investigate the efficacy of



(a) Final layer

(b) Final 4 layers

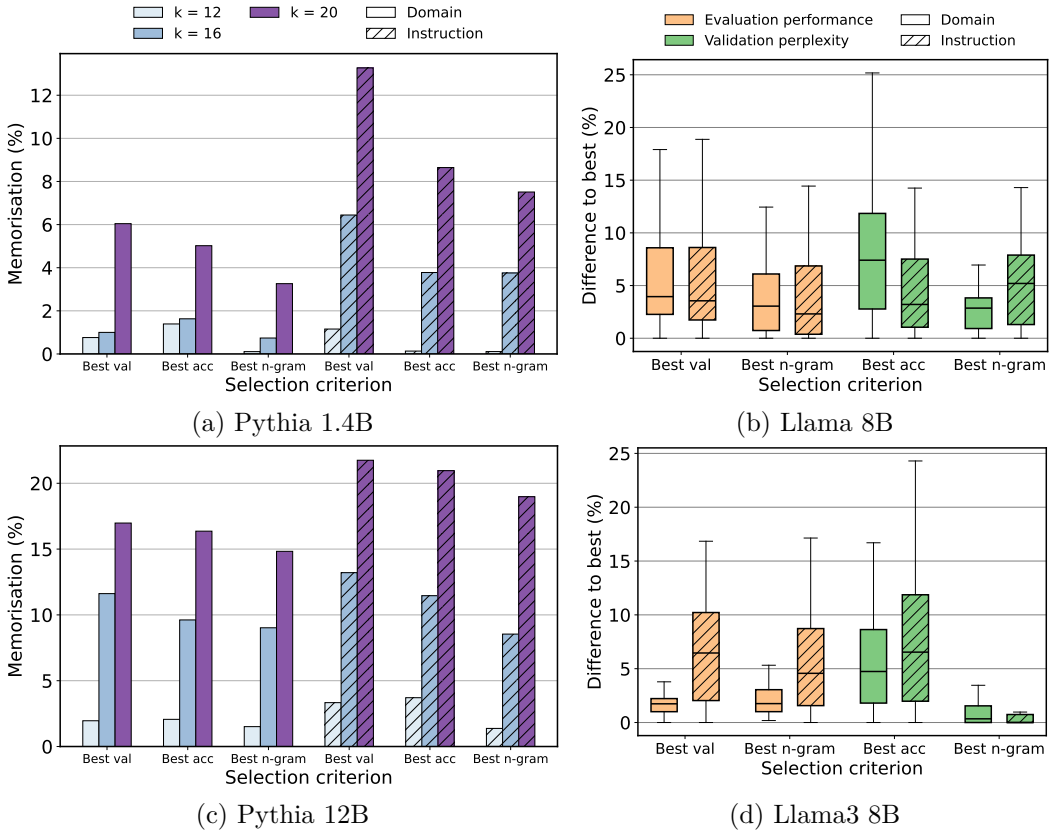
(c) Final 8 layers

(d) Final 16 layers

Figure 4.4: Final-layer partial fine-tuning comparison of the Llama3 8B model. The final n layers of the model are unfrozen and updated when fine-tuning, with the remaining layers frozen.

utilising this as an early stopping criterion. Without resorting to regularisation or unlearning strategies, we explore using n-gram memorisation as a threshold for early stopping. To adapt n-gram memorisation as an early stopping criterion, we test different threshold values for which to stop fine-tuning if exceeded. We find that an average partial memorisation threshold score of 20 on the fine-tuning set yields good results. We compare this to the naive selection criterion of validation perplexity and task evaluation for domain adaptation and instruction tuning, respectively, although we experiment with applying validation perplexity and best accuracy to both.

Results for these experiments are shown in Fig. 4.5, highlighting the trade-offs between different early stopping criteria and their impact on both memorisation



(a) Pythia 1.4B

(b) Llama 8B

(c) Pythia 12B

(d) Llama3 8B

Figure 4.5: Memorisation and performance comparison for domain adaptation and instruction tuning across different early stopping selection criteria. (a) and (c) show the verbatim memorisation percentage for different values of extraction prompt prefix length $k \in \{12, 16, 20\}$ using three early stopping selection criteria: validation perplexity (Best val), evaluation performance (Best acc), and n-gram memorisation (Best n-gram) for domain adaptation (solid) and instruction tuning (hatched). (b) and (d) present the difference to the best task evaluation performance (orange) and validation perplexity (green), across the same selection criteria and fine-tuning approaches.

and model performance. Using evaluation performance/accuracy as the selection criterion consistently reduces memorisation rates in both domain adaptation and instruction tuning scenarios (Fig. 4.5(a) and Fig. 4.5(c)). This could be due to task evaluation performance correlating more highly with the latent capabilities of the pre-trained model, rather than validation perplexity on a single domain, and therefore being optimised at lower memorisation. However, this comes at the cost of a significant decrease in validation perplexity, as indicated by the high variance and larger differences to the best perplexity scores shown in Fig. 4.5(b) and Fig. 4.5(d). Conversely, when validation perplexity is used as the selection criterion, the models tend to show the opposite behaviour by achieving better perplexity scores, but with

	QA		Summarisation		Instruction		Average	
Model + Strategy	Mem ↓	Eval ↓	Mem ↓	Eval ↓	Mem ↓	Eval ↓	Mem ↓	Eval ↓
Pythia 2.8B	9.71	-	12.59	-	14.30	-	12.20 ± 0.65	-
+ Best n-gram	4.36	7.51	5.63	6.53	6.43	8.09	5.47 ± 0.38	7.38 ± 0.28
+ n-gram reg	**2.90**	5.08	**3.75**	**4.25**	**4.29**	6.54	$\mathbf{3.65 \pm 0.29}$	$\mathbf{5.29 \pm 0.24}$
+ Goldfish reg	3.37	**5.04**	4.38	4.31	5.01	**6.07**	4.25 ± 0.22	5.14 ± 0.22
Pythia 6.9B	12.80	-	16.50	-	18.70	-	16.00 ± 0.77	-
+ Best n-gram	5.76	7.54	7.42	6.21	8.42	8.30	7.20 ± 0.38	7.35 ± 0.30
+ n-gram reg	**3.84**	5.15	**4.95**	**4.54**	5.61	**6.30**	$\mathbf{4.80 \pm 0.24}$	$\mathbf{5.33 \pm 0.23}$
+ Goldfish reg	4.48	**5.07**	5.77	4.83	**5.55**	6.59	5.27 ± 0.20	5.50 ± 0.25
Mistral 7B	13.65	-	17.50	-	19.88	-	17.01 ± 0.95	-
+ Best n-gram	6.12	7.55	7.88	6.00	8.91	8.89	7.64 ± 0.41	7.48 ± 0.29
+ n-gram reg	4.18	5.53	5.25	**4.40**	5.34	**6.08**	4.92 ± 0.19	5.34 ± 0.22
+ Goldfish reg	**4.01**	**5.40**	**5.12**	4.42	**4.97**	6.21	$\mathbf{4.70 \pm 0.21}$	$\mathbf{5.34 \pm 0.22}$
LLaMA3 8B	14.40	-	18.50	-	20.94	-	17.95 ± 0.84	-
+ Best n-gram	6.48	9.21	8.33	6.56	9.41	10.31	8.07 ± 0.47	8.69 ± 0.35
+ n-gram reg	**4.32**	**4.38**	**5.55**	**3.81**	**6.27**	**5.32**	$\mathbf{5.38 \pm 0.23}$	$\mathbf{4.50 \pm 0.20}$
+ Goldfish reg	5.04	5.02	6.47	4.33	7.32	6.99	6.28 ± 0.15	5.45 ± 0.24
Pythia 12B	17.66	-	22.50	-	25.30	-	21.82 ± 1.05	-
+ Best n-gram	7.92	9.20	10.12	6.41	11.39	8.30	9.81 ± 0.45	7.97 ± 0.36
+ n-gram reg	**5.28**	3.98	**6.75**	**4.02**	**7.59**	**4.91**	$\mathbf{6.54 \pm 0.27}$	$\mathbf{4.30 \pm 0.21}$
+ Goldfish reg	6.10	**3.90**	7.57	4.36	8.86	5.00	7.51 ± 0.14	4.42 ± 0.22
LLaMA3 70B	20.80	-	26.50	-	29.70	-	25.67 ± 1.11	-
+ Best n-gram	9.36	9.39	11.93	5.96	13.37	8.45	11.55 ± 0.56	7.93 ± 0.38
+ n-gram reg	**6.24**	5.54	**7.05**	**3.91**	**8.91**	**5.44**	$\mathbf{7.40 \pm 0.30}$	$\mathbf{4.96 \pm 0.23}$
+ Goldfish reg	7.18	**5.50**	7.27	4.01	10.40	6.11	8.28 ± 0.16	5.21 ± 0.24

Table 4.2: Main memorisation mitigation results across model scales and mitigation strategies. For each result we report the memorisation (*Mem*, lower is better), and Evaluation difference (*Eval*, lower is better) to the best performance achieved for the naive unmitigated strategy (top row of each model group). Bold values indicate the best (lowest) score within each model group (base row excluded). Memorisation scores are taken as the average of all prefix lengths $k \in \{12, 16, 20\}$ extractions. Results are averages over 10 randomly initialised fine-tuning runs, and we report mean $\pm$ s.d. in the Average columns.

substantially higher memorisation rates, particularly for instruction-tuned models which consistently exhibit the highest memorisation levels compared to domain adaptation results.

Interestingly, the n-gram selection criterion strikes a balance, reducing memorisation without the steep performance trade-offs observed in the other criteria. It provides a more favourable balance by keeping memorisation lower and maintaining better accuracy and perplexity than either of the naive criteria (evaluation accuracy or validation perplexity), as seen by the smaller performance differences at consistently lower memorisation percentages. In summary, instruction tuning appears more prone to memorisation, particularly under validation-based selection, whereas domain adaptation is relatively less affected by these selection criteria, and

n-gram thresholding as a stopping criterion is a simple and effective memorisation mitigation strategy.

4.4.3 Comparing Mitigation Strategies

We test whether our n-gram approach can be incorporated into a loss regularisation function by adapting the typical causal LLM loss to include a term that penalises high-confidence n-grams exceeding a tunable confidence threshold, relative to that of the pre-trained model. Intuitively, this penalty is designed to discourage the model from assigning excessively high probabilities to these n-grams as a proxy measure for n-gram memorisation. The key limitations of this strategy are in requiring the original model to run inference alongside fine-tuning to acquire the baseline confidence values, and keeping n-gram sizes within practical bounds to avoid becoming computationally intensive. Further details of this approach can be found in Appendix 4.9.

We compare our n-gram regularisation to the *Goldfish loss* regularisation technique (Hans et al., 2024), which incorporates random sampling of dropped tokens from the loss calculation for a given training sample. At the time of writing, this is the only comparable approach for which we can compare to. We test across all models to evaluate transferability and scalability of the approach.

We present our results in Table 4.2, grouped by model size and dataset category, including comparisons to naive baseline results for both domain adaption and instruction tuning (top row of each model group). We include the stopping criterion *Best n-gram* as a simple non-regularisation approach based on the promising findings in Section 4.4.2. We consider memorisation (Mem %) and evaluation performance (Eval %), where Eval is taken as the difference to the best achieved performance, essentially measuring the performance trade-off of the memorisation mitigation technique. We group our results by model size, and report the best (bold) within each group.

Impact of model size These results highlight key trends across different model scales and mitigation strategies. Generally, memorisation increases with model size, as observed with the unmitigated baseline for Pythia 2.8B of 12.2% rising to 21.8% for Pythia 12B and 25.7% for Llama3 70B. Importantly, the mitigation strategies show consistent reductions in memorisation across all models. For example, n-gram regularisation reduces memorisation from 12.2% to 3.6% in Pythia 2.8B, and from 21.8% to 6.5% in Pythia 12B. We see similar reductions in the Llama3 and Mistral models. Goldfish regularisation is also effective, though its impact is more pronounced on the Mistral 7B model, whereas our n-gram regularisation outperforms this on all other models. Across the board, larger models present greater challenges in balancing memorisation, validation perplexity, and accuracy. The results suggest that as model size increases, the trade-offs become more pronounced.

Impact of mitigation strategy Averaged over all models, *n-gram regularisation* delivers the best trade-off, lowering memorisation to 5.45% with a performance evaluation gap of 4.95%; this is a ≈40% relative reduction in memorisation and a ≈35% smaller performance hit compared with the simple *Best n-gram* early-stopping rule (8.29%, 7.80%). *Goldfish* is a close second (6.05%, 5.18%), performing best on Mistral 7B. While the early-stopping heuristic of Best n-gram consistently sees higher memorisation and worse evaluation performance, it still significantly reduces memorisation from the naive baseline, highlighting the importance of a simple non-regularisation approach.

4.5 Limitations

Our study provides insights into memorisation during domain adaptation and instruction tuning of generative LLMs, but has limitations. We focused on greedy decoding, while real-world applications often use more complex methods like beam search, which likely influence memorisation differently; future research should explore memorisation under various decoding strategies.

We used validation perplexity and evaluation performance as metrics, but their trade-offs with memorisation are not necessarily equivalent. Investigating alternative metrics could offer a more nuanced understanding of these relationships. Our experiments were limited to a single high-parameter model (Llama3 70B) due to computational budget limitations; ideally, we would evaluate these findings on a larger pool of models and sizes, as well as different fine-tuning protocols.

4.6 Conclusion and Future Work

This study explores memorisation dynamics during both domain adaptation and instruction tuning across eight open-weight LLMs (1.4B–70B parameters). We show that a simple n-gram partial memorisation score indicates at-risk samples. The gap between memorised and non-memorised items is widest in domain adaptation and summarisation datasets, reflecting repetition and lack of diversity often seen with instruction-tuning, whereas classification and QA tasks exhibit a smaller, but still measurable, rise. We also show that our partial memorisation metric scales very well with increasing model size, where memorisation is more pronounced.

Building on these observations, we explore memorisation mitigation strategies. A threshold-based early stopping with the n-gram score halves memorisation relative to the baseline at low performance cost, but an explicit n-gram penalty in the loss is more effective, averaging 5.45% memorisation and a 4.95% performance gap, with around a 40% reduction in memorisation. We show this scales from small models to 70B-parameter models and generalises across datasets and tasks.

Future work will extend this analysis in two directions. First, alternative decoding strategies such as beam search may surface different leakage patterns and should be audited with the same metrics. Secondly, we will test whether the n-gram regulariser curbs memorisation in code generation, mathematical reasoning, and multimodal tasks.

4.6.1 Mitigation strategies

Based on our findings in this chapter, we recommend the following practices to curb unintended memorisation when this is a concern:

- **Low overhead:** Stop when partial memorisation exceeds a threshold (we used 20). Lowers memorisation with small performance and computation cost (Fig. 4.5).
- **Dataset-aware thresholds.** Track partial memorisation per dataset and tuning mode; variability is high (Fig. 4.2).
- **High risk:** Use the n-gram regulariser (App. 4.9) when risk is high or training to convergence; strongest reductions, extra compute.

4.7 Epilogue

This chapter has established that memorisation arising during fine-tuning can be monitored and curtailed without sacrificing downstream accuracy. A simple n-gram overlap metric provides an actionable early-warning signal, and an n-gram–aware regulariser offers a scalable defence that applies across model families from 1.4B to 70B parameters.

The practical lesson from this work is methodological rather than purely thematic: careful control of the optimisation schedule and an explicit bias against verbatim copying are prerequisites for the representation-centric analyses that follow. This chapter's key insight is to treat memorisation itself as a hierarchical process. Our n-gram metric acts as a proxy for this, revealing that verbatim leakage often begins with the model learning to reproduce smaller constituent structures (n-grams and key phrases) before this behaviour escalates to the memorisation of complete sequences. By monitoring this structural progression from partial to full recall, we can try to encourage the models to not merely store their inputs, but instead generalise learned (hierarchical) structures. This control clears the ground for addressing the final research questions of this thesis: investigating *how* such structure is learned, represented, and exploited across modalities.

We now step beyond a purely linguistic focus within NLP: Chapter 5 examines how hierarchical structure manifests in the spatiotemporal dynamics of video Transformers, while Chapter 6 extends the investigation to a unified next-frame framework spanning language, vision, and audio.

4.8 Appendix: Datasets

The following datasets are used and evaluated according to their respective benchmarks, found in Wang et al. (2018); Taori et al. (2023); Wei et al. (2023).

- **SST-5**. Movie-review sentences annotated with five sentiment levels. *Template:* `Sentence: <s> - What is the sentiment?`. **Metric:** accuracy.
- **QQP**. Pairs of Quora questions labelled as duplicates or not. *Template:* `Q1: <q1>\nQ2: <q2> - Duplicate? Yes/No`. **Metric:** accuracy and F1.
- **RTE**. Premise–hypothesis pairs drawn from news and Wikipedia, framed as binary entailment. *Template:* `Premise: <p> Hypothesis: <h> - Entailed? Yes/No`. **Metric:** accuracy.
- **WANLI**. Large-scale adversarial Natural Language Inference corpus generated via human–AI collaboration. *Template:* `Premise: <p>, Hypothesis: <h>, Label: entail/neutral/contradict`. **Metric:** accuracy.
- **SQuAD v2**. Wikipedia paragraphs paired with questions, mixing answerable and unanswerable cases. *Template:* `Context: <para> Question: <q> Answer:`. **Metric:** exact match (EM) and F1.
- **HellaSwag**. Multiple-choice commonsense completion task built from WikiHow and activity narratives. *Template:* `Story: <ctx> Which ending (A–D) is most plausible?`. **Metric:** multiple-choice accuracy.
- **PubMedQA-L**. Biomedical abstracts with yes/no/maybe answers to research questions. *Template:* `Abstract: <abs> Question: <q> Answer (yes/no/maybe):`. **Metric:** accuracy.
- **XSum**. BBC news articles paired with single-sentence abstractive summaries. *Template:* `Article: <doc> \nWrite a one-sentence summary:`. **Metric:** ROUGE-1/2/L.

- **CNN/DailyMail**. Long-form news articles with multi-sentence "highlights". *Template:* `Article: <doc> Summarise concisely:`. **Metric:** ROUGE-1/2/L.
- **Alpaca-52k**. GPT-3.5-generated instruction–response pairs covering diverse tasks. *Template:* `Instruction: <i>, Input: <in>, Response: <r>`. **Metric:** GPT-4 preference win-rate.
- **FLANv2**. Composite collection of ∼1.8k tasks (12M examples) in instruction format. *Template:* `Instruction: {task}Input: {x} Answer:`. **Metric:** task-specific (Accuracy, F1, ROUGE, etc.).

4.9 Appendix: n-gram Regularisation Loss

To incorporate n-gram regularisation into the standard CLM loss function (as covered in §2.2.1, Eq. 2.5), we modify the loss function to include a penalty term that discourages the model from assigning excessively high confidence to certain n-grams compared to the pre-trained model.

The modified loss function consists of two main components:

1. **Primary Loss Term**:

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}) \tag{4.3}$$

   where T is the total length of the token sequence, x_t is the token at position t, $x_{<t} = (x_1, x_2, \ldots, x_{t-1})$ represents all previous tokens before position t, $p_\theta(x_t \mid x_{<t})$ is the probability of token x_t given previous tokens under the current model parameters θ, and θ represents the model parameters. This is the standard cross-entropy loss used for causal LLM training.

2. **N-gram Regularisation Term**:

$$\mathcal{L}_{\text{reg}} = \lambda \sum_{g \in \mathcal{G}} \mathbb{I}\left(p_\theta(g) > \tau\right) \left[p_\theta(g) - p_{\theta_0}(g)\right]^2 \tag{4.4}$$

where $p_\theta(g)$ is the probability assigned by the fine-tuned model to the n-gram g, $p_{\theta_0}(g)$ is the probability assigned by the pre-trained model to the n-gram g, $\lambda \geq 0$ is the regularisation strength, and $\tau \geq 0$ is the confidence threshold.

The key goal of this term is to penalise the model when it assigns a high probability (exceeding the threshold τ) to an n-gram g more than the pre-trained model does. Balancing this term is key to not overly-penalise the model and reduce latent pre-trained performance. We use $\mathbb{I}\left(p_\theta(g) > \tau\right)$ to ensure that the penalty is applied only when the model's confidence in the n-gram g exceeds the threshold τ.

Bibliography

Anil, R., Ghazi, B., Gupta, V., Kumar, R., and Manurangsi, P. (2022). Large-scale differentially private BERT. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6481–6491, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. (2023). Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020a). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020b). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. (2023). Quantifying memorization across neural language models.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. (2021). Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Cheng, H., Zhu, Z., Sun, X., and Liu, Y. (2021). Mitigating memorization of noisy labels via regularization between representations. *arXiv preprint arXiv:2110.09022*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., and Gardner, M. (2021). Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*.

Fedus, W., Zoph, B., and Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Feldman, V. and Zhang, C. (2020). What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., et al. (2024). The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hans, A., Wen, Y., Jain, N., Kirchenbauer, J., Kazemi, H., Singhania, P., Singh, S., Somepalli, G., Geiping, J., Bhatele, A., et al. (2024). Be like a goldfish, don't memorize! mitigating memorization in generative llms. *arXiv preprint arXiv:2406.10209*.

Jain, N., yeh Chiang, P., Wen, Y., Kirchenbauer, J., Chu, H.-M., Somepalli, G., Bartoldson, B. R., Kailkhura, B., Schwarzschild, A., Saha, A., Goldblum, M., Geiping, J., and Goldstein, T. (2024). NEFTune: Noisy embeddings improve instruction finetuning. In *The Twelfth International Conference on Learning Representations*.

Jiang, A. Q., Sablayrolles, A., Mensch, A., and *et al.* (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., and Lewis, M. (2019). Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Maini, P., Mozer, M. C., Sedghi, H., Lipton, Z. C., Kolter, J. Z., and Zhang, C. (2023). Can neural network memorization be localized? *Preprint arXiv:2307.09542*.

Mireshghallah, F., Uniyal, A., Wang, T., Evans, D., and Berg-Kirkpatrick, T. (2022). Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*.

Miyato, T., Dai, A. M., and Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. (2023). Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Patil, V., Hase, P., and Bansal, M. (2024). Can sensitive information be deleted from llms? objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J. A., Teehan, R., Biderman, S., Gao, L., Bers, T., Wolf, T., and Rush, A. M. (2021). Multitask prompted training enables zero-shot task generalization.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Tänzer, M., Ruder, S., and Rei, M. (2021). Memorisation versus generalisation in pre-trained language models. *arXiv preprint arXiv:2105.00828*.

Taori, R., Guo, P., Li, A., and et al. (2023). Stanford alpaca: An instruction-following llama model. *arXiv preprint arXiv:2303.17580*.

Tejaswin, P., Naik, D., and Liu, P. (2021). How well do you know your summarization datasets? In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3436–3449, Online. Association for Computational Linguistics.

Tirumala, K., Markosyan, A., Zettlemoyer, L., and Aghajanyan, A. (2022). Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupała, G., and Alishahi, A., editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wei, J., Tay, Y., Bommasani, R., and et al. (2023). The flan 2022 collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. (2021). Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*.

Chapter 5

Spatiotemporal Reasoning in Video

Having established in Chapter 3 that Transformer LLMs encode rich, layer-wise hierarchical information in text, we now turn our attention to the visual domain. This chapter investigates whether similar principles of hierarchical internalisation apply to *video*: a data modality inherently structured in both space and time. To explore this, we develop a causal Pixel-Space Spatiotemporal Video Transformer (PSViT), which predicts future frames autoregressively. By training PSViT on synthetic scenes governed by known Partial-Differential Equations (PDEs), we can interrogate the model's internal states using a probing approach analogous to that employed for language.

Our primary goals are twofold: first, to determine if the self-attention mechanisms within PSViT learn multi-scale physical abstractions rather than merely memorising frame sequences; and second, to establish diagnostic tools and insights into visual hierarchical reasoning that will support the unified multimodal framework proposed in Chapter 6, such as optimal hierarchical priors like different self-attention layouts, positional encodings, and network structure. Our investigation involves designing and evaluating spatiotemporal attention layouts, analysing internal network dynamics to localise learned physical signals, and probing hidden states to assess the abstraction of simulation parameters. Through this investig-

ation, we identify several hierarchically-aware optimisations for the final PSViT model, selecting the best-performing spatiotemporal self-attention layout, positional encoding scheme, and U-Net style patch-merging strategy based on extensive ablation studies. These explorations collectively argue that appropriately configured Transformers can indeed extract and localise physical hierarchical structure directly from raw video pixels, even when such structures are not directly observable in pixel-space (e.g., PDE properties governing the video).

The chapter is structured as follows. Section 5.1 introduces the challenges of video prediction, our focus on PDE-driven physical simulations as a testbed for hierarchical reasoning, and summarises this chapter's contributions. Section 5.2 reviews relevant prior work in video generative models, autoregressive video approaches, and dynamic simulation modelling. We then formalise the task of autoregressive video prediction in Section 5.3. Our novel PSViT model architecture, including its patch processing, spatiotemporal attention strategies, and U-Net style adaptations designed to capture hierarchical features, is detailed in Section 5.4. The experimental methodology, datasets providing ground truth for evolving hierarchical systems, and evaluation metrics are described in Section 5.5. Section 5.6 presents our quantitative video prediction results, comparisons with existing approaches, and qualitative assessments of learnt dynamics. Finally, Section 5.7 conducts a structural analysis of PSViT, discussing spatial and temporal reasoning, attention head mechanisms, and the probing of PDE dynamics information from learned representations to understand the internalisation of physical hierarchies.

5.1 Introduction

Building on the insights from Chapter 3 regarding linguistic hierarchy in Transformers, this chapter extends our investigation to the visual domain, specifically focusing on video. Recent progress in the development of Transformer-based generative models has led to increased efforts to extend their application beyond linguistics (Dosovitskiy et al., 2021; Yan et al., 2021; Oprea et al., 2020; Farazi et al., 2021).

Following successes in image generation with models like Variational Autoencoders (VAEs) (Razavi et al., 2019) and Diffusion models (Zhang et al., 2023), generative modelling of videos is an area of escalating research, concentrating on novel architectures and techniques for model interpretability (Castrejon et al., 2019; Oprea et al., 2020; Zhou et al., 2020). In this work, we investigate both aspects. Drawing direct inspiration from the performance and scalability of LLMs, we propose and evaluate a pure Transformer model as an end-to-end approach for unsupervised video prediction. Our primary focus is on physical simulation datasets driven by PDEs, as these provide a quantifiable measure of a model's ability to learn and apply hierarchical spatiotemporal reasoning. Our PSViT model offers a highly simplified architectural approach for end-to-end video prediction while aiming to extend the time horizon of physically accurate outputs, thereby probing the model's capacity to internalise underlying dynamic laws.

Autoregressive Transformer LLMs at scale have been shown to exhibit emergent properties beyond their apparent pre-training goals (Brown et al., 2020; Wei et al., 2022). Video generation is therefore a natural next step for causal modelling, considering both input complexity and computational demand, necessitating innovation and adaptations of existing techniques in deep generative modelling. The emergent and highly generalisable behaviour of autoregressive LLMs hints at promising applications of spatiotemporal modelling beyond conditional video generation. This is particularly true for scenarios where underlying laws governing dynamics are not directly observable from raw pixel-space but represent a deeper hierarchical understanding, such as simulating fluid dynamics (Kohl et al., 2023), weather forecasting (Sønderby et al., 2020), robot motion planning (Finn and Levine, 2017), generating future scenarios for autonomous driving (Wen et al., 2023; Hu et al., 2023), and traffic prediction (Gao et al., 2022).

The majority of existing work on video generative modelling evaluates on either pixel-based or perceptual quality metrics, conditioned on either prior video frames or a textual prompt (Xing et al., 2023; Croitoru et al., 2023; Yu et al., 2023), often with inherently stochastic outputs. These existing approaches for video pre-

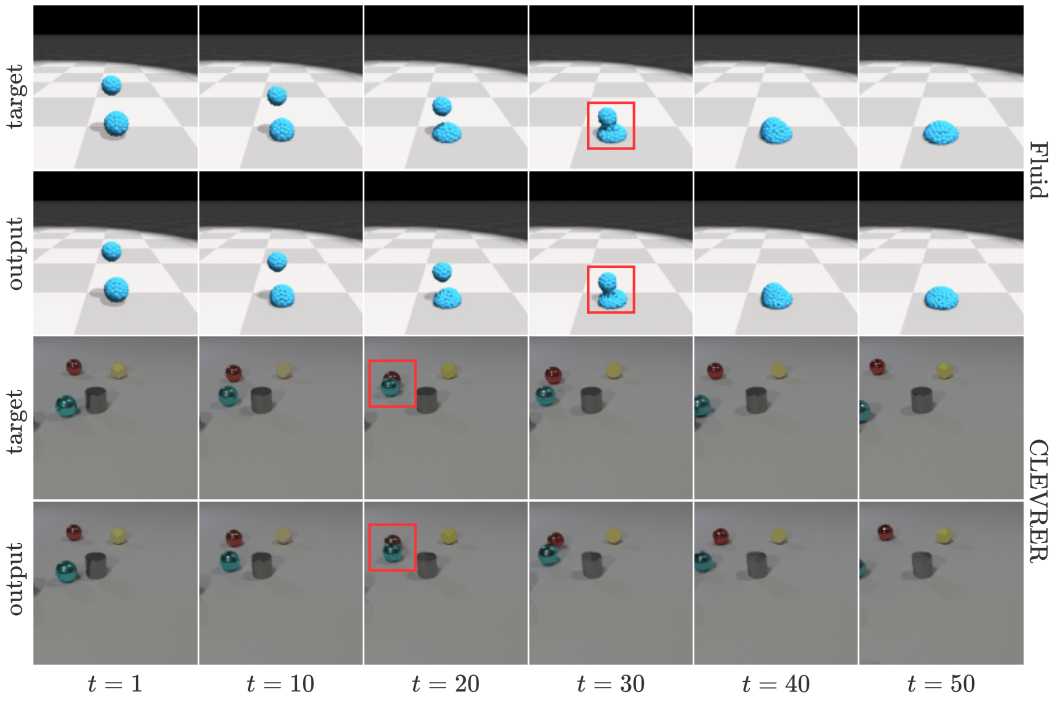


Figure 5.1: Example outputs from our PSViT model trained on video simulation data exhibiting physical dynamics, compared to ground truth frames. Successful collision prediction timestep annotated by the red boxes, indicating the model's grasp of hierarchical event structures.

diction typically employ an *encoder-predictor-decoder* architecture that learns a compressed latent feature space of image frames (discrete or continuous), with the predictor backbone model used to enforce a causal prior for predicting future frames. Yet, there is less work exploring and evaluating the physical accuracy of generated videos over time (an area where these models often falter (Ming et al., 2024)) and investigating whether modelling a continuous pixel-space representation can offer a simpler approach for improved, interpretable video prediction that captures such physical hierarchies. Owing to this, we explore the benefits of using a simple, effective, and interpretable autoregressive Transformer adapted for end-to-end video prediction (Fig. 5.1 shows example model outputs). Our focus is on modelling videos depicting physical simulations, allowing for quantitative evaluation of learnt hierarchical physical dynamics via object positioning over time, and comparison against existing state-of-the-art approaches. We argue that modelling in continuous pixel-space can provide a simple and interpretable approach for investigating Transformers as an end-to-end model for both feature learning and

autoregressive video prediction of hierarchical events. Additionally, we conduct a range of experiments highlighting layer-wise spatiotemporal reasoning, and to what extent our model encodes sequence-specific PDE parameters governing the physical simulation, which represent abstract hierarchical knowledge.

Our key contributions in this chapter are:

- We propose PSViT, a novel end-to-end Transformer, and through it, we design and evaluate several **hierarchical priors** aimed at enhancing **spatiotemporal reasoning**. These priors, including a U-Net style architecture for multi-scale processing and various spatiotemporal attention schemes, demonstrate a simple yet effective approach to video prediction that does not rely on complex, domain-specific components or multi-stage training goals.
- Using an object tracking metric that reflects the understanding of core physical dynamics (a form of hierarchical event structure), we demonstrate that our model offers increased accuracy for video prediction of PDE-driven sequences over time when compared to existing latent-space approaches, as well as competitive performance on common video prediction benchmarks (Moving MNIST and BAIR), highlighting where direct pixel-space modelling can be advantageous for physical coherence.
- We conduct interpretability experiments to identify network regions and attention heads associated with measurable physical dynamics (e.g., collisions, velocity). Furthermore, by probing internal model representations, we accurately estimate out-of-distribution simulation parameters, demonstrating a significant degree of learning and generalisation of underlying physical laws (abstract hierarchical knowledge) not directly apparent in pixel space.

5.2 Related Works

We briefly summarise recent progress in video generative models, Transformer-based approaches for video prediction, and, more specifically, video prediction

models of physical systems, highlighting work relevant to learning spatiotemporal hierarchies not covered in Chapter 2.

5.2.1 Video Models

Research into unsupervised video models explores feature learning from images and videos (Ranzato et al., 2014; van den Oord et al., 2017, 2018; Donahue and Simonyan, 2019; Chen et al., 2020), as well as exhibiting improved downstream task performance from unsupervised pre-training (Chen et al., 2020; Wu et al., 2022; Hong et al., 2023). A new category of image reconstruction loss functions, *DeePSiM* (Dosovitskiy and Brox, 2016), calculates image differences based on features extracted from pre-trained image models, helping to mitigate smoothing artefacts observed when using image-space distance metrics. Existing approaches utilising convolutional architectures include van Amersfoort et al. (2017); Dai et al. (2017); Wang et al. (2020); Yılmaz and Tekalp (2021).

Studies adapting Transformer models for image and video classification (Xie et al., 2021; Dosovitskiy et al., 2021; Bertasius et al., 2021) tend to involve a patch-based framework whereby input images are split into image patches and linearly embedded into a sequence of higher-dimensional 1D vector representations, typical for Transformer model inputs. Feature learning and classification are performed exclusively using the Transformer architecture; as such, the patch-wise processing helps control parameter efficiency and influences how locality is captured. Purely Transformer-based approaches distinguish themselves from CNN-based feature learning primarily in how they handle spatial and temporal locality. Unlike CNNs, which directly capture locality through convolutional layers, any inherent locality as a structural prior in standard Transformers is restricted to intra-patch processing by feed-forward fully-connected layers, with self-attention handling longer-range (potentially hierarchical) dependencies. Predicting high-resolution future frames by leveraging pre-trained image generators and a latent video prediction architecture (Seo et al., 2022) has been shown to significantly reduce training costs and improve prediction quality for various datasets.

5.2.2 Autoregressive Video Models

Autoregressive video prediction models have emerged as a prominent approach for forecasting future frames in video sequences, with notable examples including Weissenborn et al. (2020); Gao et al. (2022). Models for autoregressive video prediction can be broadly defined as pixel-based models (Van Den Oord et al., 2016; Denton and Fergus, 2018; Chen et al., 2020; Gao et al., 2022), or compressed latent models (Rakhimov et al., 2021; Yu et al., 2023). Transformer-based models for video prediction often employ an *encoder-predictor-decoder* structure, whereby a convolutional encoder model compresses the input image representation, with the Transformer component processing the extracted features as a causal predictor model (Yan et al., 2021; Rakhimov et al., 2021; Seo et al., 2022). An example of this is VideoGPT (Yan et al., 2021), which uses a Vector-Quantised Variational Autoencoder (VQ-VAE) encoding strategy with a Transformer model processing the compressed discrete latent representations as a causal frame predictor. SimVP (Gao et al., 2022) uses a Vision Transformer (ViT) (Dosovitskiy et al., 2021) as a predictor model, with CNN encoder/decoders for spatial feature extraction. IAM4VP (Seo et al., 2023) incorporates a stacked autoregressive approach, demonstrating improved performance in preserving temporal coherence and reducing error accumulation over long prediction horizons.

A significant shortcoming of these approaches remains the inherent propagation of errors accumulated at each prediction timestep, resulting in difficulties modelling longer sequences due to out-of-distribution predictions diverging from ground-truth (Oprea et al., 2020). This is evident when observing the physical accuracy of predictions over time, particularly for complex hierarchical dynamics, and hence why we focus on evaluating model performance on predictions involving such physical systems.

5.2.3 Dynamic Simulation Modelling

Introductory work exploring physically accurate video prediction models from unsupervised training (Finn et al., 2016) involves pixel-motion estimation and action-conditioning to generate future image frames. PhyDNet (Le Guen and Thome, 2020) attempts to separate the modelling of videos of physical dynamics governed by PDEs, from unknown residual information (e.g., texture, high-frequency details) by learning a semantic latent representation of the underlying PDE physics separate to variable pixel-based image information. The 'Physics 101' dataset (Wu et al., 2016) was introduced to study physical properties of dynamic objects in video sequences, together with a model designed to explicitly encode physical laws via supervised parameter estimation. We leverage a set of physics-based video prediction datasets (Winterbottom et al., 2024) (further details in Section 5.5), where each dataset is associated with a parameter estimation task, allowing us to probe for internalised hierarchical knowledge about these parameters. The DINo model (Yin et al., 2023) introduces a data-driven approach for PDE forecasting that operates with continuous-time dynamics and spatially continuous functions, allowing learning from sparse and irregular data and generalisation across different grids or resolutions. Other works exploring dynamic system modelling include Tompson et al. (2017); Shi et al. (2017); de Bezenac et al. (2018); Kolter and Manek (2019). A Fourier Neural Operator (Li et al., 2021) is used to directly learn solutions to families of PDEs with high efficiency and accuracy, outperforming previous learning-based PDE solvers. The DyAd model (Wang et al., 2022) employs meta-learning to improve generalisation in deep learning models for dynamics forecasting across varied domains, by partitioning them into distinct tasks. It features a two-part architecture with an encoder for inferring time-invariant task features and a forecaster learning shared dynamics, significantly outperforming existing methods in predicting complex physical phenomena like turbulent flow and ocean currents.

Our approach distinguishes itself from the above through its focus on reducing the need for explicit structural priors specific to modelling physical dynamics. Instead, we investigate an unsupervised, end-to-end training paradigm for a pure Trans-

former architecture, exploring how spatiotemporal attention layouts can themselves facilitate the learning of hierarchical physical regularities directly from pixel data.

5.3 Autoregressive Video Prediction

This section formalises the task definition for video prediction used throughout this work and outlines the associated autoregressive learning objectives.

5.3.1 Task Definition

Consider a sequence V consisting of image frames representing timesteps of a video. Let $\mathcal{X} = \{x_t \mid t = 1, \ldots, T\}$ be an input sequence comprising the first T timesteps of V, and $\mathcal{Y} = \{y_{t'} \mid t' = T+1, \ldots, T+T'\}$ be a target sequence consisting of the subsequent T' frames of V, where each $x_t \in \mathbb{R}^{C \times H \times W}$ and $y_{t'} \in \mathbb{R}^{C \times H \times W}$ represents a C-channel image of height H and width W. Given a video predictive model $\mathcal{F}$ parameterised by θ, $\mathcal{F}$ is tasked with mapping the input sequence $\mathcal{X}$ to the sequence of future frames $\mathcal{Y}$. This can be performed in an autoregressive manner as follows:

$$\hat{y}_{t'} = \mathcal{F}(x_1, \ldots, x_T, \hat{y}_{T+1}, \ldots, \hat{y}_{t'-1}; \theta), \tag{5.1}$$

where,

$$\hat{y}_{T+1} = \mathcal{F}(x_1, \ldots, x_T; \theta), \tag{5.2}$$

such that $\hat{y}_{T+1}$ represents the first predicted frame, directly dependent on the input sequence. Each subsequent frame prediction $\hat{y}_{t'}$ up to timestep $T + T'$ depends on all previous predictions $\{\hat{y}_{T+1}, \hat{y}_{T+2}, \ldots, \hat{y}_{t'-1}\}$ and the original sequence $\mathcal{X}$. The above describes the inference process of a model $\mathcal{F}$ performing autoregressive video prediction given an input sequence $\mathcal{X}$.

5.3.2 Autoregressive Learning Objective

Considering the learning objective of the autoregressive video prediction model described above, the training goal for $\mathcal{F}$ is to minimise the average reconstruction

loss between model predicted frames $\hat{y}_{t'}$ and ground truth target frames $y_{t'}$ for each timestep of the input sequence. During training, the input sequence $\mathcal{X}$ consists of the entire video sequence V minus the final frame; $\mathcal{X} = \{x_t \mid t = 1, \ldots, T-1\}$, with targets $\mathcal{Y}$ consisting of V minus the first frame; $\mathcal{Y} = \{v_t \mid t = 2, \ldots, T\}$, such that predictions at each timestep are conditioned only on ground truth frames and not model outputs. Causal masking of the input sequence ensures that only prior frames contribute to future timestep predictions. Our learning objective per video sequence $\mathcal{X}$ can therefore be expressed as follows:

$$\min_{\theta} \sum_{t=1}^{T-1} \mathcal{L}\left(x_{t+1}, \mathcal{F}(x_1, \ldots, x_t; \theta)\right), \tag{5.3}$$

where $\mathcal{L}$ represents the loss function providing a statistical measure for the reconstruction discrepancy between predicted and target image frames. Typical choices for loss function $\mathcal{L}$ include pixel-wise Mean Squared Error (MSE), Mean Absolute Error (MAE), and Structural Similarity Index Measure (SSIM) (Wang et al., 2004).

5.4 Model Architecture

We build on the ViT (Dosovitskiy et al., 2021) and TimeSformer (Bertasius et al., 2021) Transformer-based models originally designed for image understanding and video understanding, respectively, and not primarily for generative modelling. In this section, we introduce our PSViT model, a pure-Transformer backbone for testing different *spatiotemporal self-attention schemes* for end-to-end unsupervised video prediction in *continuous pixel-space*. We detail the key adaptations needed for both spatiotemporal modelling of video sequences and performing autoregressive video prediction, with a focus on architectural choices that may facilitate the learning of hierarchical representations.

5.4.1 Input Patch Processing

The PSViT model (illustrated in Fig. 5.2) takes as input a video sequence $\mathcal{X} = \{x_t\}_{t=1}^{T}$, where each $x_t \in \mathbb{R}^{C \times H \times W}$ is a C-channel image frame of height H and

width W, sampled at timestep t for $t = 1, 2, \ldots, T$. Each frame x_t is partitioned into S non-overlapping, equally-sized patches of dimension $P \times P$, with H and W being divisible by P to ensure $S = \frac{HW}{P^2}$ patches that span the input image with no padding. Therefore, for each image frame x_t, we denote $x_{t,s} \in \mathbb{R}^{C \times P \times P}$ as the resulting image patch at index s for timestep t, where $s = 1, 2, \ldots, S$ refers to the spatial location of the patch.

5.4.2 Patch Embedding Representation

Transformer models typically expect a sequence of 1D vector inputs. Following the partition of each frame into patches, each patch is further flattened into a 1D vector $\boldsymbol{x}_{t,s} \in \mathbb{R}^{CP^2}$ and linearly embedded into a higher-dimensional embedding space D via a parameterised embedding matrix $\boldsymbol{E} \in \mathbb{R}^{(CP^2) \times D}$:

$$\boldsymbol{z}^0_{t,s} = \boldsymbol{E}\boldsymbol{x}_{t,s}, \tag{5.4}$$

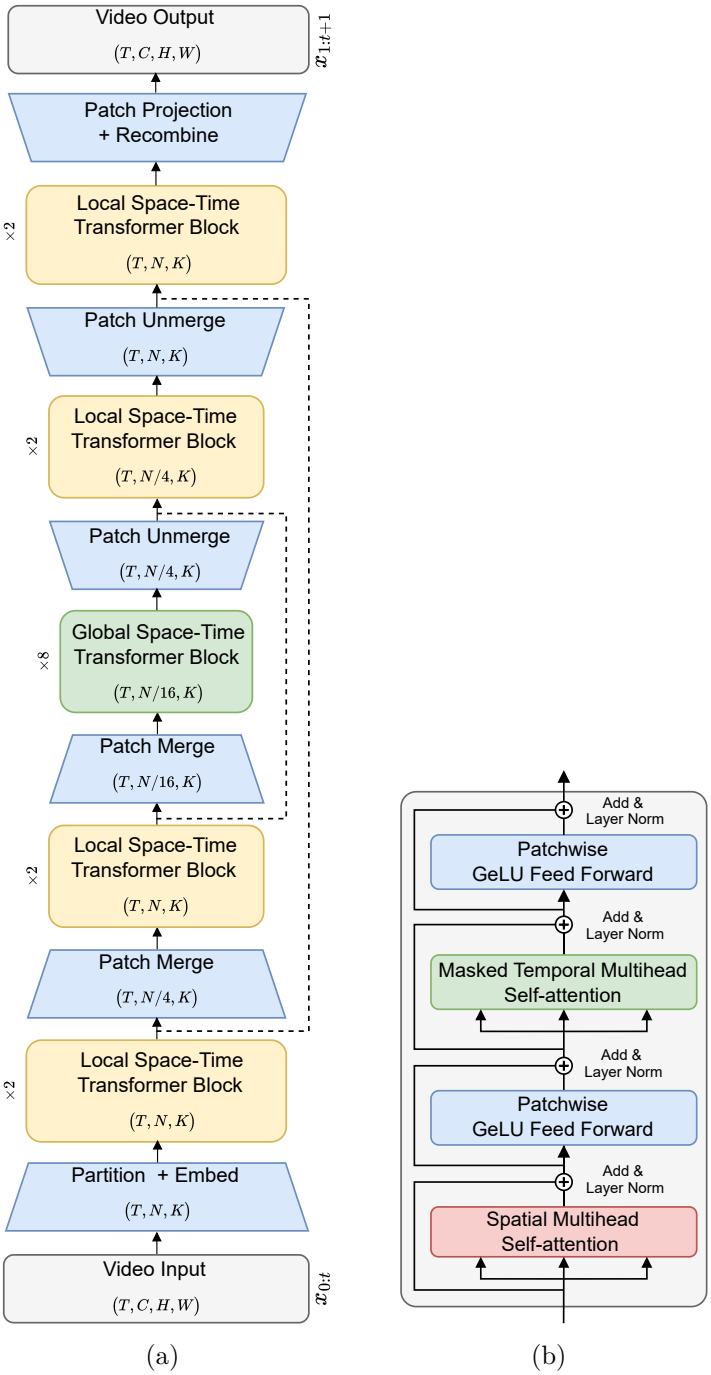
where $\boldsymbol{z}^0_{t,s} \in \mathbb{R}^D$ is the patch embedding vector of dimension D. For spatial and temporal contextual awareness, individual learnable positional encodings for each are added to the patch embedding as follows:

$$\boldsymbol{z}_{t,s} = \boldsymbol{z}^0_{t,s} + \boldsymbol{e}^{\text{time}}_t + \boldsymbol{e}^{\text{space}}_s, \tag{5.5}$$

where $\boldsymbol{e}^{\text{time}}_t \in \mathbb{R}^D$ and $\boldsymbol{e}^{\text{space}}_s \in \mathbb{R}^D$ are parameterised encodings for timesteps $t = 1, 2, \ldots, T$ and spatial positions $s = 1, 2, \ldots, S$, respectively. The resulting sequence of patch embedding vectors $Z = \{\boldsymbol{z}_{t,s} \mid t = 1, \ldots, T; s = 1, \ldots, S\}$ serves as input to the Transformer model, with the embedding process enabling the joint learning of both spatial and temporal information necessary for modelling video sequences.

5.4.3 Spatiotemporal Attention Strategies

Essential to our autoregressive video prediction model is the modification of *multi-head self-attention* (hereafter denoted *self-attention*) layers (Vaswani et al., 2017) to incorporate both intra-timestep spatial relationships and inter-timestep causal



(a) (b)

Figure 5.2: Illustration of (a) an overview of our PSViT model and (b) a space-time transformer layer (a space-time transformer block is constructed by stacking these layers). T timestep video image frames are partitioned into N non-overlapping patches and linearly embedded before being processed by a series of local and global space-time transformer blocks. Skip connections (dashed lines) are used between corresponding patch merge/unmerge operations to preserve information such as static background features, aiding in the representation of hierarchical spatial detail. Input dimensions are annotated for each component.

temporal relationships across successive frames. We use a patch-based model similar to ViT; therefore, we can view any spatial and temporal operations as capturing patch-wise spatial and temporal relationships. Dividing the input frame into patches is an effective method to avoid the quadratic growth in complexity associated with increasing image resolutions. We experiment with a range of spatiotemporal self-attention variations, building on those found in Bertasius et al. (2021). Fig. 5.3 illustrates these spatiotemporal self-attention layouts. A key consideration when adapting these layouts for autoregressive video prediction is to enforce temporal causality via masking of patches from future timesteps during training.

Through preliminary testing, we find that combining spatial and temporal information in a single (global *or* local) self-attention operation (the *Joint-ST* configuration (Bertasius et al., 2021)) performs considerably worse when compared to isolating spatial and temporal information into separately parameterised self-attention layers. We further separate spatial and temporal self-attention operations by an additional patch-wise GeLU (Hendrycks and Gimpel, 2016) FFN, as further explained in Section 5.4.6.

5.4.4 Spatial Attention

Each spatial attention layer performs self-attention over image patch inputs independently for each timestep. This approach allows the layer to learn both local and global patchwise relationships, agnostic of temporal information. We follow a similar procedure for *query*, *key*, *value* self-attention as described in (Vaswani et al., 2017; Dosovitskiy et al., 2021; Bertasius et al., 2021) and also covered in our background Section §2.3.3. Given patch inputs $\boldsymbol{z}_{t,s}$, we have:

$$[\boldsymbol{k}_{t,s}, \boldsymbol{q}_{t,s}, \boldsymbol{v}_{t,s}] = \boldsymbol{z}_{t,s}\boldsymbol{U}_{kqv} \in \mathbb{R}^{3D_{\text{head}}}, \tag{5.6}$$

where key, query, and value vectors are $\boldsymbol{k} \in \mathbb{R}^{D_{\text{head}}}$, $\boldsymbol{q} \in \mathbb{R}^{D_{\text{head}}}$, and $\boldsymbol{v} \in \mathbb{R}^{D_{\text{head}}}$, respectively, D_{head} is the head dimension for multi-headed attention, and $\boldsymbol{U}_{kqv} \in \mathbb{R}^{D\times 3D_{\text{head}}}$ is a linear projection matrix. Note that $\boldsymbol{U}_{kqv}$ is shared across timesteps.

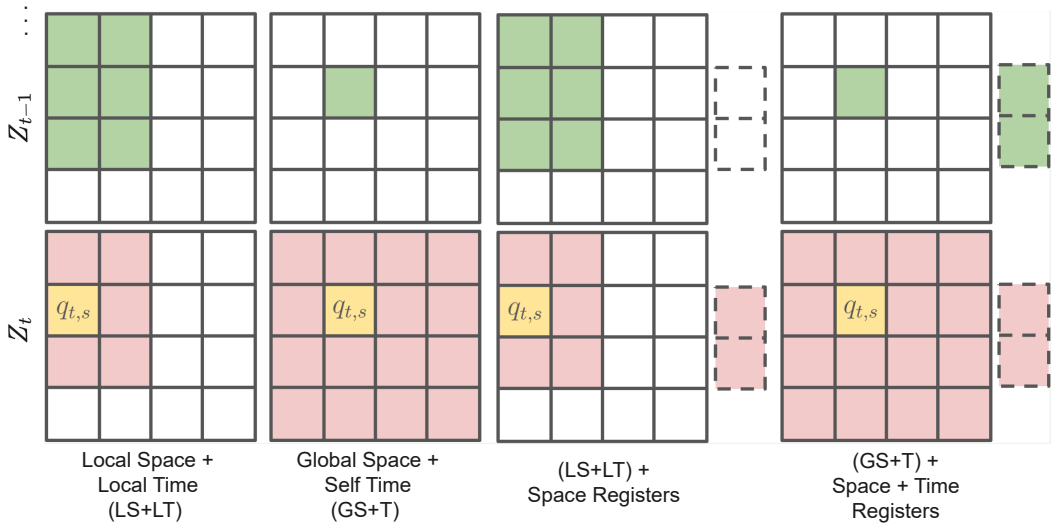


Figure 5.3: Examples of various spatiotemporal self-attention masking strategies using video image frames split into 16 patches, with register tokens (additional empty patch tokens added to the sequence used to accumulate sequence-level information) visualised by dashed patches. For each strategy we illustrate the following: a query patch $q_{t,s}$ shown in yellow for timestep t and patch s, corresponding patches used for spatial self-attention computation shown in red, patches used for temporal self-attention computation shown in green, and patches not used or masked shown in white. Not visualised are *all* previous timestep frames (computation remains the same for these), and future frames (for which all are causally masked during training). These strategies influence how local and global hierarchical dependencies are captured.

We then calculate spatial attention as follows:

$$\boldsymbol{z}_{t,s}^{\text{space}} = \text{SM}\left(\frac{\boldsymbol{q}_{t,s} \cdot \begin{bmatrix} \boldsymbol{k}_{t,1}^\top & \boldsymbol{k}_{t,2}^\top & \cdots & \boldsymbol{k}_{t,S}^\top \end{bmatrix}}{\sqrt{D_{\text{head}}}}\right) \cdot \begin{bmatrix} \boldsymbol{v}_{t,1} \\ \boldsymbol{v}_{t,2} \\ \vdots \\ \boldsymbol{v}_{t,S} \end{bmatrix}, \tag{5.7}$$

where SM is the softmax operator, and $\boldsymbol{z}_{t,s}^{\text{space}}$ is a single head spatial attention output for spatial patch s and timestep t. For Local-Space attention (shown in Fig. 5.3), spatial patch indices not included in the computation are masked. We follow typical multi-headed self-attention practices for unifying attention heads and applying layer normalisation as covered in our background Section §2.3.3.

5.4.5 Causal Temporal Attention

To enable generative autoregressive behaviour, temporal self-attention is causally masked, allowing each patch to attend only to its own past and current timesteps across different frames, preventing information leakage from the future. Similar to spatial attention, for each spatial patch we apply temporal attention across timesteps, but limited to corresponding spatial patch positions. Due to better observed performance in preliminary testing, we differ from existing approaches and learn separate temporal query, key, and value projections calculated from the output of the preceding spatial attention layer.

5.4.6 Layer Outputs

The output for each patch at each intermediate layer ℓ for L layers is as follows:

$$\boldsymbol{z}_{t,s}^{\ell} = \text{FFN'}(\boldsymbol{z}_{t,s}^{\text{time}}(\text{FFN}(\boldsymbol{z}_{t,s}^{\text{space}}))), \tag{5.8}$$

where FFN and FFN' are patchwise non-linear FFN components with layer normalisation and residual connections, typical of those found on the output of the original Transformer layers. We note the following modifications to existing approaches: we further separate the computation of spatial and temporal attention via an FFN layer between spatial and temporal attention operations, and perform spatial attention before temporal attention due to consistently better results. Results are reported for ablations of this setup. The final layer FFN' output is down-projected to $C \times P \times P$ and reshaped to the original image patch dimensions, with patches at each timestep being reconstructed to form an output image $\hat{y}_{t'} \in \mathbb{R}^{C \times H \times W}$ as illustrated in Fig. 5.2(a).

5.4.7 U-Net Style Adaptation

We further adapt the typical Transformer backbone architecture to progressively process patches at smaller resolutions, similar to that of U-Net style models (Ronneberger et al., 2015), such that adjacent input patches are linearly merged to

reduce the effective image resolution at each block on the encoding side. This design choice allows us to train end-to-end models efficiently without resorting to pre-trained encoder/decoder models and encourages the learning of hierarchical spatial features. As illustrated in Fig. 5.2, partitioning into patches is followed by a series of *local* space-time Transformer blocks separated by patch merge operations, with a *global* space-time Transformer block operating on the reduced resolution patch-based representations. In the example shown, local blocks are restricted to local spatial and temporal attention windows, separated by non-overlapping patch merge/unmerge operations with a patch window size of 2. Global blocks operate on a smaller effective image resolution (e.g., for an input of 64×64 and a patch size of 8×8, the effective resolution at the global blocks is 16×16). This has the benefit of capturing multiscale spatial features, while reducing the computational complexity of the global self-attention operations due to a smaller number of image patches. The reverse process is applied on the decoding side of the model to produce a predicted next image frame for each input frame.

5.4.8 Register Tokens

We experiment with the addition of a set of register tokens (Darcet et al., 2024) added to the sequence of embedded image patches at each timestep as a possible encoding mechanism for sequence-level information and PDE dynamics, representing higher-level hierarchical information. N_r learnable register tokens $\boldsymbol{r}_n \in \mathbb{R}^D$, for $n = 1, 2, \ldots, N_r$ are appended to the set of S patches for each timestep t after the positional encodings step (Eq. (5.5)), such that our input to the Transformer becomes:

$$Z_t = [\boldsymbol{z}_{t,1}, \ldots, \boldsymbol{z}_{t,S}, \boldsymbol{r}_{t,1}, \ldots, \boldsymbol{r}_{t,N_r}] \text{ for } t = 1, \ldots, T, \tag{5.9}$$

(where Z is the full sequence of these Z_t). These tokens are discarded before recombining image patches to generate an output image. The intuition behind incorporating these tokens is that they act as dedicated "memory" slots that aggregate non-local, sequence-level signals, such as global boundary conditions or

slowly varying PDE modes, which can be accessed by each spatial patch. These can serve as a structural prior to enable the aggregation of useful hierarchical information, both spatially and temporally. We explore the encoding of information in these register tokens later in the Chapter.

5.5 Methodology

This section covers the datasets used for self-supervised video prediction training, outlines our experimental setups for training and evaluation, and details the different model configurations we use.

5.5.1 Datasets

Our work focuses on unsupervised prediction of physics-based simulations involving objects moving and interacting according to well-defined physical laws, providing ground truth for complex, evolving hierarchical systems. Specifically, we employ a set of physics-based *simulation* datasets introduced in Winterbottom et al. (2024). These datasets offer a controlled and visually-simple set of dynamic PDE simulations (calculated using the Runge-Kutta method (Butcher, 1996)), useful for isolating physical accuracy of frame predictions, and evaluating spatiotemporal reasoning. Each sequence per dataset is generated under unique initial conditions and simulation parameters, ensuring no parameter contamination between train, validation, and test splits. We use the following datasets: *Moon* - an orbital dynamics simulation involving a static body and an orbiting moon, *Pendulum* - a swinging pendulum under gravity, *Roller* - a rolling ball on a curved surface acting under gravity, and *3D Balls* - a 3D environment containing elastic collisions between moving balls and the environment walls.

Each simulation dataset contains 3-channel image frame sequences at a resolution of 128×128 pixels. We split into training, evaluation, and testing subsets following an 80/10/10 split, ensuring no overlapping initial conditions or parameters across the splits. In addition to the simulation datasets above, we experiment with video

Table 5.1: Dataset Specifications

Dataset	Samples	Seq. Length	Resolution	Num. Objects	PDE Variables
Roller	5,000	100	128×128	1	3
Pendulum	5,000	100	128×128	1	5
Moon	5,000	100	128×128	2	4
3D Balls	5,000	100	128×128	1-3	1
CLEVRER*	5,000	140	128×128	5	-
Fluid*	1,000	100	128×128	-	-

Table 5.2: Parameter Estimation Details for Probing Hierarchical Knowledge

Dataset	Parameter	In-distribution range	Out-of-distribution range
Roller	Gravity	0-100	100-150
Pendulum	Gravity	0-6	6-10
Moon	Mass	72-200	200-300

prediction on the CLEVRER (Yang et al., 2019) colliding objects dataset, and the Fluid simulation from DPI-Net (Li et al., 2019). Additionally, we benchmark our model on two common video prediction benchmark datasets, namely Moving MNIST (Srivastava et al., 2015), and BAIR robot pushing datasets (Ebert et al., 2017). For the CLEVRER, Fluid, Moving MNIST, and BAIR datasets, we use predefined training and testing splits. Specifically, for CLEVRER and Fluid, where the original resolution exceeds 128×128 pixels, we apply a central square crop and downsample to 128×128. A full description of each dataset follows.

Dataset Details Table 5.1 summarises details of the simulation datasets used in the main work. We further detail the datasets used for parameter estimation probing tasks below, with Table 5.2 summarising details of the training in-distribution and out-of-distribution PDE parameter ranges used in the probing experiments.

The *Moon* simulation consists of an *orbiting moon*; treated as a rigid body with a random initial velocity, and a *static celestial body*. The simulation models the gravitational interaction between the moon and the static celestial body as follows:

$$m_m \frac{d^2\vec{r}}{dt^2} = -\frac{GM_c m_m}{|\vec{r}|^3}\vec{r}, \tag{5.10}$$

where $\vec{r}$ is the position vector of the moon relative to the centre of the celestial body, $|\vec{r}|$ is the distance between the two bodies, m_m is the mass of the moon, M_c

is the mass of the celestial body, G is the gravitational constant, and $\frac{d^2\vec{r}}{dt^2}$ represents the acceleration of the moon due to gravity. For each sequence, we vary the initial tangential velocity of the moon, the radius of the moon, the radius of the celestial body, and the mass of the celestial body, simulating different orbital trajectories. G and m_m are kept constant for all samples.

The *Pendulum* simulation consists of a single pendulum modelled as a point mass at the end of a massless rod. The pendulum swings about a fixed pivot point, which serves as the centre of rotation for all simulated sequences. The dynamics are governed by the following:

$$\theta''(t) = -\frac{g}{l}\sin(\theta(t)), \tag{5.11}$$

where $\theta(t)$ is the angle of the pendulum relative to the vertical, g represents the gravitational strength, and l is the length of the pendulum. For each sequence, we vary the following parameters: the *initial angle* of the pendulum, *gravitational strength*, *pendulum length*, *pendulum mass*, and the *size of the pendulum*.

The *Roller* simulation exhibits the motion of a ball of mass M rolling down a curved track under the influence of gravity. The force acting on the ball along the track is given by the equation:

$$F = M \cdot g \cdot \cos(\alpha), \tag{5.12}$$

where F is the force, M is the mass of the ball, g is the acceleration due to gravity, and α is the angle between the track and the horizontal plane. The ball transitions to free flight if the normal acceleration exceeds the limit set by the curve's radius of curvature at any point. This condition is mathematically represented as $a_n > \frac{v^2}{k}$, where a_n is the normal acceleration, v is the velocity of the ball, and k is the radius of curvature at the current point on the track. We vary the gravity strength g and the initial position.

Dataset generation All four in-house simulation datasets (Moon, Pendulum, Roller, 3D Balls) were rendered from deterministic ODE/PDE solvers (Runge–Kutta

integration (Butcher, 1996)) with unique seeds and parameter draws per sequence, ensuring disjoint train/val/test splits by construction. We validated sequence quality in three ways: (i) we checked numerical stability and simple invariants (e.g., pendulum length constancy, elastic collision momentum symmetry, orbital radius boundedness under the chosen mass–gravity pairs), (ii) parameter coverage: we verified the intended parameter ranges and non-overlap across splits (also used later for the probing ranges in Table 5.2), (iii) render checks: automatic centroid/extent tracking to flag off-screen objects, or overlaps, followed by a quick visual pass. These checks ensured the videos reflect the target dynamics without visual artefacts that could complicate evaluation. We render the in-house datasets in greyscale to simplify the visuals and isolate physical accuracy. By contrast, the other benchmarks include richer visual complexity and colour. For consistency, all datasets are supplied to the model as three-channel images.

Across all datasets, all scenes are rendered from a fixed camera pose. Consequently, apparent scale is controlled by object parameters (e.g., moon radius, pendulum bob size) rather than viewpoint. This helps isolate the effect of the underlying physical variables we later probe.

5.5.2 Object Divergence Metrics

For each simulation dataset as well as the CLEVRER and Fluid dataset, we perform evaluation using an object divergence metric, whereby the centroid positions of objects contained in the observed scene are compared between predicted and ground-truth sequences. This allows us to have an image quality-invariant evaluation of object positional prediction over time, reflecting a key aspect of understanding hierarchical event structure. For each predicted timestep during inference, the centroid position of each object in the image frame is compared to its ground truth position via 2D Euclidean pixel distance. A rolling average pixel divergence up to t timesteps is taken as the *Divergence* score up to t, with lower indicating closer prediction to ground truth. We normalise scores assuming a 128×128 resolution to allow for fair comparison. For our main results, we evaluate using $t = 50$.

It is worth noting that the CLEVRER dataset involves frames in which moving objects appear from outside of view, and therefore are not predictable and not considered for object divergence scores. Additionally, the Fluid dataset contains groups of many particles which collide; we consider the centroid of each group for tracking purposes.

5.5.3 Positional Encodings

We experiment with multiple positional encoding schemes for our model. A common approach to providing positional information to each token input is to modify the input representation with absolute positional embeddings for each token, *e.g.,* in the form of a periodic sinusoidal function, or a Learnable Positional Encoding (LPE) via a parameterised embedding trained jointly with the model (Zhang et al., 2022). Relative embeddings via positional lookup tables are used in Raffel et al. (2020), and Rotary Positional Embeddings (RoPE) (Chowdhery et al., 2023; Touvron et al., 2023) perform rotational operations on the query and key self-attention matrices, using angular values from absolute positions. We experiment with APE, RoPE, and LPE for both spatial and temporal encodings (e.g., Eq. (5.5)).

5.5.4 Self-attention Strategies

We experiment with three different spatiotemporal self-attention strategies (illustrated in Fig. 5.3) for the middle space-time layers (the *Global Space-Time Block* in Fig. 5.2(a)), namely local-space + local-time (LS+LT), global-space + self-time (GS+T), and global-space + local-time (GS+LT). While reducing the patch-wise receptive field at each layer, local self-attention operations benefit from linear complexity scaling relative to image resolution and choice of patch size P. Additionally, we test the impact of adding register tokens to the input patches as described in Section 5.4. We experiment with four learnable register tokens, shared across timesteps. Via preliminary testing we find that a patch size $P = 8$ performs better than $P = 16$ (used in the original ViT work). Increasing the patch size beyond this becomes parameter inefficient with no observed increase in performance, while

reducing patch size below $P = 8$ degrades performance alongside prohibitively high memory usage. Finally, we test the impact of model scale, with Table 5.3 detailing the configurations of each.

5.5.5 Training Setup

For all experiments and model configurations, training is performed using the Adam optimiser (Kingma and Ba, 2015) with a learning rate set at 3×10^{-5}, a weight decay factor of 1×10^{-4}, and a batch size of 32. These parameters were selected following a thorough evaluation of hyperparameters across various datasets and models. Each model is trained using the SSIM loss function (experiments using other loss functions were conducted, with SSIM generalising best to all datasets), with input pixel values normalised to the range $[0, 1]$. We train each model for a maximum of 500 epochs. For all setups, we use patch-wise Gaussian noise applied to a random subset of image patches, due to significant improvements in reducing error propagation during inference. For simplicity, we do not include any additional data augmentation techniques. All models are trained on an NVIDIA A100 GPU.

5.5.6 Existing Approaches

We compare our method against state-of-the-art video prediction approaches (model parameter sizes in brackets). For latent-space baselines, we use: *a) MAGVIT* (300M) (Yu et al., 2023), which tokenises clips with a 3-D VQ-VAE and reconstructs them non-autoregressively via masked-token prediction with a Transformer; *b) Latent Diffusion Transformer* Open-Sora (1B) (Zheng et al., 2024), whose Transformer denoiser learns spatiotemporal dynamics directly on compressed video latents; and *c) CV-VAE* (160M) (Zhao et al., 2024), a continuous 3-D video VAE made compatible with pre-trained image VAEs for efficient spatiotemporal latent modelling. Additionally, we benchmark against *SimVP* (33M) (Gao et al., 2022), a convolutional encoder–decoder that autoregressively forecasts pixels without recurrence or attention. We follow the original authors' training protocols for each baseline wherever possible.

Table 5.3: Model Configurations.

Model	Global ST Layers	Heads	Model dim. D	Head dim.	FFN dim.	Param.
PSViT small	8	12	512	64	2048	49M
PSViT medium	12	12	512	128	2048	84M

These four baselines span key design paradigms of modern video prediction: latent vs. pixel space, and non-autoregressive, autoregressive, and diffusion Transformer architectures, allowing for a broad comparison regarding the ability to capture hierarchical physical dynamics.

Table 5.4: Video Prediction Results and Divergence Scores. Divergence scores assess the model's ability to predict object trajectories accurately over time.

			Divergence ($t = 50$) ↓				
Approach	**Roller**	**Moon**	**Pendulum**	**3D Balls**	**Fluid**	**CLEVRER**	**Avg**
PSViT Attn. Strategy							
- LS + LT	1.52	2.20	2.08	5.83	2.54	6.01	3.36
- GS + LT	1.32	2.17	1.94	5.41	2.31	5.30	3.08
- GS + T	1.30	2.18	1.92	5.41	2.28	5.28	**3.06**
PSViT Pos Encoding							
- APE	1.41	2.18	2.03	5.57	2.34	5.71	3.21
- RoPE	1.30	2.16	1.97	5.37	2.24	5.41	3.08
- LPE	1.28	2.17	1.92	5.28	2.25	5.41	**3.05**
CV-VAE (160M) (Zhao et al., 2024)	1.18	2.10	1.98	5.35	2.88	5.29	3.13
Diffusion Transformer (1B) (Zheng et al., 2024)	1.20	2.14	2.06	5.42	2.95	5.35	3.19
MAGVIT (300M) (Yu et al., 2023)	1.17	2.20	2.12	6.02	2.54	5.40	3.24
SimVP (33M) (Gao et al., 2022)	1.19	2.12	2.04	5.39	2.93	5.32	3.16
PSViT small (49M)	1.28	2.17	1.92	5.28	2.25	5.41	3.05
PSViT medium (84M)	1.18	2.06	1.84	5.10	2.04	5.18	**2.90**
			SSIM ↑				
MAGVIT (Yu et al., 2023)	0.9996	0.9996	0.9993	0.9943	0.9654	0.9803	0.9901
SimVP (Gao et al., 2022)	0.9996	0.9996	0.9994	0.9946	0.9674	0.9839	0.9908
PSViT medium	0.9998	0.9997	0.9997	0.9961	0.9752	0.9951	**0.9943**
			PSNR ↑				
MAGVIT (Yu et al., 2023)	59.70	56.56	55.59	50.99	47.03	50.76	53.44
SimVP (Gao et al., 2022)	58.42	56.60	55.36	50.53	48.23	50.21	53.23
PSViT medium	59.78	56.63	56.57	53.91	47.19	53.30	**54.56**

5.6 Results

The following section evaluates our proposed PSViT model for autoregressive video prediction, presenting comparisons with existing approaches, the impact of input context size, and qualitative assessment of model outputs. All results are evaluated using held-out test sets for each dataset and model.

5.6.1 Video Prediction Results

Our full set of results for video prediction on the simulation datasets are presented in Table 5.4. As well as object divergence (lower is better), we report image quality metrics SSIM (a perception-based index that compares luminance, contrast, and structure to approximate human-judged fidelity), and PSNR (a decibel-scale measure based on the mean-squared error that quantifies reconstruction quality). Further details and a comparison can be found in Horé and Ziou (2010). SSIM and PSNR scores (higher is better for both) are an average over the first 5 output frames, while divergence scores are taken after 50 output timesteps. All scores are test set averages, with the best average performance for each comparison in bold. We first consider PSViT attention strategy comparisons, finding that GS+T and GS+LT space-time attention schemes perform significantly better than LS+LT across the board, clearly showing that reducing the receptive field for each spatial attention operation is detrimental for these datasets. Restricting temporal attention to the same patch location (the +T scheme) has very little difference in performance compared to global temporal attention, with the benefit of linear complexity with regard to sequence length. The difference is most apparent for the more visually-complex 3D datasets (3D balls and CLEVRER), implying much better performance at stable object prediction up to this time horizon while handling increased visual complexity. Learned spatial positional encodings (LPE) provide a clear benefit across the majority of datasets compared to existing encoding techniques adapted from language modelling (APE, RoPE). Example model outputs representative of the median performance for object divergence are shown in Fig. 5.4 for the CLEV-

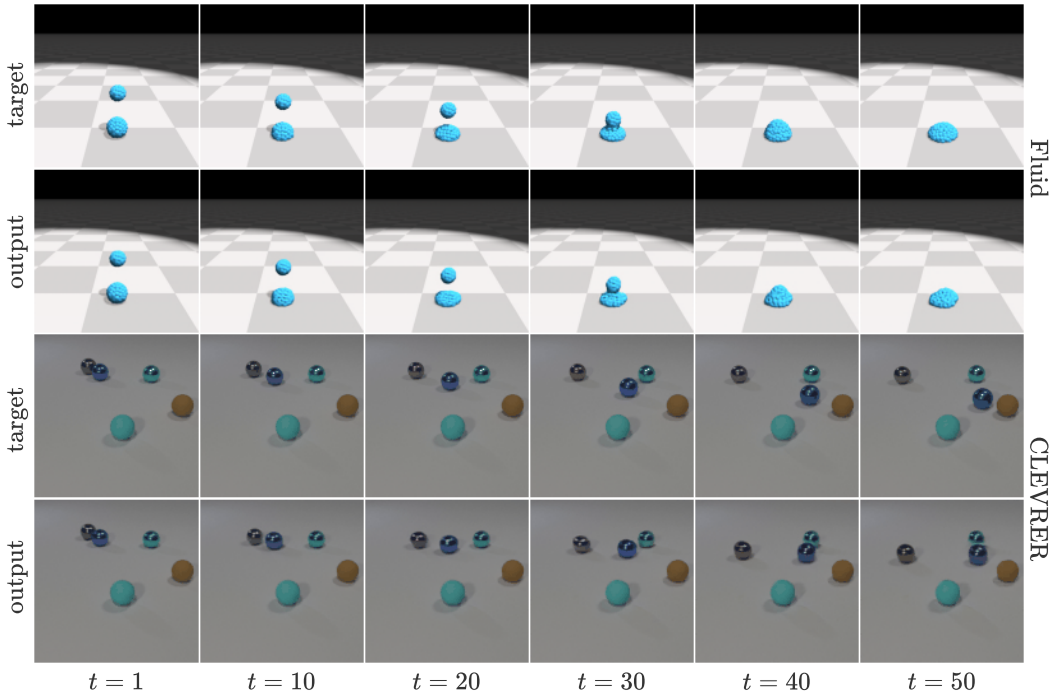


Figure 5.4: Sample outputs from our PSViT model on the Fluid and CLEVRER datasets, conditioned on 12 input frames, alongside ground truth comparison. Intermediate timestep outputs are not shown. These examples illustrate the model's ability to generate complex, evolving scenes.

RER and Fluid datasets. We observe accurate object trajectories and interactions of multiple objects relative to ground truth frames up to $t = 30$, after which we observe deviations from target object positions. Fig. 5.5 shows example outputs for the Roller, Moon, and 3D Balls datasets, where a similar observation is made, with accurate predictions of object trajectories (a key hierarchical dynamic) up to $t \approx 30$.

When considering different model scales and the addition of parameterised register tokens, we observe a performance increase across all tasks. Perhaps unsurprisingly, increasing model size has a bigger performance impact on the 3D datasets when compared to the 2D physics simulations, suggesting that the increased parameterisation is necessary for handling the increase in visual complexity rather than solely for learning PDE dynamics.

Fig. 5.6 compares performances at each output timestep across datasets for divergence, L1, SSIM, and PSNR. More generally, we note a large relative drop in performance (higher divergence) for the 3D and CLEVRER datasets over longer ho-

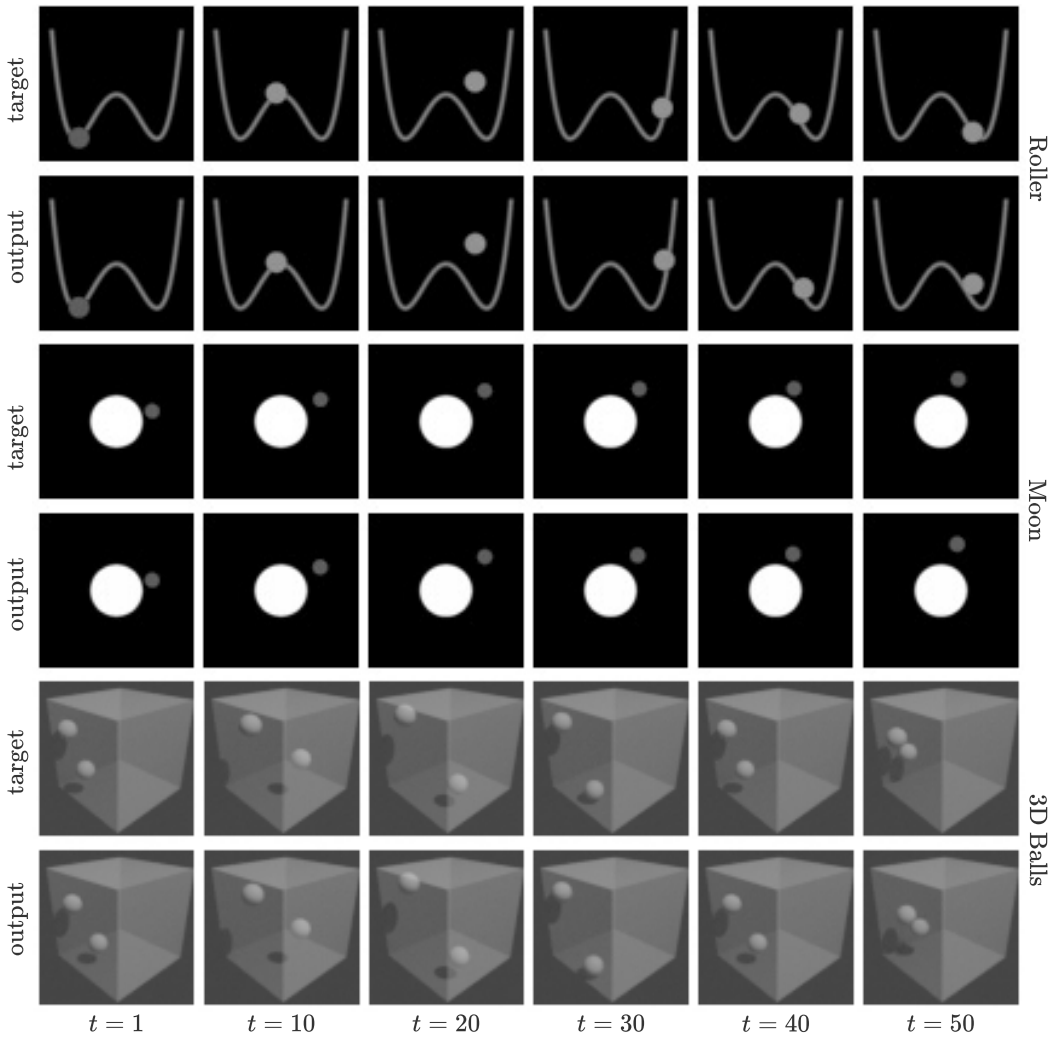


Figure 5.5: Sample outputs from our PSViT model on the Roller, Moon, and 3D Balls datasets, conditioned on 12 input frames, alongside ground truth comparison. Intermediate timestep outputs are not shown. Each example shown is representative of the median divergence performance for each dataset, illustrating the model's ability to capture accurate physical trajectories.

rizons. Interestingly, this difference in sustained physical accuracy is not captured as markedly by standard L1, SSIM, and PSNR metrics, suggesting that including our object divergence metric is an important factor in determining model performance regarding the understanding of underlying physical hierarchies. All tasks show low object divergence up to $t \approx 20$, with the 2D simulation datasets maintaining slightly better coherence for longer. The divergence rate for the Roller, Moon, and Pendulum datasets appears to correlate with the number of underlying PDE variables; the Pendulum dataset, having five PDE variables, performs the worst in terms of divergence, compared to the Roller dataset with two. Considering the

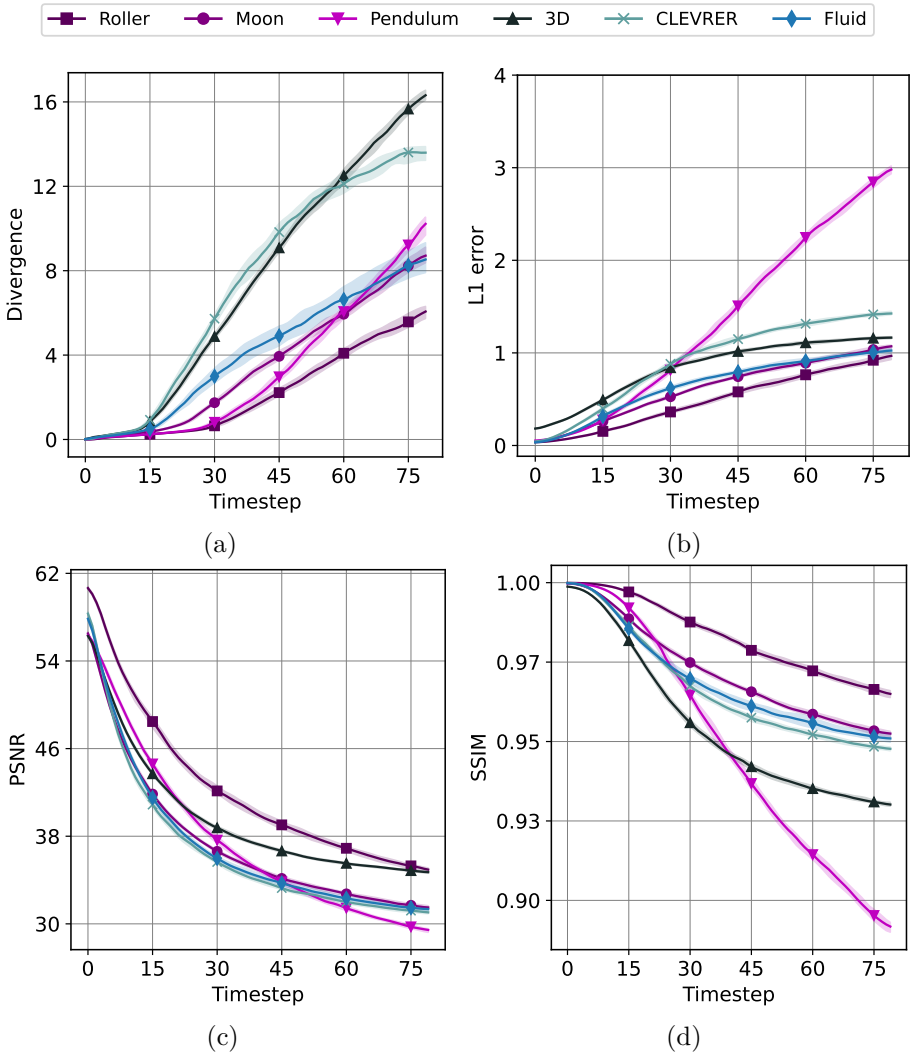


Figure 5.6: Model performance metrics over time for (a) Object divergence , (b) L1 Error, (c) PSNR, and (d) SSIM. All simulations are run with an input-frame context size of 12 timesteps, self-conditioned up to 80 timesteps of output frames. Object Divergence scores are a taken as a median of rolling averages over all test set sequences, reflecting the model's ability to maintain object trajectory coherence.

relatively high performance of the Fluid dataset given its visual complexity, we attribute this to lower variance between samples in terms of overall flow patterns, where object interactions (particle group collisions) typically occur near the centre of the image.

5.6.2 Comparing Input Context Sizes

We experiment training our model with different fixed input context sizes, shown in Fig. 5.7. The impact of additional context frames differs significantly between

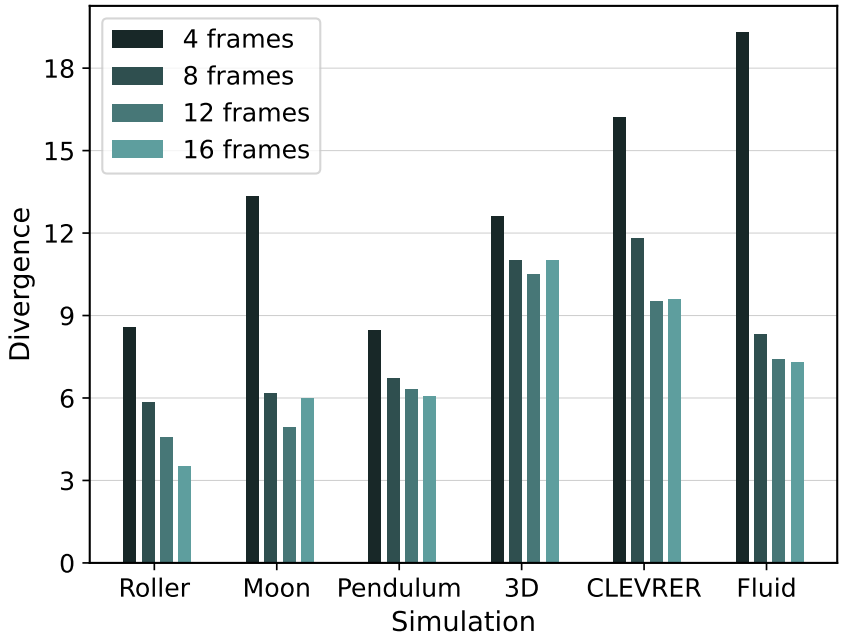


Figure 5.7: Comparison of different input context sizes for each dataset using the PSViT model. Object divergence scores are taken after 50 timesteps of autoregressive prediction. Scores are an average over the test sets.

tasks. Interestingly, we see that the maximum context size does not achieve the best performance for three of the datasets studied, indicating a limitation on handling increased context length, or perhaps that shorter contexts are sufficient for these simpler PDE dynamics. We find that the number of context frames has a significant impact on the timestep at which divergence from ground truth begins to occur, but does not necessarily impact the rate of divergence past this point. The significant reduction in divergence between 4 and 8 frames of context for the Fluid dataset is expected due to the high number of objects being modelled and the complexity of their spatiotemporal interactions.

5.6.3 Comparison with Existing Approaches

We compare our approach with recent state-of-the-art models covering different paradigms in video prediction, namely MAGVIT (Yu et al., 2023), SimVP (Gao et al., 2022), Latent Diffusion Transformer (Zheng et al., 2024), and CV-VAE (Zhao et al., 2024). Fig. 5.8 compares object divergence performance between four models on the Roller, Moon, Pendulum, and 3D Balls datasets. Both sizes of our PSViT

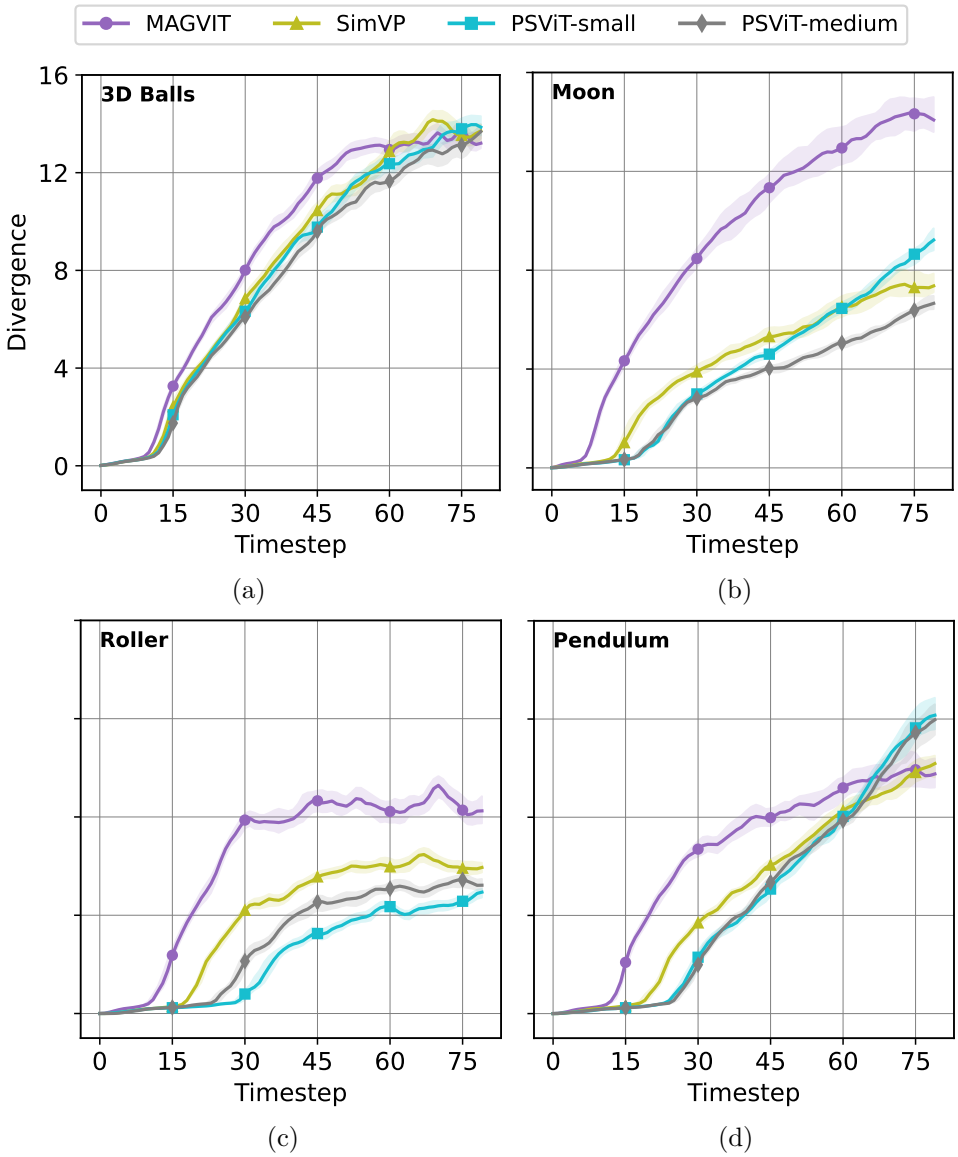


Figure 5.8: Object divergence performance over time comparing our model (*PS-ViT*) and existing approaches (*MAGVIT* and *SimVP*), on the (a) 3D Balls, (b) Moon, (c) Roller, and (d) Pendulum datasets. Lower divergence indicates better adherence to physical trajectories.

model perform significantly better (lower divergence) across the first 50 timesteps in comparison to SimVP and MAGVIT, with the exception of the 3D Balls dataset where all models have similar performances. We attribute this to the relatively simple dynamics of the 3D Balls dataset, as all objects share a constant velocity across all sequences (only size and initial trajectory are varied). The most notable difference is the timestep at which divergence begins to increase compared to both existing approaches, with our approach significantly increasing the time horizon of

Table 5.5: Video prediction performance on BAIR and Moving MNIST

Approach	BAIR (FVD ↓)	Moving MNIST (SSIM ↑)
SimVP (Gao et al., 2022)	67.1	0.948
MAGVIT (Yu et al., 2023)	62.4	0.938
CV-VAE (Zhao et al., 2024)	63.6	0.945
Diffusion Transformer (Zheng et al., 2024)	**61.0**	0.950
PSViT-medium (ours)	64.1	**0.963**

accurate prediction of these hierarchical trajectories on all datasets by up to 50%.

Averaged over all datasets, Table 5.4 shows both scales of our approach perform favourably for Divergence scores against all comparisons, especially considering the parameter differences to models such as the Diffusion Transformer.

In addition to the physical simulation datasets, we also compare our model on two common video prediction benchmark datasets, BAIR and Moving MNIST. Results are detailed in Table 5.5. Fréchet Video Distance (FVD) (Unterthiner et al., 2019) is used for the BAIR dataset. We observe competitive performance of our model on the BAIR probabilistic dataset, although the latent space approaches outperform both ours and SimVP on FVD. For the Moving MNIST dataset, the latent space approaches perform worse on SSIM. This reaffirms that while latent models can produce high-quality video generation needed for stochastic datasets such as BAIR, they may fall short in terms of consistent physical coherence over time when compared to models directly predicting in pixel space.

5.6.4 Qualitative Assessment

We perform a qualitative analysis of model outputs over time, and where they begin to fall short and diverge from the true spatiotemporal evolution of the scene. Fig. 5.5 shows example model outputs from the visually-simpler physics simulation datasets. Our model is clearly capable of preserving the shape and colour of objects over time, regardless of divergence to ground truth. Prediction errors in the spatiotemporal dynamics begin to appear at the later timesteps; for instance, in the 3D Ball scenario shown, there are clear inconsistencies between the shadows of the colliding objects. The same is true for the CLEVRER and Fluid examples shown

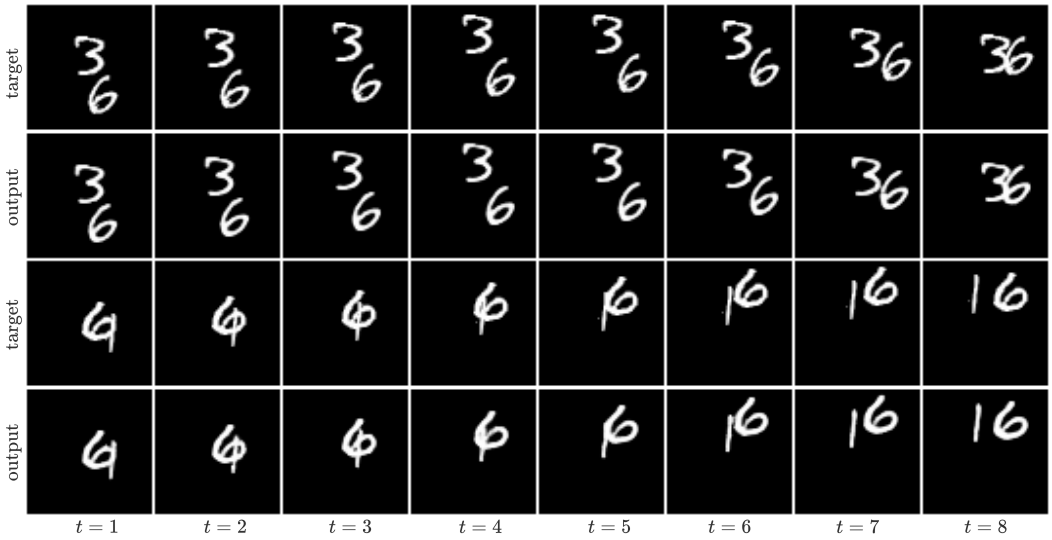


Figure 5.9: Randomly selected model predictions from the Moving MNIST dataset conditioned on 4 input frames using PSViT Medium. For each sample, top rows are ground truth and bottom rows are model outputs, visualised at timesteps t. The model maintains digit identity and trajectory, key aspects of spatiotemporal hierarchy.

in Fig. 5.1 and Fig. 5.4, where fine-detail errors increase in late output timesteps, although shadows and object reflections appear to be modelled successfully. Additionally, distortions in shape and incorrect modelling of object rotation following collisions (a higher-order hierarchical interaction) are notable. The Moving MNIST examples shown in Fig. 5.9 illustrate correct positioning of the characters, though finer details are smoothed out over time. Additional model output sequences are included in Appendix 5.9.

5.7 Discussion and Structural Analysis

In the following section, we investigate the representations learned by the patch-wise spatial and temporal self-attention layers to understand how PSViT internalises hierarchical spatiotemporal information. Each of these structures is responsible for attending to information across a range of temporal and spatial hierarchies. We assume the GS+T self-attention scheme throughout the remainder of this section. We investigate how these layers attend to information at different model depths, and what sequence-specific, potentially hierarchical, information can be extracted

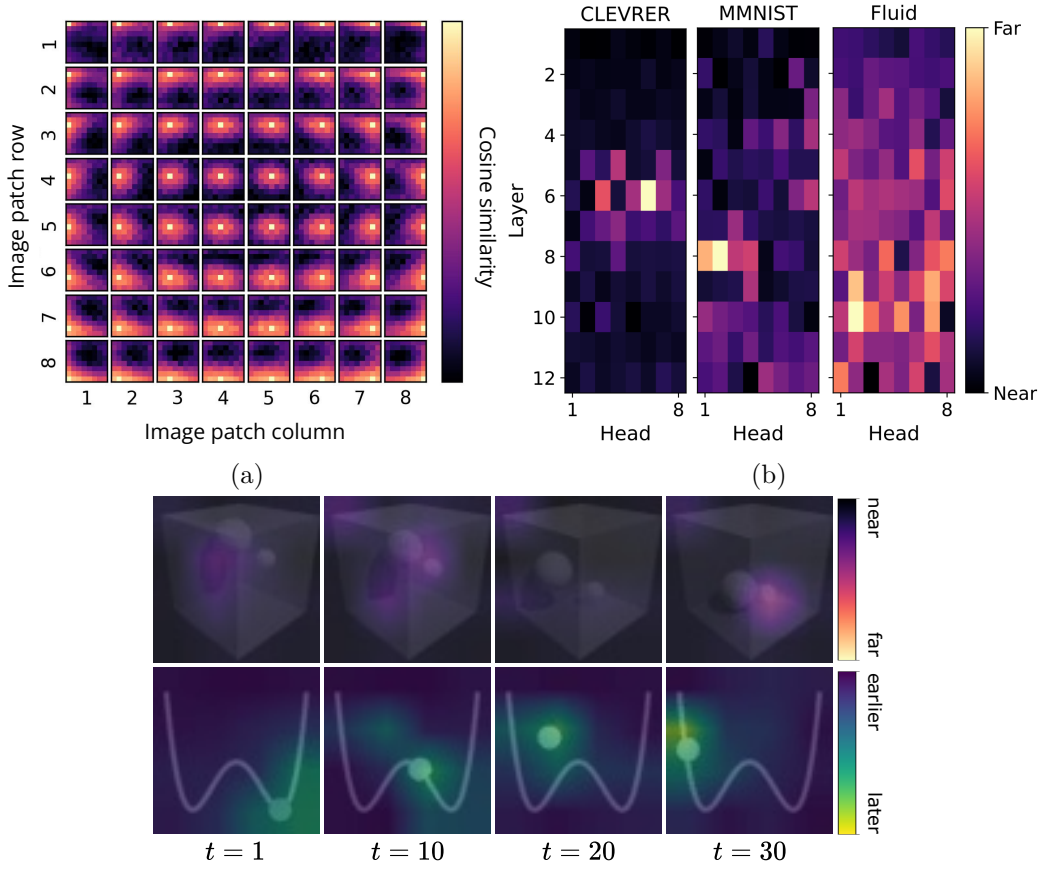


Figure 5.10: **(a)** Visualisation of learned image-patch spatial positional encodings showing cosine similarity between patches. **(b)** Layer-wise spatial self-attention head weightings on CLEVRER, Moving MNIST, and Fluid datasets, with activations normalised by median patch distance scores, such that 0 is the query patch, and 1 is the most distant patch. **(c)** Attention activation heatmaps on example outputs for individual spatial (top, correlated with object interactions) and temporal (bottom, correlated with object velocity) self-attention heads.

from these representations.

5.7.1 Spatial and Temporal Reasoning

Fig. 5.10(a) visualises the learned patch-wise spatial positioning encodings. We observe that the learned positioning encodings do not learn image-symmetrical representations for patches, such that patch-wise similarity does not necessarily scale linearly with patch distance. This can be observed in the corner and edge patch encodings, which show higher similarity to other edge patches than the middle

patches, despite being further away. This is a learned distinction that a sinusoidal absolute encoding would not provide, suggesting a beneficial property for patches to explicitly encode edge-aware information directly into the embeddings, which could be a precursor to identifying more complex spatial hierarchies.

Determining how structural hierarchical information is encoded is an important step in model interpretability. Highlighting the information processed at different model depths can aid in understanding decision making and the internal reasoning processes of the model. To this end, Fig. 5.10(b) shows a heatmap of the relative spatial self-attention distances at each layer and each attention head of the model. We show test-set averages for the CLEVRER, Moving MNIST, and Fluid datasets. We calculate this as the median distance over image patches to which each patch is attending, averaged over all patches. Intuitively, this is the average attended receptive field per head, indicating the scale of spatial hierarchy being processed. We observe a general trend of increased receptive field between the middle and final layers of the model. For instance, the CLEVRER example shows high spatial receptive field concentrated in the middle layers, with smaller receptive fields in the shallow and deep layers. A much broader distribution is observed for the Fluid dataset with an increased receptive field throughout the model, though the middle and deeper layers attend to more distant patches, likely reflecting the more distributed nature of fluid dynamics.

5.7.2 Attention Head Mechanisms

In Fig. 5.10(c) we visualise the activation maps for a single spatial (top) and temporal (bottom) attention head on the 3D ball and Roller datasets, respectively, to understand their role in processing hierarchical events. In the top row, we isolate an attention head correlating with object collisions (a key hierarchical interaction): activation is high when the larger ball collides with the box boundary ($t = 10$), and when the two balls approach and collide ($t = 20$, $t = 30$), followed by reduced activation over these patches post-collision. Additionally, we identify a temporal self-attention head highly correlated with object velocity on the Roller dataset (bot-

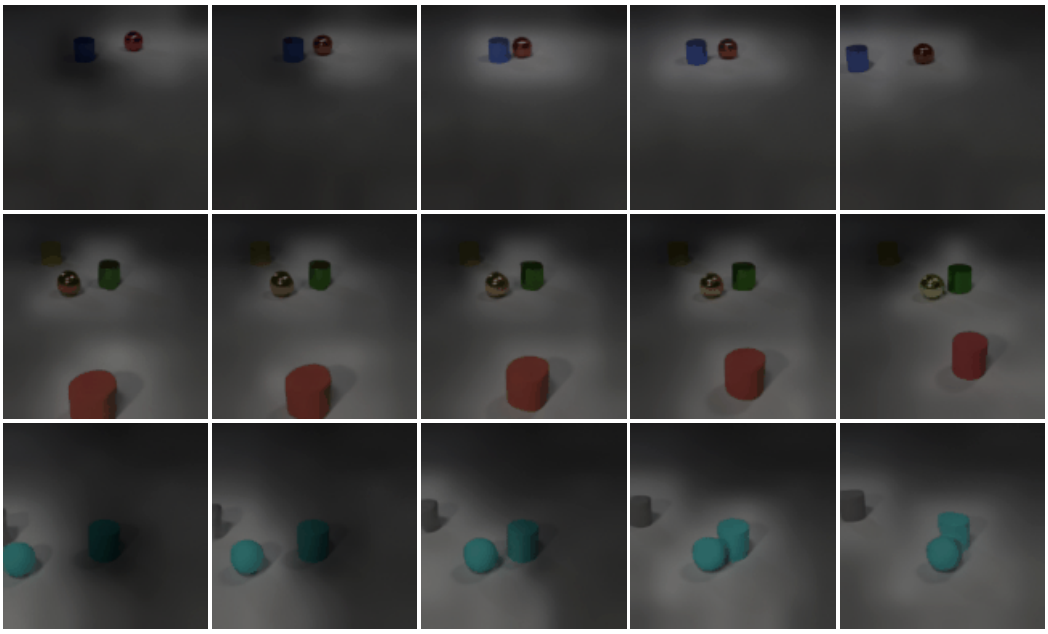


Figure 5.11: Visualisation of register spatial attention activations from randomly selected model outputs on the CLEVRER dataset. The focus on moving objects suggests registers capture high-level dynamic elements of the scene's hierarchy.

tom). We observe attention head activity on patches containing the moving object, where later timesteps are attended to with increased object velocity, suggesting an encoding of this dynamic aspect of the event hierarchy.

Fig 5.11 shows spatial attention activations for register tokens averaged over all layers, clearly highlighting attention focus on moving objects. This supports the hypothesis that register tokens learn to capture and store information about salient, high-level dynamic elements within the scene's spatiotemporal hierarchy.

5.7.3 PDE Dynamics Information Probing

In this section, we examine the internal representations of the space-time layers of our PSViT model to uncover the encoding and extractability of sequence-specific PDE parameters. These parameters, which define the underlying physical laws, represent a form of abstract hierarchical knowledge crucial for accurate object prediction over time and for understanding the model's generalisation capabilities. We train linear probes on top of frozen intermediate representations extracted from each layer of our PSViT model to determine what information about these govern-

Table 5.6: Parameter estimation probing results for PSViT-small, showing Mean Absolute Error (MAE) for predicting underlying PDE parameters (Gravity G or Mass M) from layer representations. Lower MAE indicates better recovery of the PDE parameters. Results are averaged across datasets.

	Regression Error (MAE) ↓			
Model Depth	Roller (G)	Pendulum (G)	Moon (M)	Average
Layer 1	0.47	0.52	0.63	0.54
Layer 2	0.21	0.34	0.45	0.33
Layer 3	0.20	0.31	0.42	0.31
Layer 4	**0.16**	0.27	0.32	0.25
Layer 5	0.17	**0.24**	**0.29**	**0.23**
Layer 6	0.21	0.30	0.33	0.28
Layer 7	0.35	0.41	0.47	0.41
Layer 8	0.62	0.78	0.80	0.73
Concatenation (all layers)	0.15	0.25	0.32	0.24
Scalar mixing (all layers)	0.15	0.28	0.30	0.24
Registers (all layers, concatenated)	**0.14**	0.20	**0.19**	**0.18**
In-distribution best	0.14	0.18	0.19	0.17
Out-of-distribution best	0.16	0.22	0.24	0.21
Baseline (random model)	0.83	0.90	0.91	0.88

ing parameters is extractable. We also experiment with non-linear estimators to explore the limits of what can be learned from internal representations.

Table 5.6 reports our probing results for parameter estimation. We perform gravity estimation on the Roller and Pendulum datasets, and mass estimation on the Moon Orbit dataset. All probing results are an average over a held-out test set of 1,000 sequences. Baseline scores are taken from a randomly initialised equivalent model. Layers used for individual layer probing are the global space-time layers. We observe that for all three tasks, middle-layer representations (Layers 4-5) are most capable of facilitating the extraction of these PDE parameters, with later layers performing worse, presumably as these layers are closer to the output pixel regression layer and thus are more concerned with image generation than abstract parameter representation. Interestingly, we clearly see that the register tokens encode a high degree of this sequence-specific information, as we observe the best overall performance for 2 of the 3 tasks when probing concatenated register token representations from all layers. This, together with the improved video prediction performance when including register tokens (Table 5.4), highlights their value for capturing and utilising high-level PDE information and spatiotemporal context.

Finally, we include probing tests on out-of-distribution parameter ranges not seen during training (details of these ranges can be found in Appendix 5.9). We see only a small increase in MAE averaged over each dataset, from 0.17 (in-distribution) to 0.21 (out-of-distribution), suggesting strong generalisation of the learned PDE dynamics and therefore a genuine internalisation of these hierarchical physical principles rather than mere memorisation of training instances.

5.7.4 Limitations

For three of the simulation datasets, increasing PSViT model size from small to medium has minimal impact on long-horizon divergence performance, though it improves on standard metrics. Further experiments would be needed to show the impact of larger scale models on these more challenging physical reasoning aspects. For stochastic video-generation results on BAIR (evaluated by FVD), our approach falls short compared to latent space models, suggesting higher quality image synthesis with those approaches for such data. This is supported by our qualitative findings that object shapes, although positioned accurately, can distort over time, particularly when rotation is involved. Furthermore, by performing end-to-end training, we do not benefit from the large-scale pre-trained image encoders/decoders often employed by latent space models, which might excel at perceptual fidelity. It remains to be seen if our pixel-space approach can benefit from similar large-scale visual pre-training.

5.8 Conclusion and Future Work

In this chapter, we explored the application of a pure Transformer model, PSViT, for end-to-end autoregressive video prediction, emphasising a simple and interpretable approach to understanding how such models might learn hierarchical spatiotemporal dynamics. Our model leverages several carefully optimised hierarchical priors, including its U-Net style architecture and specific spatiotemporal self-attention layouts, designed to improve the model's capacity for spati-

otemporal reasoning, without requiring complex multi-stage training or dedicated latent feature-learning components. We have shown that PSViT can achieve an improved horizon for physically accurate prediction (by up to 50%) on PDE-driven simulation datasets compared to existing latent-space approaches, while maintaining competitive performance on standard video quality metrics and benchmarks like Moving MNIST. Furthermore, our interpretability studies, including attention analysis and probing for PDE parameters, indicate that the model does internalise significant hierarchical information about the underlying physical dynamics of the scenes, rather than simply memorising pixel patterns. We identified specific network regions and register tokens that correlate with spatiotemporal events and encode sequence-specific PDE parameters, even generalising to out-of-distribution parameter values. This high degree of interpretable, internalised hierarchical reasoning, combined with the model's relative architectural simplicity and effectiveness, highlights the benefit of an end-to-end approach for investigating how Transformers learn from video. This work serves to further our understanding and refinement of attention-based spatiotemporal modelling techniques for tasks requiring deep hierarchical understanding of video content.

Future work will involve developing more sophisticated approaches for evaluating the nuances of physical accuracy beyond object-based pixel distance, which is limited in its application to more complex, non-object-centric dynamics. Additionally, training at larger model scales to accommodate datasets with higher resolution, increased visual fidelity, and greater physical complexity would be a valuable next step. This research paves the way for further focus on simple and effective spatiotemporal modelling, and on building more accurate and interpretable generative models for video content involving complex hierarchical physical systems.

5.9 Epilogue

This chapter has widened the thesis lens from linguistic hierarchies in sentences (Chapter 3) to spatiotemporal hierarchies in video, demonstrating that *hierarchical*

structure again emerges as a key organising principle inside a Transformer, even when processing raw pixel data. Layer-wise analyses and probing of our PSViT model revealed concrete forms of learned hierarchy relevant to physical dynamics:

- **Spatial → temporal → sequence abstraction**. Early blocks encode local patch geometry; mid-blocks bind patches across frames to track object motion; late blocks and dedicated register tokens distil sequence-level variables such as gravitational constant or mass. Linear probes successfully recover these abstract parameters, even when they are out-of-distribution, indicating genuine internalisation of hierarchical physical principles.
- **Register-token memory for global context**. Dedicated learnable tokens were shown to accumulate global context and concentrate physical-parameter information, confirming that Transformers can to store high-level hierarchical variables when provided with an appropriate architectural mechanism.
- **Attention-controlled scale separation for robust prediction**. By alternating local and global attention heads within a U-Net like hierarchy, the model learns to pass coarse layout information upwards while retaining fine detail near the image-reconstruction layers. This architectural prior for hierarchical processing extended physically correct prediction horizons significantly over latent-space baselines.

These results echo and extend the findings from Chapter 3: hierarchy is not an artefact of language alone but a general property that Transformers can discover and internalise. This chapter has thus provided further evidence regarding our first research question (how and where hierarchy is internalised, now in the visual domain) and touched upon the second by showing generalisation of these learned physical parameters beyond the training data distributions. Having investigated this phenomenon in two key modalities, we now move in Chapter 6 to (our final) research question 3: testing whether a *single* next-frame prediction objective can induce compatible and shared hierarchical representations across diverse modalities like text, image, video, and audio, thereby advancing towards the goal of unified, hierarchy-aware multimodal systems.

Appendix

A. Qualitative Outputs

Fig. 5.12 shows CLEVRER example outputs where objects appear from out of view. We observe that, although object position may appear correct, object rotation is often poorly modelled, particularly following a collision. Fig. 5.13 contains examples of the Fluid dataset, and Fig 5.14 shows examples of the Roller, Moon, Pendulum, and 3D Balls datasets.

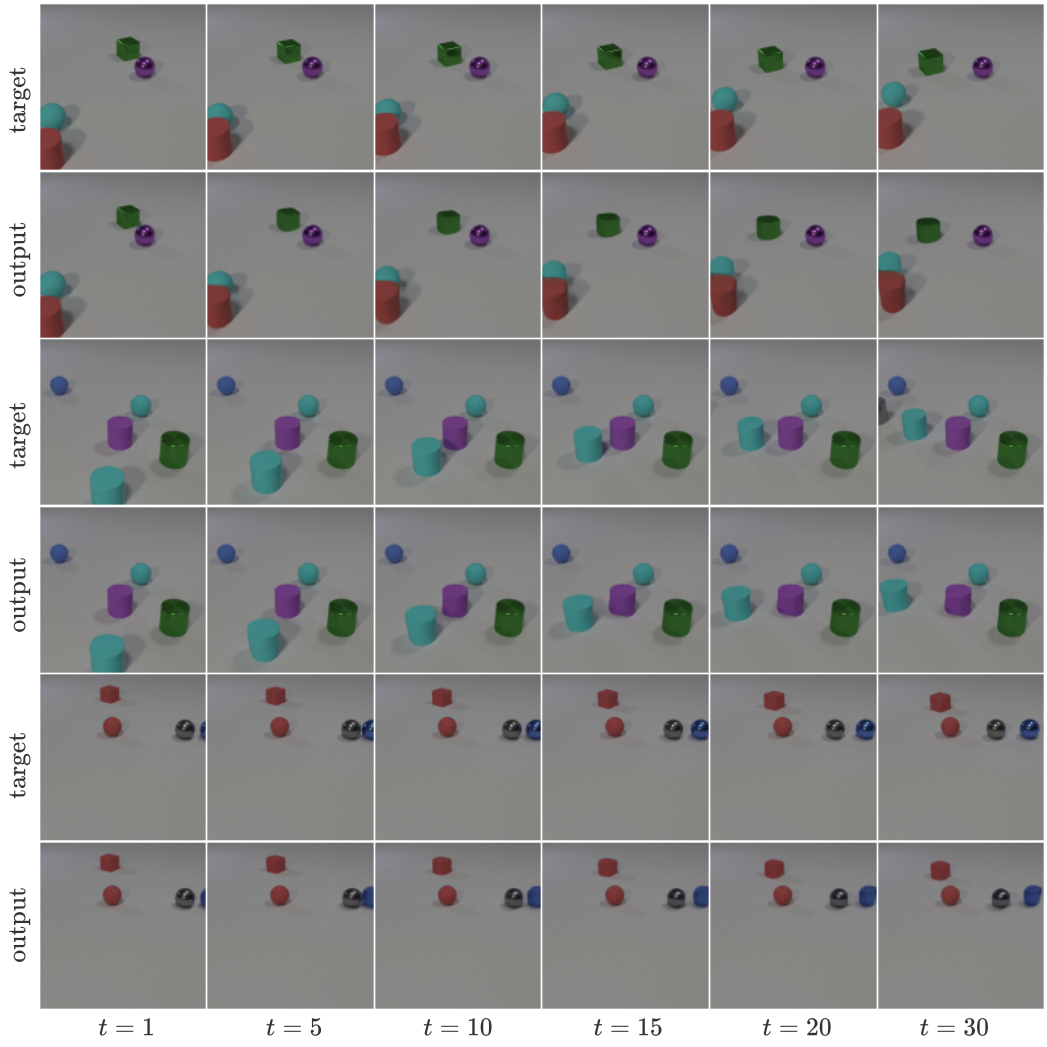


Figure 5.12: Example outputs from our PSViT model trained on the CLEVRER dataset, containing objects entering the scene partially or mostly obscured.

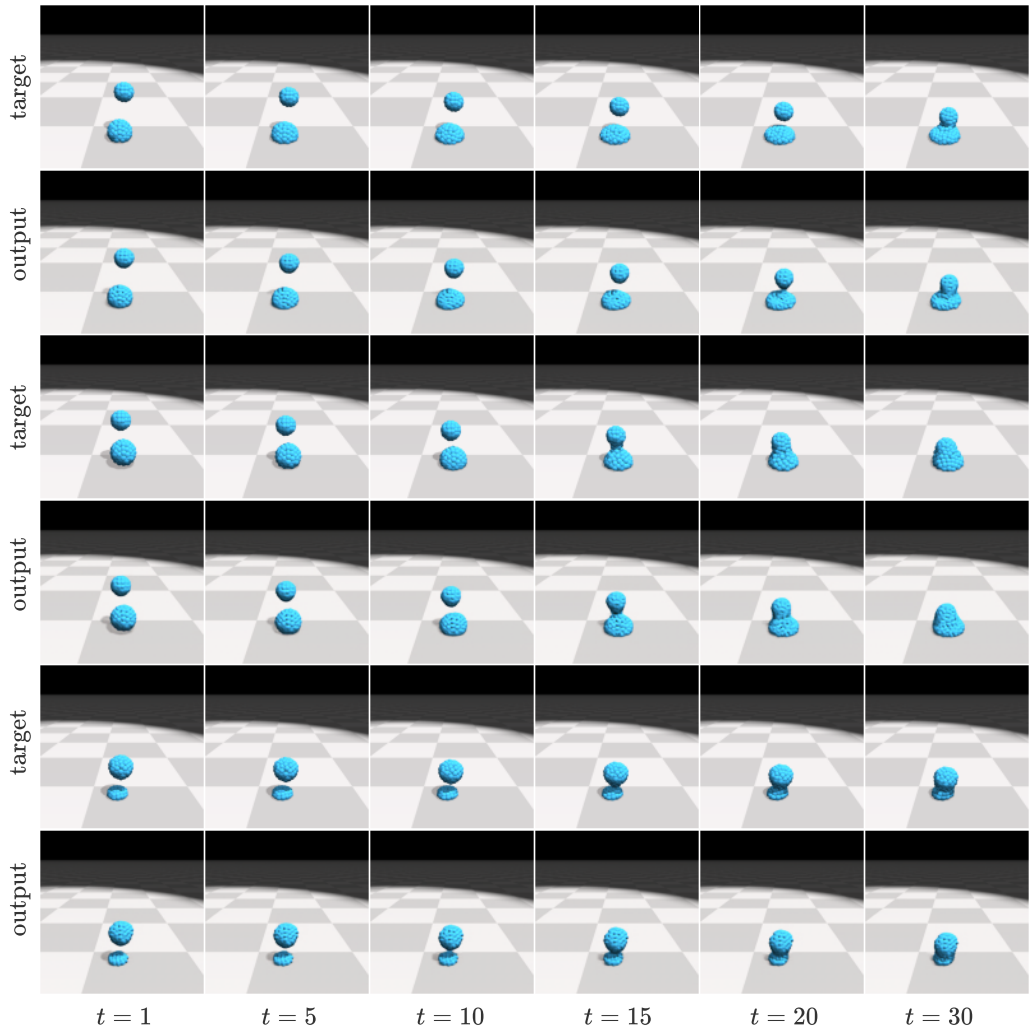


Figure 5.13: Example outputs from our PSViT model trained on the Fluid dataset. Examples shown are randomly sampled.

B. Parameter Range Details for Probing

Table 5.2 in the main text provides the specific in-distribution and out-of-distribution ranges used for the PDE parameter estimation probing tasks for the Roller (Gravity), Pendulum (Gravity), and Moon (Mass) datasets. These out-of-distribution ranges were selected to be adjacent to, but not overlapping with, the ranges seen during the model's training phase, to test for true generalisation of the learned physical principles.

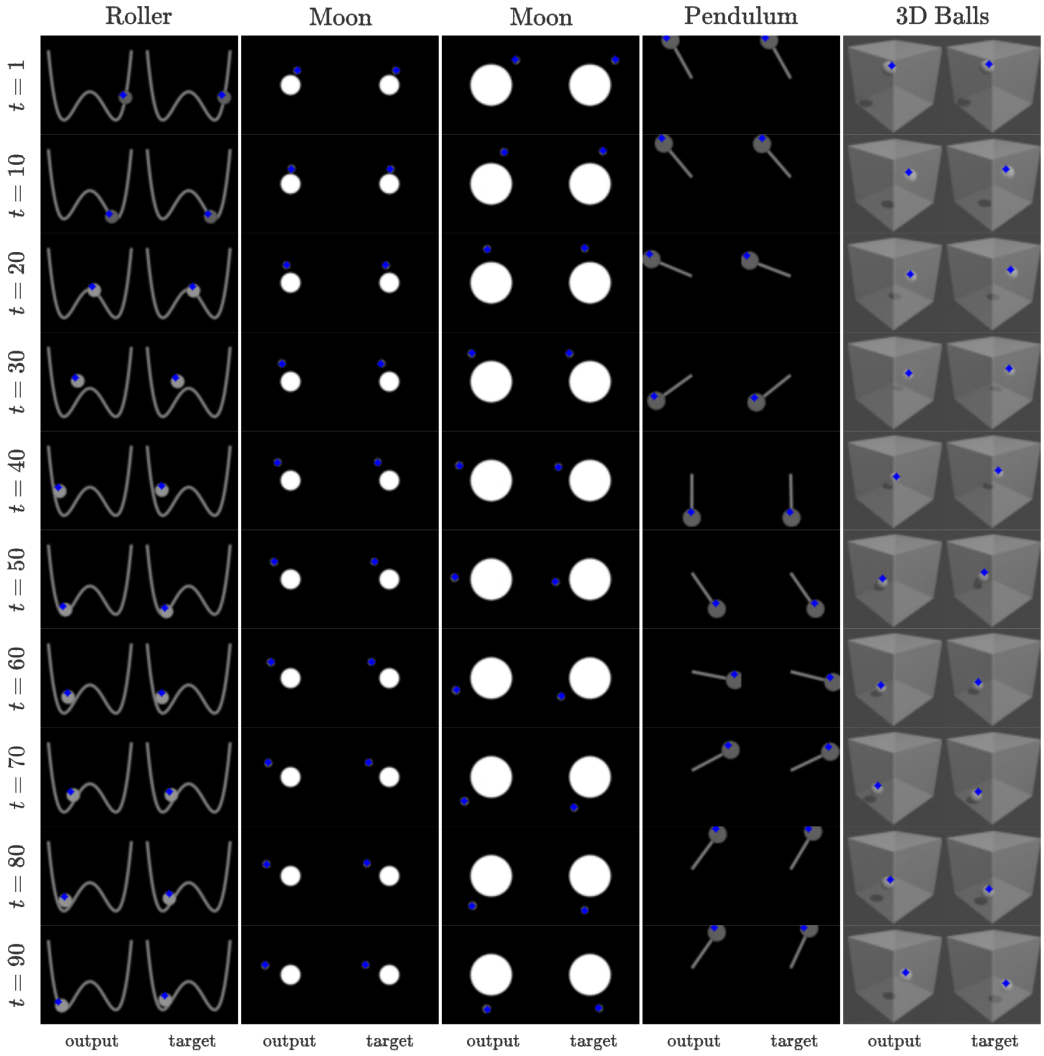


Figure 5.14: Example outputs from our PSViT model trained on the Roller, Moon, Pendulum, and 3D Balls datasets, annotated with object tracking positions. Examples shown are randomly sampled.

Bibliography

Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4.

Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901.

Butcher, J. C. (1996). A history of runge-kutta methods. *Applied numerical mathematics*, 20(3):247–260.

Castrejon, L., Ballas, N., and Courville, A. (2019). Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7608–7617.

Chen, M., Radford, A., Wu, J., Jun, H., Dhariwal, P., Luan, D., and Sutskever, I. (2020). Generative pretraining from pixels. In *ICML*.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2023). Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. (2023). Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10850–10869.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. *ICCV*, pages 764–773.

Darcet, T., Oquab, M., Mairal, J., and Bojanowski, P. (2024). Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*.

de Bezenac, E., Pajot, A., and Gallinari, P. (2018). Deep learning for physical processes: Incorporating prior scientific knowledge. In *ICLR*.

Denton, E. and Fergus, R. (2018). Stochastic video generation with a learned prior. In *International conference on machine learning*, pages 1174–1183. PMLR.

Donahue, J. and Simonyan, K. (2019). Large scale adversarial representation learning. In *NeurIPS*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In *NIPS*.

Ebert, F., Finn, C., Lee, A. X., and Levine, S. (2017). Self-supervised visual planning with temporal skip connections. *CoRL*, 12(16):23.

Farazi, H., Nogga, J., and Behnke, S. (2021). Local frequency domain transformer networks for video prediction. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE.

Finn, C., Goodfellow, I., and Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 64–72, Red Hook, NY, USA. Curran Associates Inc.

Finn, C. and Levine, S. (2017). Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793.

Gao, Z., Tan, C., Wu, L., and Li, S. Z. (2022). Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3170–3180.

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hong, W., Ding, M., Zheng, W., Liu, X., and Tang, J. (2023). Cogvideo: Large-scale pretraining for text-to-video generation via transformers. In *The Eleventh International Conference on Learning Representations*.

Horé, A. and Ziou, D. (2010). Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369.

Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., and Corrado, G. (2023). Gaia-1: A generative world model for autonomous driving.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Kohl, G., Chen, L., and Thuerey, N. (2023). Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *arXiv*.

Kolter, J. Z. and Manek, G. (2019). Learning stable deep dynamics models. *Advances in neural information processing systems*, 32.

Le Guen, V. and Thome, N. (2020). Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11471–11481.

Li, Y., Wu, J., Tedrake, R., Tenenbaum, J. B., and Torralba, A. (2019). Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. In *ICLR*.

Ming, R., Huang, Z., Ju, Z., Hu, J., Peng, L., and Zhou, S. (2024). A survey on video prediction: From deterministic to generative approaches. *arXiv preprint arXiv:2401.14718*.

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., and Argyros, A. (2020). A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2806–2826.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rakhimov, R., Volkhonskiy, D., Artemov, A., Zorin, D., and Burnaev, E. (2021). Latent video transformer. In *VISIGRAPP*.

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. (2014). Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*.

Razavi, A., Van den Oord, A., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer.

Seo, M., Lee, H., Kim, D., and Seo, J. (2023). Implicit stacked autoregressive model for video prediction.

Seo, Y., Lee, K., Liu, F., James, S., and Abbeel, P. (2022). Harp: Autoregressive latent video prediction with high-fidelity image generator. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3943–3947. IEEE.

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.-Y., Wong, W.-k., and WOO, W.-c. (2017). Learning stable deep dynamics models. In *NeurIPS*, volume 30.

Srivastava, N., Mansimov, E., and Salakhutdinov, R. (2015). Unsupervised learning of video representations using lstms. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 843–852. JMLR.org.

Sønderby, C. K., Espeholt, L., Heek, J., Dehghani, M., Oliver, A., Salimans, T., Agrawal, S., Hickey, J., and Kalchbrenner, N. (2020). Metnet: A neural weather model for precipitation forecasting.

Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. (2017). Accelerating eulerian fluid simulation with convolutional networks. In *ICML*, page 3424–3433.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. (2019). Fvd: A new metric for video generation. In *DGS@ICLR*.

van Amersfoort, J. R., Kannan, A., Ranzato, M., Szlam, A., Tran, D., and Chintala, S. (2017). Transformation-based models of video sequences. *ArXiv*, abs/1701.08435.

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1747–1756. JMLR.org.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. In *NIPS*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, R., Walters, R., and Yu, R. (2022). Meta-learning dynamics forecasting using task inference. *Advances in Neural Information Processing Systems*, 35:21640–21653.

Wang, Y., Wu, J., Long, M., and Tenenbaum, J. B. (2020). Probabilistic video prediction from noisy data with a posterior confidence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10830–10839.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022). Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Weissenborn, D., Täckström, O., and Uszkoreit, J. (2020). Scaling autoregressive video models. In *International Conference on Learning Representations*.

Wen, Y., Zhao, Y., Liu, Y., Jia, F., Wang, Y., Luo, C., Zhang, C., Wang, T., Sun, X., and Zhang, X. (2023). Panacea: Panoramic and controllable video generation for autonomous driving.

Winterbottom, T., Hudson, G. T., Kluvanec, D., Slack, D., Sterling, J., Shentu, J., Xiao, C., Zhou, Z., and Moubayed, N. A. (2024). The power of next-frame prediction for learning physical laws.

Wu, C., Liang, J., Ji, L., Yang, F., Fang, Y., Jiang, D., and Duan, N. (2022). Nüwa: Visual synthesis pre-training for neural visual world creation. In *European conference on computer vision*, pages 720–736. Springer.

Wu, J., Lim, J. J., Zhang, H., Tenenbaum, J. B., and Freeman, W. T. (2016). Physics 101: Learning physical object properties from unlabeled videos. In *BMVC*.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*.

Xing, Z., Feng, Q., Chen, H., Dai, Q., Hu, H., Xu, H., Wu, Z., and Jiang, Y.-G. (2023). A survey on video diffusion models.

Yan, W., Zhang, Y., Abbeel, P., and Srinivas, A. (2021). Videogpt: Video generation using vq-vae and transformers.

Yang, X., Yang, X., Liu, M.-Y., Xiao, F., Davis, L. S., and Kautz, J. (2019). Step: Spatio-temporal progressive learning for video action detection. In *CVPR*.

Yılmaz, M. A. and Tekalp, A. M. (2021). Dfpn: Deformable frame prediction network. In *2021 IEEE international conference on image processing (ICIP)*, pages 1944–1948. IEEE.

Yin, Y., Kirchmeyer, M., Franceschi, J.-Y., Rakotomamonjy, A., and patrick gallinari (2023). Continuous PDE dynamics forecasting with implicit neural representations. In *The Eleventh International Conference on Learning Representations*.

Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A. G., Yang, M.-H., Hao, Y., Essa, I., et al. (2023). Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469.

Zhang, C., Zhang, C., Zhang, M., and Kweon, I. S. (2023). Text-to-image diffusion models in generative ai: A survey. *arXiv preprint arXiv:2303.07909*.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhao, S., Zhang, Y., Cun, X., Yang, S., Niu, M., Li, X., Hu, W., and Shan, Y. (2024). CV-VAE: A compatible video VAE for latent generative video models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., and You, Y. (2024). Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*.

Zhou, Y., Dong, H., and El Saddik, A. (2020). Deep learning in next-frame prediction: A benchmark review. *IEEE Access*, 8:69273–69283.

Chapter 6

Multimodal Multi-Task Hierarchical Reasoning

Building upon our explorations of hierarchical structures in language (Chapter 3) and vision (Chapter 5), this final empirical chapter addresses our third research question: whether a single hierarchical framework can unify diverse modalities such as text, image, video, and audio. We investigate this by reformulating a range of multimodal tasks into a common next-frame prediction problem on a shared visual canvas. This unified input is processed by a single autoregressive Transformer, adapted from the PSVIT architecture (introduced in Chapter 5), thereby avoiding modality-specific encoders and late-fusion mechanisms. Our central aim is to determine if this approach can enable the model to learn both modality-specific sub-hierarchies (e.g., for language or visual events) and their effective integration into a cohesive, cross-modal hierarchical understanding. This chapter demonstrates the viability of such task reformulation, presenting evidence of competitive performance on several benchmarks and analysing how the model appears to develop multi-stage, cross-modal hierarchical representations. These findings offer insights into creating more universal, hierarchy-aware AI systems and conclude the main investigative arc of this thesis.

The chapter proceeds as follows. Section 6.1 further introduces the motivation for a unified multimodal approach and details this chapter's specific contributions to-

wards developing systems capable of learning shared hierarchical abstractions. We then review relevant literature in multimodal learning, task reformulation, next-frame prediction, and the visual representation of text in Section 6.2. Section 6.3 describes the diverse task reformulation strategies for various modalities and the adaptation of the PSViT architecture for this unified framework, along with training specifics. The empirical results across a range of multimodal tasks, including an analysis of attention mechanisms to probe cross-modal hierarchical understanding, are presented in Section 6.4, which also discusses the limitations of this initial exploration and avenues for future work.

6.1 Introduction

Having explored hierarchical reasoning in unimodal contexts (language in Chapter 3; video in Chapter 5), we now address the challenge of extending these principles to multimodal settings, a key component of our third research question. The ability to process and integrate information from multiple modalities such as text, image, audio, and video has proven crucial for a range of complex tasks, including cross-modal retrieval (Wang et al., 2024), visual question answering (Wu et al., 2017; Zou and Xie, 2020; Lu et al., 2023), and caption generation (Agarwal and Verma, 2023). The capacity to understand and reason about information from different sensory inputs simultaneously mirrors human perception and cognition, making the development of effective multimodal models essential for advancing artificial intelligence towards more generalised and versatile capabilities.

The predominant approach to building multimodal models involves using separate encoders for each modality. For example, in vision-and-language models, a CNN or a vision transformer (ViT) (Dosovitskiy et al., 2021) might be used to encode images, while an RNN or another Transformer is used to encode text. These modality-specific encoded representations are then fused at a later stage, typically via concatenation or attention mechanisms, to enable the model to perform joint reasoning. While effective for specific tasks, this paradigm has inherent limita-

tions. It necessitates careful design choices for each modality and often struggles with scalability and flexibility, especially when extending to new combinations of modalities or tasks not seen during training (Baltrušaitis et al., 2018; Li et al., 2021; Jaegle et al., 2021), potentially hindering the development of a more integrated and potentially universal mechanism for learning shared hierarchical representations.

To address these challenges, we propose a novel approach centred around task reformulation. Task reformulation is a well-established paradigm in NLP that facilitates multitask learning by transforming diverse tasks into a common format (McCann et al., 2018). A prominent example is prompt-based learning, where NLP tasks are reimagined as instructions or prompts to a generative LLM. This paradigm allows a wide range of NLP tasks, such as translation, summarisation, and question answering, to be represented uniformly in a chat message format (OpenAI, 2023). By doing so, it leverages the generalisation capabilities of large pre-trained models, enabling them to perform multiple tasks with minimal task-specific customisation. Our work seeks to find a similar common framework for studying how hierarchical reasoning principles apply and transfer across different modalities.

In this work, we extend the concept of task reformulation beyond NLP to the realm of multimodal learning. Specifically, we propose a framework in which tasks across various modalities are reformulated as a unified next-frame prediction problem. This approach simplifies the design of multimodal models by enabling them to handle diverse input types — text, images, audio, or video — using a single, coherent mechanism. By representing all tasks as next-frame prediction, we create a shared interface for the model to learn and reason about information across modalities. This not only facilitates the integration of new modalities with minimal effort but also enhances the model's ability to transfer knowledge across different tasks and domains. Ultimately, analogous to how LLMs serve as foundation models for NLP, our aim is to explore whether this unified predictive framing can foster the development of foundational models capable of learning shared hierarchical abstractions across diverse modalities.

The main contributions of this chapter are as follows:

- We propose a task-reformulation method and demonstrate how a range of multimodal tasks can be represented as next-frame prediction in a common visual canvas.
- We explore how this reformulation allows a single Transformer-based model, adapted from the PSViT architecture (Chapter 5), to solve text, image, video, and audio processing tasks without any modality-specific input encoders.
- **Demonstrating viable multimodal learning and evidence of emergent cross-modal hierarchical representations**: Our experimental results show that this unified model achieves competitive performance compared to single-task models trained under similar conditions. Furthermore, our analysis of its internal mechanisms, particularly attention patterns, provides initial evidence for the development of shared, multi-stage hierarchical representations that integrate information across diverse modalities.

6.2 Related work

Multimodal Learning With the rise of Transformers, LLMs, and very large-scale pre-training, new paradigms for harmonising different modalities have been pushed to the forefront of multimodal learning. UNITER (Chen et al., 2019) uses a single universal Transformer to process both image and text inputs into a joint embedding space, with image representations prepended to the beginning of the input embeddings to the Transformer. However, the image inputs are first processed by an RNN to obtain an embedding such that it is compatible with the textual inputs. Our framework, in contrast, has no need to preprocess input modalities differently as they are unified into a single visual input space, theoretically capable of fully representing all multimodal interactions using a single backbone architecture (adapted from PSViT, Chapter 5) to process these reformulated inputs, encouraging the learning of shared hierarchical features. Baevski et al. (2022)

propose data2vec, a framework for instilling features from multiple modalities into a single latent representation. Where data2vec seeks to process raw text, images, and audio using the same method, our work here reformulates the task inputs from multiple modalities into the *same visual input medium*. UniT (Hu and Singh, 2021), a multimodal multitask Transformer, uses separate encoders for each modality followed by a shared *decoder*. Our work uses a single visual encoder (as part of the unified PSViT architecture) for our unified visual input paradigm. The most powerful state-of-the-art multimodal models such as FLAVA (Singh et al., 2021) and GPT-4V(ision) (OpenAI, 2023) integrate vision inputs via a patch-wise processing pipeline through a visual Transformer fully integrated into the training pipeline. Both the architectural design and scale at which these foundation models are trained yield revolutionary performance at a variety of open-ended multimodal tasks. Nonetheless, though these foundation models are the closest that benchmarks have practically come to seamlessly integrating images with textual inputs, fundamentally the input modalities are often handled by distinct initial processing stages before fusion. Our work here aims to bridge this remaining gap by reformulating typically supra-visual tasks into a visual input domain with *no loss of information* (*i.e.*, text and audio can be fully represented as image sequences). This reformulation allows us (for the first time, to the best of our knowledge) to explore the generative pre-training capacity for multimodal information within one truly unified representational and predictive framework.

Task Reformulation Task reformulation involves enabling a single model to solve multiple tasks by converting them into a single 'supertask'. This technique is common in NLP, where a range of tasks such as sentiment classification and coreference resolution have been reformulated as span-extraction (Keskar et al., 2019), or question answering (McCann et al., 2018), among others.

The recent trend of prompt-based learning uses language modelling on prompt-answer sequences as the 'supertask' (OpenAI, 2023). In this paradigm, the model is trained to follow an instruction describing the task and provide a response. In this

way, any NLP task can be reformulated as the language modelling 'supertask'. In our work, we extend this concept, exploring how next-frame prediction can be used as a 'supertask' to unite multiple modalities and foster the learning of transferable hierarchical representations.

Next-frame Prediction CNN-based deep learning architectures for the autoregressive generation of future video frames have been steadily improving over the last decade (Ranzato et al., 2014; van Amersfoort et al., 2017; Yilmaz and Tekalp, 2021; seok Seo et al., 2023). Transformer-based models trained at scale, such as VideoGPT (Yan et al., 2021), have substantially improved the performance, quality, and fidelity of long and short-term future frame predictions. Modern diffusion-based frame prediction models are now able to generate photorealistic outputs (Gupta et al., 2023). Our aim in this work is not to propose a new state-of-the-art video prediction architecture itself, but rather to leverage a capable autoregressive framework (based on PSViT from Chapter 5) as a testbed for our task reformulation and analysis of multimodal hierarchical learning.

Visual Representation of Text Inspired by the success of unsupervised next-word prediction with language models, learning the next pixel of images was proposed (Chen et al., 2020) and shown to be able to learn image representations. Learning textual semantics with vision models, by rendering text into images, has drawn increasing attention (Rust et al., 2022; Tai et al., 2024; Gao et al., 2024) due to the drawbacks of tokenisation in traditional language models and limitations in cross-lingual transferability. Xiao et al. (2024) show that vision models pre-trained on rendered images have stronger robustness to typo and word-order shuffling perturbations, and their representations display better isotropy on out-of-distribution (OOD) languages. In this work, we extend this approach when reformulating NLP tasks to video, as well as for other modalities which output text, as part of our unified visual canvas.

6.3 Method

Our proposed framework consists of two key components: (1) methods for reformulating a diverse range of input and output modalities into the single task of next-frame prediction; and (2) a pure Transformer-based model architecture adapted for this unified task. We introduce these components in the following sections.

6.3.1 Reformulation

The datasets we use are carefully selected to cover a diverse range of input and output modalities, illustrating the versatility of our approach in reformulating various tasks as next-frame prediction problems. Our selection spans tasks from text and image classification to more complex audio, video, and multimodal tasks, enabling us to evaluate the model's ability to generalise across different data types and task requirements. By converting each task into a uniform format: a 64×64 RGB video sequence, we create a consistent framework where the model treats every input and output as sequential frames, simplifying the multimodal learning process and providing a basis for learning shared hierarchical representations. For each task, we insert a separator token (|) rendered as a distinct frame between the input and output frames to clearly delineate where the input ends and the prediction begins.

In tasks involving textual data, we use a simple tokeniser (splitting on spaces and also on punctuation) to break down the text into individual tokens, which we then render as video frames in the sequence. Each token is represented in a consistent format: a fixed-width font scaled to fill the 64×64 frame, ensuring clarity and compatibility with our video-based input structure. This approach allows the model to read and predict text as it would any other frame, effectively bypassing the need for traditional text-based tokenisation while integrating text seamlessly with other modalities. By converting each token to a visual format, we enable cross-modal knowledge transfer, allowing the model to process text, images, and other modalities through a shared, frame-based learning paradigm.

Text-to-Text The SST2 dataset (previously used in Chapter 3 for ancestor sentiment classification and probing) is a widely-used benchmark for sentiment classification, consisting of thousands of movie review excerpts labelled with binary sentiment labels (positive or negative) (Socher et al., 2013). We use the text-encoding method described above, rendering each token as a video frame using a fixed-width font (Figure 6.1).

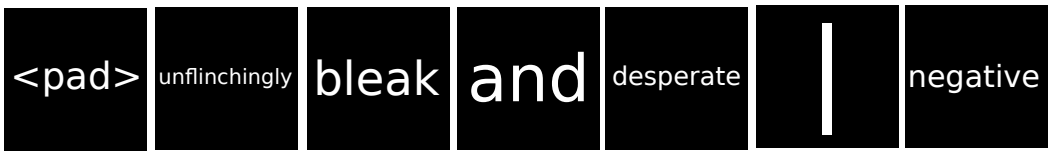


Figure 6.1: Truncated example of the SST2 sentiment dataset rendered as a video. Each square is a frame of the video sequence.

Image-to-Text As a simple test of image recognition ability, we employ the CIFAR-10 dataset. CIFAR-10 is a benchmark for image classification and consists of 60,000 colour images, each with dimensions of 32×32 pixels (Krizhevsky et al., 2009). These images are equally divided into ten distinct, mutually exclusive classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

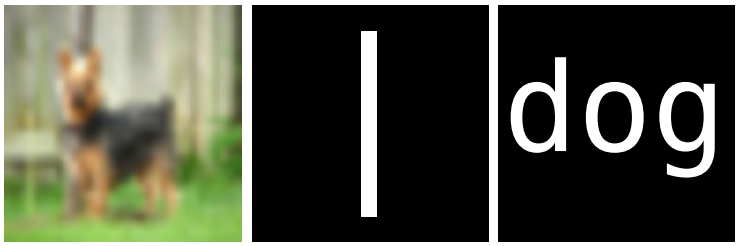


Figure 6.2: Truncated example of the CIFAR10 image classification dataset rendered as a video. Each square is a frame of the video sequence.

In our reformulation approach, each CIFAR-10 image is resized to 64×64 pixels to fit our standardised input frame size. Following the resized image, we insert a separator frame, which visually signifies the end of the input and the beginning of the output sequence, followed by a frame containing the class label text rendered as an image (Figure 6.2).

Video-to-Text We utilise the TinyVIRAT action classification dataset to rigorously assess the model's capacity for understanding video content (Demir et al.,

Figure 6.3: Truncated example of the TinyVIRAT video classification dataset rendered as a video. Each square is a frame of the video sequence.

2021). The TinyVIRAT dataset comprises 7,663 training and 5,166 testing examples, each annotated with one or more of 26 distinct action labels that describe the activities depicted within low-resolution video sequences. This dataset is particularly noteworthy for its multi-label nature, allowing for nuanced action recognition; for instance, a video sequence might feature a man walking while carrying a backpack, which would be simultaneously labelled with both "walking" and "carrying" actions. This characteristic enables the evaluation of the model's ability to recognise and differentiate between overlapping actions occurring in dynamic environments. As discussed in the background Chapter on hierarchical reasoning in video (§ 2.4.3), establishing the correct high-level action classification requires identifying these simpler sub-actions, together forming a spatiotemporal hierarchy. To reformulate the task for next-frame prediction, we represent each video sequence by first presenting the input video frames, followed by a separator token frame, and concluding with a comma-separated list of action labels (rendered as text frames) that correspond to the activities depicted in the video (Figure 6.3).

Video+Text-to-Text To explore the integration of multiple modalities within a single input, we leverage the CLEVRER (CoLlision Events for Video REpresentation and Reasoning) dataset (as previously explored in Chapter 5 for video prediction of colliding objects), which is designed for video question answering (VQA) (Yi et al., 2019). This dataset comprises synthetic video sequences depicting rendered 3D objects interacting through collisions, movements, and occlusions within a simple, controlled environment. Each video is paired with questions that require the model to understand and reason about these visual events. The questions in CLEVRER span various reasoning tasks, including descriptive, explanatory, and predictive questions about the events in the video, such as identifying

Figure 6.4: Truncated example of the CLEVRER task rendered as a video. Each square is a frame of the video sequence.

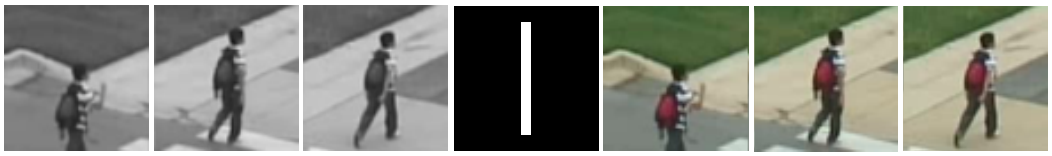


Figure 6.5: Truncated example of the colorization task rendered as a video. Each square is a frame of the video sequence.

specific objects, understanding object interactions, and predicting future states.

In our framework, we represent each question and corresponding video as a unified sequence of frames. Each input sequence begins with frames of the CLEVRER video sequence subsampled to 4 frames. This is followed by the question text (rendered as frames), the separator token frame, and finally the answer text rendered as frames (Figure 6.4). Similarly, we use the frame QA task from the TGIF-QA dataset (Jang et al., 2019), containing animated GIFs with QA pairs.

To keep the sequence length manageable, we exclude complex counterfactual questions and multiple-choice answers from the CLEVRER dataset (using only the 'descriptive' subset), as these would require additional frames and could lead to overly long sequences that complicate training. Correctly answering questions concerning observed object interactions over time requires sophisticated understanding and encoding of spatiotemporal events.

Video-to-Video We employ two tasks to test video-to-video performance. Firstly, we convert the TinyVIRAT videos into greyscale and test the model's ability to colourise the output. The sequence consists of the greyscale-converted video sequence, the separator frame, then the original RGB video sequence (Figure 6.5).

Additionally, we test the model's ability to perform object tracking. The Large-scale Single Object Tracking dataset (LaSOT) consists of 1,550 video sequences hand-annotated with bounding boxes following an object of interest (Fan et al.,

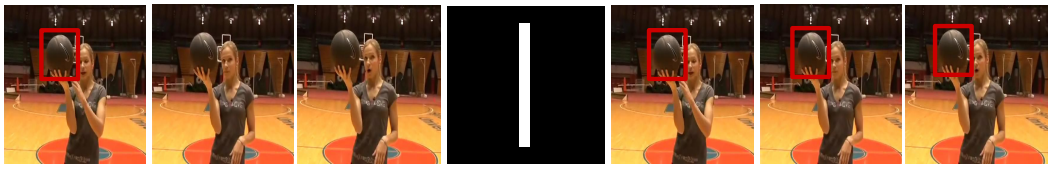


Figure 6.6: Truncated example of the LaSOT object tracking task rendered as a video. Each square is a frame of the video sequence.

2019). We reformulate this task by converting it into a sequence which consists of the first frame of the video with the bounding box overlayed, the rest of the input video sequence, the separator frame, and finally the full video sequence including the overlayed bounding box for all frames (Figure 6.6).

Audio-to-Text To demonstrate the ability to process audio data, we utilise the AudioMNIST dataset, which contains 30,000 audio recordings of spoken digits ranging from zero to nine (Becker et al., 2024). These recordings are spoken by multiple speakers with varying accents and intonations, providing a diverse dataset that challenges the model to recognise and classify spoken language across different vocal characteristics.

In our framework, each audio sample is preprocessed into a spectrogram, which is used as the first frame in the video sequence, representing the audio content visually for consistency with other modalities. Following the spectrogram, a separator token frame is added, and the label digit is represented in the final frame as text (Figure 6.7). This structured sequence allows the model to interpret audio information in a form similar to visual or textual data, enabling it to predict the spoken digit without using a specialised audio encoder.

6.3.2 Video Prediction Model

We adapt the PSVIT architecture, introduced and detailed in Chapter 5 (§ 5.4), for all tasks in this chapter. This allows us to leverage a consistent Transformer-based framework designed for end-to-end video processing directly in pixel space. The core principles of PSViT, including its U-Net style structure for capturing multi-scale features and its patch-based spatiotemporal self-attention mechanisms,

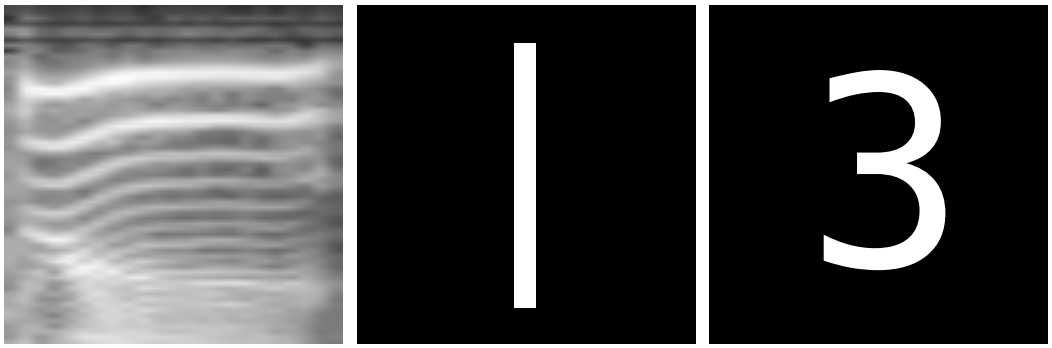


Figure 6.7: Example of the AudioMNIST audio classification dataset rendered as a video. Each square is a frame of the video sequence.

are retained. This continuity allows us to investigate whether a model architecture shown to learn hierarchical physical dynamics in unimodal video can extend its capabilities to learn shared hierarchical representations across diverse modalities when presented through a unified next-frame prediction task.

Briefly, the adapted PSViT model takes as input a sequence of T video frames, where each frame is a $C \times H \times W$ tensor (here, 64×64 RGB frames). Each frame is partitioned into non-overlapping patches, which are linearly embedded. The sequence of embeddings is then processed by a series of Transformer blocks employing spatiotemporal self-attention. Spatial self-attention operates within each frame, while causal temporal attention operates across frames for corresponding patch locations, ensuring autoregressive prediction. The U-Net style architecture involves patch merging and unmerging operations to process features at different resolutions, facilitating the capture of hierarchical information. The model is trained end-to-end to predict the next frame in the sequence.

6.3.3 Training Details

We train our adapted PSViT model on each dataset independently, with no language model or image pre-training on external large-scale datasets, to determine the feasibility of our task reformulation approach and the model's capacity to learn from scratch within this unified framework. We train end-to-end with a Multi-Scale Structural Similarity Index Measure (MS-SSIM) loss (Wang et al., 2004), with a constant learning rate of 3×10^{-4}, and a batch size that varies per task

Task	Input Len	Target Len	Batch Size
Text Classification	21	1	16
Image Classification	2	1	128
Video Classification	11	1	16
Audio Classification	2	1	256
Video QA	21	1	16
Object Tracking	21	20	8
Video Colorization	11	10	16

Table 6.1: Training configurations for each task. Input/Target Lengths are in number of frames.

(Table 6.1) depending on sequence length and memory constraints. We use the AdamW (Loshchilov and Hutter, 2017) optimiser using default parameters, and set dropout to 0.1 for all layers except the final output layer. All models are evaluated on checkpoints corresponding to the best MS-SSIM validation performance, with all models trained on a single NVIDIA A100 GPU for a maximum of 7 GPU days. The input/output lengths and batch sizes are detailed in Table 6.1.

6.4 Experiments

We train our adapted PSViT model on each task independently (we leave training on all tasks jointly in a multitask setting as future work). For tasks that output text, we perform Optical Character Recognition (OCR) on the generated output frames using the open-source Tesseract OCR engine*. For tasks that have a fixed vocabulary (e.g., classification tasks), the OCR text is matched to the closest vocabulary word using Levenshtein distance to minimise OCR-related errors. For classification tasks, the resulting text is evaluated using F1-score and Accuracy.

The object tracking task is evaluated by extracting the bounding box in the outputted frames (rendered as a distinct colour) and then comparing these to the labelled boxes using Intersection over Union (IoU).

For video colorisation, we evaluate both the colorisation performance and temporal consistency across the video sequence. Measuring both is particularly important

*https://github.com/tesseract-ocr/tesseract

here, as the model could achieve consistency by simply repeating the greyscale input sequence as the output. To this end, we measure both PSNR and the colorfulness measure proposed in Liu et al. (2024) which serves as a simple measure of colour diversity. Finally, we use the Colour Distribution Consistency index (CDC) (Liu et al., 2024) to evaluate the temporal consistency of the output sequence. CDC computes the Jensen-Shannon (JS) divergence of the colours between a temporal offset of t frames:

$$CDC_t = \frac{1}{3 \times (N_{frames} - t)} \sum_{c \in \{r,g,b\}} \sum_{i=1}^{N_{frames}-t} JS(P_c(I^i), P_c(I^{i+t}))$$

where N_{frames} is the video sequence length and $P_c(I^i)$ is the normalised probability distribution of frame I^i on colour channel c, calculated from the image histogram. To measure colour consistency over a range of time intervals, the overall measure is:

$$CDC = \frac{1}{3}(CDC_1 + CDC_2 + CDC_4).$$

6.5 Results

The results across each task are presented in Table 6.2, while Figure 6.8 and Figure 6.9 illustrates several sample outputs produced by our model, offering a qualitative perspective on its performance.

For the SST-2 text classification task, our model achieves an F1-score of 76.8. Although this is lower than the state-of-the-art performance of 91.3 reported by Zhong et al. (2023), that result was achieved using extensive pre-training and complex fine-tuning processes. Our model is trained directly on the reformulated SST-2 dataset without any external pre-training, making its performance more comparable to other models trained from scratch or simpler baselines (e.g., BERT-base trained from scratch on SST-2 often yields accuracy around 80-85%, though direct F1 comparison varies). Some portion of this lower score can also be attributed to truncating inputs over 20 tokens (limiting the evaluation to shorter inputs of 20 tokens or fewer increases the F1-score to 80.0).

Task	Dataset	Model	PSNR ↑	SSIM ↑	F1 ↑	Acc ↑	IoU ↑	CDC ↓	Colorfulness ↑
Text Classification	SST-2	Ours (PSViT adapted)	41.8	0.987	76.8	75.5	-	-	-
		BERT-base (from scratch baseline)	-	-	-	81.2	-	-	-
		Vega v1 (Zhong et al., 2023)†	-	-	91.3	-	-	-	-
Image Classification	CIFAR-10	Ours (PSViT adapted)	45.7	0.959	89.1	89.1	-	-	-
		ViT-base (from scratch baseline)	-	-	71.3	71.3	-	-	-
		ResNet-101 (from scratch baseline)	-	-	90.0	90.0	-	-	-
		PCANet (Chan et al., 2015)	-	-	77.1	77.1	-	-	-
		ViT-huge (Dosovitskiy et al., 2021)†	-	-	99.5	99.5	-	-	-
Video Classification	TinyVIRAT	Ours (PSViT adapted)	50.1	0.973	74.1 (30.4*)	60.4	-	-	-
		ResNet50†	-	-	29.1*	-	-	-	-
		WideResNet†	-	-	32.6*	-	-	-	-
Audio Classification	AudioMNIST	Ours (PSViT adapted)	50.9	0.989	96.9	97.1	-	-	-
		AlexNet (Becker et al., 2024)	-	-	-	95.8	-	-	-
Video QA	CLEVRER	Ours (PSViT adapted)	25.7	0.798	52.4	52.5	-	-	-
		LSTM (Yi et al., 2019)†	-	-	-	34.7	-	-	-
		+ CNN (Yi et al., 2019)†	-	-	-	51.8	-	-	-
	TGIF-QA	Ours (PSViT adapted)	42.0	0.985	53.2	52.3	-	-	-
		LSTM + Attn (Jang et al., 2019)	-	-	-	51.9	-	-	-
Object Tracking	LaSOT	Ours (PSViT adapted)	35.7	0.987	-	-	0.63	-	-
Video Colorization	TinyVIRAT	Ours (PSViT adapted)	42.4	0.997	-	-	-	0.02	73.1

Table 6.2: Performance metrics across various tasks using our proposed next-frame prediction framework with an adapted PSViT model. Metrics include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), F1-score and Accuracy (F1, Acc), Intersection over Union (IoU), Colour Distribution Consistency index (CDC), and Colorfulness, each tailored to evaluate specific task outputs. *Macro F1-score used for TinyVIRAT to allow direct comparison with baseline models. †indicates models with pre-training on other datasets or different architectural components not used in our "from scratch" end-to-end setup.

Figure 6.8: Sample outputs from our next frame prediction model across multiple modalities. The model receives frames to the left of the red line as input and everything right of the red line has been generated by the model. Tasks in order (top to bottom): Audio Classification, Image Classification, Text Classification, Video Colorization, Video Object Tracking, Video QA, Video Classification (bottom 3 rows)

On the CIFAR-10 image classification task, our model attains an accuracy of 89.1. While this score is lower than fine-tuned ViT models leveraging large-scale pre-trained weights, which can reach accuracies near 99.5 (Dosovitskiy et al., 2021), our model, trained from scratch on CIFAR-10 alone, performs significantly better than early deep learning models like PCANet (77.1 accuracy (Chan et al., 2015)) and is competitive with ResNet architectures of moderate size when also trained from scratch on CIFAR-10 (e.g., ResNet-101 from scratch is around 90-92%). This performance suggests that our reformulation approach enables the model to learn effective image classification capabilities without specific image pre-training.

In multi-label video classification on the TinyVIRAT dataset, our approach successfully outputs multiple labels per instance, achieving an F1-score of 74.1 (unnormalised) and 30.4 (macro F1, for comparison with baselines). This surpasses the ResNet50 baseline (29.1 macro F1) but does not reach the level of the WideResNet model (32.6 macro F1) reported by Demir et al. (2021), both of which are strong vision-specific architectures.

On the AudioMNIST audio classification task, the model achieves an accuracy of 97.1, surpassing the AlexNet baseline score of 95.82 from the original dataset paper (Becker et al., 2024). This notable improvement demonstrates the model's efficacy in audio-based classification tasks when audio is represented visually as a spectrogram. Analysis of misclassifications reveals that the most frequent error involves confusion between the spoken digits "four" and "five", suggesting areas for potential refinement in distinguishing phonetically similar sounds.

For the CLEVRER VQA (Visual Question Answering) dataset, our model achieves an accuracy of 52.5 on descriptive questions. This performance surpasses both the LSTM (34.7) and LSTM+CNN (51.8) baselines reported by Yi et al. (2019), illustrating our model's ability to understand relationships between visual and textual information within the unified framework. It is important to note that due to hardware limitations and the long question lengths, we subsampled the video to only four frames and reduced the resolution to 64×64, significantly limiting the accuracy achievable by our approach compared to models trained on full data. On TGIF-QA,

we achieve 52.3% accuracy, comparable to the 51.9% from an LSTM+Attention baseline (Jang et al., 2019).

In the LaSOT object tracking task, the model reaches an intersection over union (IoU) of 0.63, demonstrating a consistent ability to track objects throughout video sequences. Analysis of tracking outputs reveals that the model effectively follows the object of interest across frames with accurately drawn bounding boxes. However, tracking accuracy diminishes slightly toward the end of longer sequences, with the borders of the overlaid bounding box eroding, likely due to autoregressive pixel-level error propagation.

On the video colorisation task with TinyVIRAT, our model shows improved colour diversity in the generated outputs compared to the ground truth (73.1 colour diversity for our model versus 70.6 in the original dataset). However, it exhibits less consistency in colour application across the video sequence (0.0169 CDC for our model versus 0.00522 in the original dataset, where lower CDC is better). This trade-off suggests that while our approach introduces a more vibrant colour palette, further refinement could improve temporal coherence.

6.5.1 Attention Maps

To examine what is being learned by our unified model, we visualise internal patch-wise attention scores both spatially (within a frame) and temporally (between frames) in Figure 6.9.

In the TinyVIRAT video classification task, the model pays most attention to the object/person performing the action, as well as frames and pairs of frames which indicate the class. In the first example (top row), spatial attention focuses on image patch locations containing the edges of the car which suggest relative movement, and temporal attention is highest for the final two frames in which the vehicle moves position, indicating the label “vehicle moving” should be output.

For colorisation, the greatest attention is paid to the first frames of the input sequence, which we suggest allows the model to produce consistent colouring across

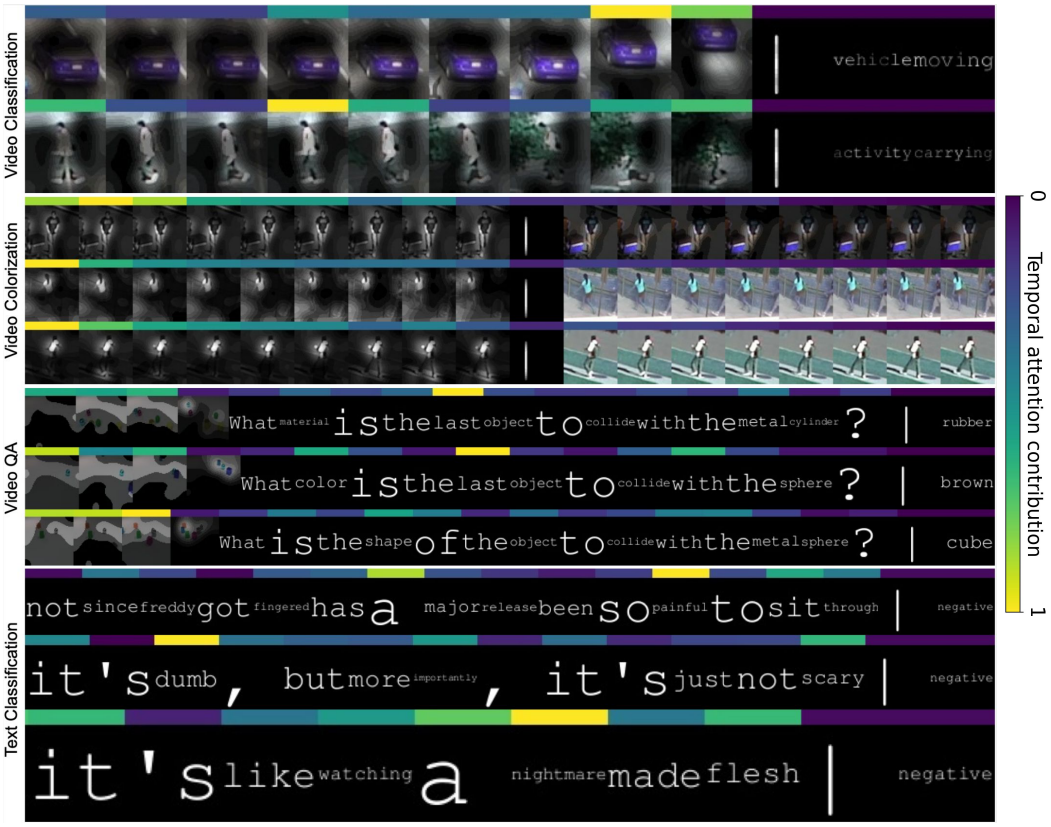


Figure 6.9: Attention visualisations showing where the model attends both spatially (indicated by light/dark areas overlaid on the frame) and temporally (indicated by a colour scale above the sequence). Spatial attention scores are calculated by taking the patch-wise contribution per frame, averaged over all layers. Temporal attention scores are calculated by taking the temporal attention weights for each input frame when generating the target frames, averaged over all global space-time attention layers. These maps offer insights into how the model integrates information across modalities and time to perform hierarchically complex tasks.

the whole sequence (more attention is also paid to the first frame of the output sequence, presumably to maintain colour consistency). Spatially, the model clearly attends to key moving objects and areas of distinct colour.

In the CLEVRER examples, we see how the model is able to focus on question frames which contain key words needed to produce the output, such as “colour” and “metal”. Within the moving object scene frames, we see how the model spatially attends to objects and their trajectory paths, correctly identifying objects such as “the metal cylinder” and “the sphere”.

Lastly, for sentiment text classification, we observe a similar pattern to attention maps in LLMs, where the model pays more attention to words with a strong

emotive sentiment. In our model, we see frames containing the words 'nightmare', 'painful', and 'dumb' exhibiting high temporal attention weightings, as they are strong indicators of the overall sentiment of the review. These patterns suggest an emergent division of labour where the model first processes unimodal features before integrating them into higher-level, cross-modal, and task-relevant hierarchical representations.

6.5.2 Limitations and Future work

The aim of this work is not necessarily to surpass state-of-the-art performance on any of the individual tasks, but rather to demonstrate that unifying these diverse modalities and tasks via a common next-frame prediction framework is feasible, achieving competitive performance relative to task-specific baselines trained under similar from-scratch conditions. Future work should aim to scale this approach, improving performance and tackling more complex tasks. We suggest that this can be done by mirroring the improvements gained from pre-training in NLP and computer vision, by first training the model on unstructured, self-supervised next-frame prediction tasks across a vast corpus of mixed-modality data. Tasks such as language modelling (rendered as text frames) and general video/image completion could be naturally reformulated as next-frame-prediction and included in such a pre-training paradigm. This is especially important as a large portion of the current training time on tasks such as classification and question answering is dedicated to learning to output words as images rather than solely focusing on the core reasoning aspect of the task. Beginning with a checkpoint already pre-trained on generating diverse visual frame sequences (including rendered text and spectrograms) would likely mitigate this and allow the model to better focus on learning the higher-level hierarchical relationships required for complex multimodal tasks.

OCR can fail when text is complex or when generation introduces artefacts such as blur, or compression. To evaluate the idea rather than OCR behaviour, we use simple, high contrast, upright sans serif text at sufficient size and we verified correct recognition across the full dataset vocabulary. In our pipeline the cost grows

with the number of frames and with the number of spatial tokens per frame, so temporal and spatial dimensions compound, whereas in text models length is one dimensional in tokens. Because outputs are images, inference includes rendering and OCR, which adds latency and makes exact string evaluation less direct than token decoding. Increasing the output resolution improves OCR legibility but also raises memory and compute costs roughly quadratically with image size. It can slow both training and inference, and may require retuning the patch size.

6.6 Conclusion

In an era increasingly dominated by large foundation models, the trend is towards more unified, end-to-end systems capable of handling diverse tasks and modalities without task-specific components. In this chapter, we introduced a novel multimodal learning framework that reformulates various tasks from different modalities into a unified next-frame prediction paradigm, processed by a single Transformer architecture adapted from the PSViT model (Chapter 5). This approach addresses critical limitations in current multimodal model designs, which often require modality-specific encoders and are limited in scalability and flexibility when adapting to new tasks. By unifying diverse multimodal tasks under a single framework, our model can handle text, image, audio, and video inputs without modality-specific input encoders, significantly simplifying the design and training process for learning shared hierarchical representations. We have shown it is possible to train such a model to solve these tasks from scratch, with many of them achieving performance comparable to, or exceeding, baseline models trained without large-scale external pre-training data.

Ultimately, our work aims to lay the groundwork for building more generalised and efficient multimodal foundation models, contributing to research question 3 of this thesis. It points towards systems that can process and understand information from diverse modalities through a common predictive lens, potentially learning shared hierarchical abstractions in a unified and scalable way.

6.7 Epilogue

This chapter demonstrated that a single Transformer architecture, employing the principles of the PSViT model from Chapter 5 and trained only with a next-frame prediction objective, can successfully process and reason over tasks spanning text, images, video, and audio without recourse to modality-specific encoders. Instead of a clear layer-wise separation of features, our analysis of the model's internal mechanisms (particularly its cross-modal attention) suggests a different kind of emergent structure. The model learns to create a shared representational space where it can dynamically align and compose information from different modalities. For instance, to solve a visual question answering task, the model was observed attending simultaneously to key instruction words rendered as text frames and to the corresponding, semantically relevant objects in the video frames. This ability to form on-the-fly, task-relevant connections between disparate data types is a powerful form of relational reasoning, indicating the development of an internal strategy for building a cohesive, cross-modal understanding via end-to-end "from scratch" training. This emergent strategy of cross-modal alignment, while different in form from the layer-wise hierarchies observed in Chapters 3 and 5, reinforces the central theme of this thesis. It demonstrates that when presented with complex, structured data, Transformers find ways to build hierarchical understandings. In this case, composing meaning from different modal parts to form a cohesive whole. The evidence accumulated throughout this thesis suggests that *the learning and utilisation of hierarchical representations are not artefacts of any single domain but constitute a recurring and fundamental characteristic of how Transformer models process complex, structured information.* This demonstration of a unified framework capable of inducing cross-modal hierarchical understanding directly addresses research question 3.

The final chapter, Chapter 7, will revisit the findings from all our empirical investigations, and discuss their implications for the design, scaling, interpretation, and safe deployment of future hierarchy-aware foundation models.

Bibliography

Agarwal, L. and Verma, B. (2023). From methods to datasets: A survey on image-caption generators. *Multimedia Tools and Applications*, 83:28077–28123.

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. (2022). data2vec: A general framework for self-supervised learning in speech, vision and language. *ArXiv*, abs/2202.03555.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.

Becker, S., Vielhaben, J., Ackermann, M., Müller, K.-R., Lapuschkin, S., and Samek, W. (2024). Audiomnist: Exploring explainable artificial intelligence for audio analysis on a simple benchmark. *Journal of the Franklin Institute*, 361(1):418–428.

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., and Ma, Y. (2015). Pcanet: A simple deep learning baseline for image classification? *IEEE transactions on image processing*, 24(12):5017–5032.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. (2020). Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR.

Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. (2019). Uniter: Learning universal image-text representations. *ArXiv*, abs/1909.11740.

Demir, U., Rawat, Y. S., and Shah, M. (2021). Tinyvirat: Low-resolution video action recognition. In *2020 25th international conference on pattern recognition (ICPR)*, pages 7387–7394. IEEE.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and

Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., and Ling, H. (2019). Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5374–5383.

Gao, T., Wang, Z., Bhaskar, A., and Chen, D. (2024). Improving language understanding from screenshots. *arXiv preprint arXiv:2402.14073*.

Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Li, F.-F., Essa, I., Jiang, L., and Lezama, J. (2023). Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*.

Hu, R. and Singh, A. (2021). Unit: Multimodal multitask learning with a unified transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1419–1429.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., et al. (2021). Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*.

Jang, Y., Song, Y., Kim, C. D., Yu, Y., Kim, Y., and Kim, G. (2019). Video Question Answering with Spatio-Temporal Reasoning. *IJCV*.

Keskar, N. S., McCann, B., Xiong, C., and Socher, R. (2019). Unifying question answering, text classification, and regression via span extraction. *arXiv preprint arXiv:1904.09286*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. (2021). Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*, volume 34, pages 9694–9705.

Liu, Y., Zhao, H., Chan, K. C., Wang, X., Loy, C. C., Qiao, Y., and Dong, C. (2024). Temporally consistent video colorization with deep feature propagation and self-regularization learning. *Computational Visual Media*, 10(2):375–395.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lu, S., Liu, M., Yin, L., Yin, Z., Liu, X., and Zheng, W. (2023). The multi-modal fusion in visual question answering: a review of attention mechanisms. *PeerJ Computer Science*, 9:e1400.

McCann, B., Keskar, N. S., Xiong, C., and Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

OpenAI (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. (2014). Video (language) modeling: a baseline for generative models of natural videos. *ArXiv*, abs/1412.6604.

Rust, P., Lotz, J. F., Bugliarello, E., Salesky, E., de Lhoneux, M., and Elliott, D. (2022). Language modelling with pixels. *arXiv preprint arXiv:2207.06991*.

seok Seo, M., Lee, H., Kim, D.-Y., and Seo, J. (2023). Implicit stacked autoregressive model for video prediction. *ArXiv*, abs/2303.07849.

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. (2021). FLAVA: A foundational language and vision alignment model. *CoRR*, abs/2112.04482.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Tai, Y., Liao, X., Suglia, A., and Vergari, A. (2024). Pixar: Auto-regressive language modeling in pixel space. *arXiv preprint arXiv:2401.03321*.

van Amersfoort, J. R., Kannan, A., Ranzato, M., Szlam, A., Tran, D., and Chintala, S. (2017). Transformation-based models of video sequences. *ArXiv*, abs/1701.08435.

Wang, Y., Wang, L., Zhou, Q., Wang, Z., Li, H., Hua, G., and Tang, W. (2024). Multimodal llm enhanced cross-lingual cross-modal retrieval. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8296–8305.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., and Van Den Hengel, A. (2017). Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40.

Xiao, C., Huang, Z., Chen, D., Hudson, G. T., Li, Y., Duan, H., Lin, C., Fu, J., Han, J., and Moubayed, N. A. (2024). Pixel sentence representation learning. *arXiv preprint arXiv:2402.08183*.

Yan, W., Zhang, Y., Abbeel, P., and Srinivas, A. (2021). Videogpt: Video generation using vq-vae and transformers.

Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., and Tenenbaum, J. B. (2019). Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*.

Yilmaz, M. A. and Tekalp, A. M. (2021). Dfpn: Deformable frame prediction network. *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1944–1948.

Zhong, Q., Ding, L., Peng, K., Liu, J., Du, B., Shen, L., Zhan, Y., and Tao, D. (2023). Bag of tricks for effective language model pretraining and downstream adaptation: A case study on glue. *arXiv preprint arXiv:2302.09268*.

Zou, Y. and Xie, Q. (2020). A survey on vqa: Datasets and approaches. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, pages 289–297. IEEE.

Chapter 7

Discussion & Concluding Remarks

7.1 Contributions

This thesis set out *to discover how hierarchical information is learned, encoded, and represented inside Transformer networks*, driven by three central research questions: (1) How and where do foundation models internalise hierarchical structures? (2) What is the relationship between this internalisation of hierarchy and behaviours like generalisation versus memorisation? (3) How can insights into hierarchical reasoning guide the development of more unified, robust, and interpretable multimodal systems? Through a series of empirical investigations spanning language, video, and multimodal tasks, we have uncovered regularities and proposed practical tools for studying and shaping how Transformers engage with hierarchical structure. The main innovations are summarised by chapter below:

Chapter 3. ***Hierarchical Information in Contextual Representations*** This chapter aimed to address our first research question:

- We introduced **ancestor-level probes** that require a single token embedding to predict labels for increasingly distant syntactic or sentiment parents, re-

vealing how different levels of linguistic hierarchy emerge layer-by-layer within models.

- We showed across four Transformer families that *bidirectional* models tend to concentrate this hierarchical information near the top of the network, whereas *autoregressive and permutation-masked* models distribute it more evenly, an architectural bias that persists even after fine-tuning.

Chapter 4. *Hierarchy in Language Model Memorisation* Focusing on our second research question, this chapter investigated the interplay between useful hierarchical learning and detrimental memorisation:

- We demonstrated that verbatim leakage during fine-tuning often spikes *before* validation perplexity or task accuracy plateaus, and proposed an **n-gram partial-memorisation score** that reliably predicts which samples are at imminent risk of being leaked.
- We leveraged this signal to develop two practical defences: an early-stopping rule that cuts leakage by approximately 50% at minimal performance cost, and an n-gram-aware loss regulariser that can reduce leakage by a further 40%, promoting the learning of generalisable structures over verbatim memorised recall.

Chapter 5. *Spatiotemporal Reasoning in Video* Here, we extended our inquiry into hierarchy internalisation to the visual domain and laid groundwork relevant to multimodal systems:

- We introduced PSViT, a Transformer whose architecture incorporates several **hierarchically-aware priors and optimisations**, including a U-Net style structure and a tailored spatiotemporal attention layout, which were selected through extensive ablation studies to best capture physical dynamics. This model outperforms latent-space baselines on physical-simulation benchmarks, extending physically accurate prediction horizons by up to 50%.

- Using layer-wise object-tracking probes and parameter regression, we showed that abstract physical parameters (e.g., gravity, mass), which govern the hierarchical evolution of scenes, are most linearly recoverable from mid-layers, echoing the mid-layer abstraction peak observed for linguistic hierarchy in Chapter 3.

Chapter 6. ***Multimodal Multi-Task Hierarchical Reasoning*** This chapter directly addressed our third research question, demonstrating that principles of hierarchical reasoning can extend to, and help unify, multimodal understanding:

- We established the viability of reformulating diverse tasks (spanning text, image, audio, and video) into a single **next-frame prediction super-task** on a common visual canvas. This approach enabled a single adapted PSViT model, operating without modality-specific encoders, to achieve competitive performance across seven benchmarks, indicating the development of effective shared internal representations necessary for such cross-modal proficiency.
- Our analysis of the model's attention mechanisms revealed an emergent strategy for compositional reasoning. We observed attention patterns similar to those in our unimodal models, where the system learned to dynamically align key instruction words in text frames with corresponding, semantically-relevant objects in video frames. This suggests our task reformulation was effective, and demonstrates that the architectural principles of PSViT, designed to capture hierarchy, can successfully generalise to guide the model in building a cohesive, cross-modal understanding from the ground up.

Together, these studies shine a light on how Transformers engage with hierarchical structure. Whether processing language (Chapter 3) or visual dynamics (Chapter 5), we observe that causal Transformers first gather local features; then organise them into more abstract, domain-specific hierarchical representations in their middle layers; and finally compress these into task-relevant decisions or generations at their output layers. This emergent hierarchical processing is detectable, and can be influ-

enced by architectural choices and training regimes. The work in Chapter 6 revealed this same underlying principle in a new form. Its emergent strategy of cross-modal alignment, while different from the layer-wise hierarchies observed previously, reinforces the central theme of this thesis. It demonstrates that when presented with complex, structured data, Transformers consistently find ways to build hierarchical understandings: in this case, by composing meaning from different modal parts to form a cohesive whole. This emergent processing, whether layered or compositional, appears to be a fundamental characteristic of how these models achieve generalisation across diverse and complex data.

7.2 Limitations and Future Work

While this thesis offers several insights, each empirical chapter also presents limitations that suggest avenues for future research.

Chapter 3. The probing tasks relied primarily on the Stanford Sentiment Treebank and English Penn Treebank-style syntax, which limits the linguistic diversity of our findings. Future work should extend ancestor probing to morphologically rich languages, different syntactic formalisms (e.g., dependency grammar, mathematical problems, coding), and broader discourse structures.

Chapter 4. Leakage was measured under greedy decoding using datasets of up to $5,000$ samples. Different decoding strategies (e.g., beam search, temperature sampling) or industrial-scale fine-tuning datasets may exhibit different memorisation dynamics. Scaling the n-gram-aware loss regulariser to operate efficiently on hundreds of billions of tokens, perhaps via low-rank adaptation methods or more selective n-gram tracking, remains an open engineering challenge.

Chapter 5. PSViT was evaluated at 128×128 resolution and predominantly on deterministic physical simulations. Larger-scale models, more complex stochastic

environments, and real-world video (e.g., depicting human actions with unconstrained camera motion) will be necessary to test the generalisability of the physical abstraction capabilities we observed. Integrating a differentiable renderer could also allow for translating mid-layer object and dynamics codes into symbolic physics engines for more direct interpretability.

Chapter 6. The multimodal model was trained separately for each task. True *joint* multitask training on a diverse corpus of reformulated multimodal data might better foster the emergence of shared hierarchical representations and could potentially close the remaining performance gap with specialist systems. Furthermore, rendering long text sequences or high-fidelity audio into 64×64 frames is a somewhat crude representation; exploring hierarchical or vector-graphic encodings within the visual canvas could allow for richer information transfer per token or sound segment.

Across chapters. Our analyses predominantly used linear or shallow non-linear probes. Understanding how more complex mechanisms interact with learned hierarchies, and, more fundamentally, whether the identified hierarchical representations are causally responsible for downstream task success or specific model behaviours, calls for more sophisticated interventions and mechanistic analysis.

7.3 Epilogue

Over the lifecycle of this thesis, the community has gone from marvelling at the contextual capabilities of BERT; to an era dominated by commonplace usage of generative models like GPT-4. Throughout this time, the impact of model scale has been unavoidable: "more data, more parameters, better scores". The work presented in this thesis does not dispute the empirical success of scaling, but it argues that *size alone is not the whole story.* Across four empirical chapters and three distinct modalities, we have observed a different, fundamental force at work: the emergent organisation of raw input streams into nested, reusable hierarchical

abstractions, and we have learned that attending to this force can grant practical improvements for interpretability, safety, and performance.

Hierarchy is structural, not merely statistical. Our layer-wise probes in language (Chapter 3) and PDE-driven video (Chapter 5) consistently uncovered a multi-stage pattern of information processing. In these unimodal contexts, local features are captured at the bottom layers, more structured and abstract concepts emerge in the middle layers, and task-specific decisions are finalised at the top. While our multimodal investigation in Chapter 6 did not focus on layer-wise analysis, it revealed a different but equally important form of structural organisation: a **compositional hierarchy**, where the model dynamically aligned and composed features across modalities to build a cohesive understanding.

These findings reflect a deep inductive bias of the Transformer architecture when processing structured data, rather than being a quirk of any single training corpus. This directly informs our first research question on how and where hierarchy is internalised. Consequently, designers of future foundation models may wish to consider the middle layers identified in our unimodal studies as crucial *interfaces*: deep enough to have distilled useful abstractions, yet potentially accessible enough for diagnosis, steering, or extracting interpretable representations.

Remembering with care requires understanding structure. The same powerful capacity that builds useful hierarchies can also lead to the memorisation of personal, identifiable, or sensitive text, posing significant risks. Chapter 4 approached memorisation not as an inevitable side effect but as something which could be mitigated with simple optimisation, where partial n-gram echoes often precede verbatim reproduction. Detecting this progression enabled the development of simple yet effective defences: early stopping based on n-gram overlap and a lightweight n-gram-aware loss regulariser, that substantially reduce leakage while largely preserving task accuracy. This helps address our second research question, demonstrating that *interpretability* (of internal structure related to n-grams) *and safety* (mitigating memorisation) *are not necessarily opposing goals*; understanding

internal learning dynamics can directly inform concrete mitigations, promoting the generalisation of useful structures over memorised learning.

Prediction as a common language for multimodal hierarchy. Re-encoding text, audio, and imagery as inputs to a single next-frame prediction task initially seemed like an ambitious experiment. However, as shown in Chapter 6, it proved to be a viable approach. Without modality-specific encoders, our adapted PSViT model handled seven diverse benchmarks. More importantly, analyses of its attention mechanisms suggested the formation of cross-modal alignments that mirrored the hierarchical processing seen within unimodal language and video contexts. This way of learning, which emerged naturally from the unified predictive task, mirrors the hierarchical patterns we observed in our earlier unimodal studies of language (Chapter 3) and video dynamics (Chapter 5). This consistency across different contexts strongly supports our third research question, highlighting that the ability to learn and use hierarchical representations is a fundamental principle not limited to any single type of data. By requiring the model to find common rules to predict the next visual frame for all these varied data streams, the next-frame prediction approach appears to actively encourage the discovery of shared, deeply structured ways of understanding. This insight is a step towards developing AI systems that are truly *general* and unified in their capabilities.

Open horizons. Several lines of inquiry naturally extend from this work:

- **Scaling interpretability of hierarchy.** The probes used here were primarily linear. Richer causal intervention techniques could reveal whether the observed hierarchical representations are not just correlated with, but are prerequisites for, robust generalisation and specific emergent capabilities.
- **Energy and data efficiency through hierarchy.** If mid-layer hierarchical abstractions are indeed stable and transferable across model sizes or tasks, could smaller, specialised models inherit them through knowledge distilla-

tion or parameter-efficient adaptation techniques, leading to more efficient systems?

- **Ethics at the abstraction boundary.** Controlling what models remember and generalise becomes even more complex when inputs mix private text, personal images, and other sensitive data streams. Developing regularisers or training methodologies that operate on *conceptual or hierarchical abstractions*, rather than just surface-level tokens or pixels, to ensure privacy and fairness is a critical open research agenda.

Transformers will certainly continue to grow larger; supporting increasingly complex data structures and modalities. Whatever form the next generation of foundation models takes, the evidence gathered in this thesis argues that their success, safety, and interpretability will increasingly depend on *how well we understand, measure, and respect the hierarchical structures those networks inevitably build.* If we can achieve this, and make the analysis of structural understanding and hierarchical reasoning as central to our engineering practices as training loss curves and benchmark evaluation scores, then we can move a step closer towards developing AI systems that reason, remember, and create with the layered discipline, robustness, and generality characteristic of natural intelligence itself.

Dean Lewis Slack

Department of Computer Science

Durham University, United Kingdom

May 2025