Content Generation for Open-World Physics-Based Environments

A thesis submitted for the degree of
Doctor of Philosophy

Chathura Gamage

School of Computing
College of Engineering, Computing and Cybernetics (CECC)
The Australian National University

April 2024

© Chathura Gamage 2024

Except where otherwise indicated, this thesis is my own original work.

Chathura Gamage
8 April 2024

In memory of my loving father.

Acknowledgements

I would like to express my heartfelt gratitude to my primary supervisor, Prof. Jochen Renz, for his consistent guidance, support, and mentorship throughout the journey of pursuing my PhD. His expertise, advice, and motivational encouragement have been instrumental in shaping my research path. I am also thankful for his approachable and friendly nature, which made our interactions highly productive.

I extend my appreciation to my co-supervisor, Dr. Matthew Stephenson, whose valuable insights and advice have been a guiding light during my research. His readiness to provide assistance whenever I sought it has been invaluable. Additionally, I would like to acknowledge my co-supervisor, Assoc. Prof. Patrik Haslum, for his wisdom and thoughtful suggestions that have enriched my work.

I am deeply grateful to the Australian National University (ANU) for awarding me a PhD scholarship and to the Defense Advanced Research Projects Agency (DARPA) for their financial support, which has allowed me to focus on my research without financial constraints. The facilities and resources provided by the ANU School of Computing have also been indispensable in facilitating my research endeavours.

My gratitude extends to my alma mater, Mahinda College, Galle, and the University of Peradeniya, Sri Lanka. Moreover, I am indebted to the people of Sri Lanka for making free education a reality and ensuring equal access to quality education for all.

A heartfelt appreciation goes out to my dear friends and colleagues, Cheng, Katya, Vimukthini, Peng, and Alex, who have been instrumental in numerous meetings and late-night discussions, offering their guidance, advice, and companionship.

I am deeply thankful to my cherished wife, Chamodi, for her constant presence in my life, her boundless love, commitment, and unwavering emotional support throughout all these years. I am also grateful for her delicious food and the joyful trips we have shared.

Last but certainly not least, my profound gratitude goes to my beloved mother, Chandra, whose enduring love and faith in me have been my pillars of strength. I also extend my thanks to my siblings, Ruvani and Supun, my extended family and friends, and everyone who has stood by me throughout this journey.

Abstract

Researchers are driven by the aspiration to develop AI systems that can seamlessly operate in the real physical world, aiming to enhance human life through automation and assistance. However, such AI systems encounter an array of challenges. Foremost among these hurdles is the essential need for physical reasoning capabilities, enabling the AI to make informed decisions regarding the behaviour of objects under the influence of physics. Furthermore, the open nature of the real world means that novel and unforeseen situations frequently arise, necessitating AI to possess adaptive capabilities in order to thrive in such dynamic environments. In this research, we propose various techniques to design and generate physics-based content, which serves as an experimental and evaluative platform to facilitate addressing the above challenges faced by those AI systems.

Frequently, simulation environments are used by researchers to solve complex real-world problems as they offer controllable environments for experiments. Our study centres on Angry Birds, a physics-simulating puzzle game, to present various approaches for designing and generating tasks to facilitate the development of the aforementioned AI systems. However, generating physics-based content tailored for AI evaluation poses challenges, demanding physical reasoning and intricate design to meet evaluation requirements. Regrettably, existing physics-based content generation methods lack the specificity of producing content for evaluating AI systems in open-world physical environments.

In this thesis, firstly, we propose techniques to design and generate content for evaluating the physical reasoning capabilities of AI systems. These techniques involve generating tasks capable of deceiving AI systems in physics-based environments, crafting tailored tasks to measure the physical reasoning intelligence of agents in comparison with human intelligence, and developing methods to generate tasks based on the causal physical interactions between objects in the environment. Secondly, we discuss methods to design and generate physics-based tasks that incorporate novel situations. These works include creating testbeds with tasks to measure AI systems' performance in open-world physics-based environments and generating tasks with detectable and adaptable novelties for AI agents, allowing confident evaluation of their capabilities. Through the application of our research findings, we aim to foster the development of AI systems that can effectively and efficiently thrive in open-world physics-based environments.

List of Publications

The following is the list of publications to which I have contributed throughout my PhD research.

Publications that are included in this thesis.

- C. Gamage, V. Pinto, J. Renz, and M. Stephenson, Deceptive Level Generation for Angry Birds, 2021 IEEE Conference on Games, COG 2021. Published. (Chapter 3)

  Own Contributions: I contributed to all aspects of this research with the support mentioned below from my co-authors.

  Contributions by Co-Authors: J. Renz and M. Stephenson were involved in conceptualizing the study. M. Stephenson offered consistent feedback to enhance the research. Both M. Stephenson and V. Pinto contributed to improving the paper's clarity. Additionally, J. Renz offered supervision throughout the study.

- C. Gamage*, C. Xue*, V. Pinto*, E. Nikonova, P. Zhang, and J. Renz, Phy-Q as a Measure for Physical Reasoning Intelligence, Nature Machine Intelligence, Nat Mach Intell 5, 2023. *Equal Contributions. Published. (Chapter 4)

  Own Contributions: I played a significant role in conceiving, conceptualizing, and formulating this study. I was responsible for designing and developing task templates for the testbed and developing the automated task generator. I contributed to conducting experiments involving heuristic agents and the evaluation of the AIBIRDS competition. Additionally, I also contributed to the writing of the paper. Furthermore, I was involved in organizing and conducting the human player evaluation. In addition, I played a part in the development of the testbed software.

  Contributions by Co-Authors: The study's conceptualization and formalization also involved contributions from C. Xue, V. Pinto, and J. Renz. In terms of the writing process, both C. Xue and V. Pinto made significant contributions. C. Xue was responsible for implementing learning agents, while V. Pinto played a role in designing task templates and analyzing the data. E. Nikonova authored the section related to learning agents. The implementation of the testbed software involved

C. Xue and P. Zhang. Throughout the study, J. Renz provided valuable feedback and supervision.

- C. Gamage, V. Pinto, M. Stephenson, and J. Renz, Physics-based Task Generation through Causal Sequence of Physical Interactions, The 19th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2023. Published. (Chapter 5)

  Own Contributions: I contributed to all aspects of this research with the support mentioned below from my co-authors.

  Contributions by Co-Authors: The conception and initial discussions about the study were contributed by J. Renz and M. Stephenson. Additionally, M. Stephenson and V. Pinto played instrumental roles in enhancing the clarity of the paper. The study also benefited from the supervisory guidance provided by J. Renz.

- C. Gamage, V. Pinto, C. Xue, M. Stephenson, P. Zhang, and J. Renz, Novelty Generation Framework for AI Agents in Angry Birds Style Physics Games, 2021 IEEE Conference on Games, COG 2021. Published. (Chapter 6)

  Own Contributions: I contributed to all aspects of this research with the support mentioned below from my co-authors.

  Contributions by Co-Authors: V. Pinto played a significant role in formulating the tests within the framework. Both V. Pinto and C. Xue were instrumental in the paper-writing process, and M. Stephenson played a key role in enhancing the paper's clarity. P. Zhang contributed to implementing the framework. J. Renz consistently provided supervision throughout the study.

- C. Gamage*, V. Pinto*, C. Xue*, P. Zhang, E. Nikonova, M. Stephenson, and J. Renz, NovPhy: A Testbed for Physical Reasoning in Open-World Environments, Artificial Intelligence Journal, AIJ. *Equal Contributions. Under Review. (Chapter 7)

  Own Contributions: I contributed to conceiving this study, as well as to conceptualizing and formulating the testbed. I took responsibility for designing task templates for the testbed, as well as developing the automatic task generator. I was also involved in conducting experiments on non-learning agents. I actively participated in writing the paper. Additionally, I was involved in designing the evaluation process and played a role in conducting the human player evaluation. Furthermore, I contributed to the development of the testbed software.

  Contributions by Co-Authors: V. Pinto, C. Xue, and J. Renz played pivotal roles in conceptualizing and formalizing the study. V. Pinto made significant contributions to the writing process, designing task templates, and data analysis. C. Xue took charge of implementing learning agents and authored the corresponding section in the paper. E. Nikonova contributed to discussions involving learning agents. P. Zhang was involved in implementing the testbed software. M. Stephenson provided

valuable insights to enhance the paper's clarity. Throughout the study, J. Renz provided essential feedback and consistent supervision.

- C. Gamage, M. Stephenson, and J. Renz, Physics-Based Novel Task Generation through Disrupting and Constructing Causal Interactions, IEEE Transactions on Games, ToG. Published. (Chapter 8)

  Own Contributions: I contributed to all aspects of this research with the support mentioned below from my co-authors.

  Contributions by Co-Authors: J. Renz offered valuable supervision and played a key role in shaping the task design theory. M. Stephenson played a crucial role in enhancing the evaluation measures and overall clarity of the paper.

Publications that are not included in this thesis.

- V. Pinto, C. Gamage, P. Zhang, M. Stephenson, and J. Renz, Towards a Systematic Evaluation of Novelty Detection in Open-World AI Systems, Artificial Intelligence Journal, AIJ. Under Review.
- V. Pinto, C. Gamage, M. Stephenson, and J. Renz, The Difficulty of Novelty Detection and Adaptation in Physical Environments, The Thirty-Sixth Australasian Joint Conference on Artificial Intelligence, AJCAI 2023. Published.
- V. Pinto, C. Xue, C. Gamage, M. Stephenson, and J. Renz, The Difficulty of Novelty Detection in Open-World Physical Domains: An Application to Angry Birds, The Twenty-Sixth European Conference on Artificial Intelligence - Qualitative Reasoning, ECAI-QR 2023. Published.
- C. Xue, V. Pinto, P. Zhang, C. Gamage, E. Nikonova, and J. Renz, Science birds novelty: An open-world learning test-bed for physics domains, AAAI Designing Artificial Intelligence for Open Worlds, 2022. Published.
- E. Nikonova, C. Xue, V. Pinto, C. Gamage, P. Zhang, and J. Renz, Measuring Difficulty of Novelty Reaction, AAAI Designing Artificial Intelligence for Open Worlds, 2022. Published.

Table of Contents

1 Introduction **1**
1.1 Overview . . . 1
1.2 Motivation . . . 3
1.3 Research Problems . . . 4
1.4 Methodology . . . 4
1.5 Thesis Outline . . . 6

2 Background and Related Work **9**
2.1 Physical Reasoning . . . 9
2.1.1 Physical Reasoning Environments . . . 10
2.1.2 Angry Birds . . . 16
2.1.3 Physical Reasoning Agents . . . 18
2.2 Open-World Novelty . . . 20
2.2.1 Concept of Novelty . . . 22
2.2.2 Novelty Centric Environments . . . 23
2.3 Procedural Content Generation . . . 27
2.3.1 Procedurally Generated Content and Techniques . . . 28
2.3.2 PCG in Physics-Based Environments . . . 29

3 Deceptive Level Generation for Angry Birds **33**
3.1 Introduction . . . 33
3.2 Background and Related Work . . . 34
3.2.1 PCG in Angry Birds . . . 34
3.2.2 Deceptive Games . . . 35
3.3 Terminology . . . 36
3.4 Proposed Procedure . . . 37
3.4.1 Physical Entity Extraction Module . . . 37
3.4.2 Physical Entity Analyzer Module . . . 38
3.4.3 Level Template Matching Module . . . 39
3.4.4 Level Validation Module . . . 44
3.5 Results and Evaluations . . . 44
3.5.1 Stability . . . 46

3.5.2 Solvability 46
3.5.3 Deceptiveness 47
3.5.4 Comparison with Human-Created Levels 47
3.6 Conclusions and Future Work 48

4 Phy-Q: A Testbed for Physical Reasoning 51
4.1 Introduction 52
4.2 Background and Related Work 53
4.3 Phy-Q Testbed 56
4.3.1 Introduction to the Phy-Q Testbed 56
4.3.2 Physical Scenarios in Phy-Q Testbed 56
4.3.3 Task Templates and Task Generation 58
4.3.4 Proposed Evaluation Settings 59
4.4 Experiments 62
4.4.1 Baseline Agents 62
4.4.2 Experimental Setups 64
4.5 Results and Analysis 66
4.5.1 Human Performance 66
4.5.2 Generalization Performance and Phy-Q Score 67
4.5.3 AIBIRDS Competition Performance 69
4.6 Conclusion and Future Work 70

5 Physics-Based Task Generation through Causal Sequence of Physical Interactions 73
5.1 Introduction 73
5.2 Background and Related Work 75
5.3 Grammar and De ning Scenarios 76
5.3.1 Grammar for Angry Birds 76
5.3.2 De ning Scenarios 78
5.4 Task Generation Process 81
5.4.1 Generating the Layout Constraint Graph 81
5.4.2 Transforming Layout Constraints into QSRs 82
5.4.3 Generating a Spatial Con guration for QSRs 83
5.4.4 Satisfying Constraints through Simulation 85
5.4.5 Incorporating Distractions 86
5.4.6 Ensuring the Intended Solvability 86
5.5 Results and Evaluations 88
5.5.1 Physical Stability 88
5.5.2 Intended Solvability 89
5.5.3 Accidental Solvability 89
5.5.4 Generation Time 90
5.6 Conclusion and Future Work 90

6 Novelty Generation Framework for Physics-Based Environments 91
6.1 Introduction 91
6.2 Background and Related Work 93
6.2.1 Novelties Considered in this Study 93
6.2.2 Related Work 93
6.3 Overview of the Novelty Generation Framework 94
6.3.1 Novel Object Generator 94
6.3.2 Novelty Injector 95
6.4 Instantiating Novelty generation Framework to Angry Birds 95
6.4.1 Novel Object Generator 96
6.4.2 Novelty Injector 97
6.5 Angry Birds Novelty Objectives 98
6.5.1 Novelty Outcomes 99
6.5.2 Novelty Objective Example 101
6.6 Results and Evaluations 102
6.6.1 Examples of Generated Novelties 102
6.6.2 Evaluating the Generated Novelties 103
6.6.3 Experiments Using a Learning Agent 104
6.6.4 Discussion 106
6.7 Conclusion and Future Work 106

7 NovPhy: A Testbed for Physical Reasoning in Open-World Environments 107
7.1 Introduction 108
7.2 Background and Related Work 109
7.3 Designing Novel Tasks and Evaluations 110
7.3.1 Designing Novel Tasks 110
7.3.2 Designing Agent Performance Evaluations 111
7.4 NovPhy Testbed 112
7.4.1 Introduction to NovPhy 112
7.4.2 Physical Scenarios in NovPhy 112
7.4.3 Novelties used in NovPhy 113
7.4.4 Task Templates 114
7.4.5 Task Generation 116
7.4.6 Evaluation Protocol 116
7.4.7 Evaluation Measures 119
7.5 Experiments 121
7.5.1 Baseline Agents 121
7.5.2 Experimental Setups 122
7.6 Results and Analysis 123
7.6.1 Human Performance 123
7.6.2 Baseline Agent Performance 125
7.7 Conclusion and Future Work 129

8 Physics-Based Novel Task Generation through Manipulating Causal Interactions 131
8.1 Introduction . . . 131
8.2 Background and Related Work . . . 133
8.3 Novelty Scope . . . 134
8.4 Theory for Designing Tasks . . . 134
8.5 Grammar and De ning Scenarios . . . 137
8.5.1 Extending Grammar for Angry Birds . . . 137
8.5.2 De ning Scenarios . . . 138
8.6 Task Generation Process . . . 140
8.6.1 Qualitative Phase . . . 140
8.6.2 Simulation Phase . . . 144
8.7 Results and Evaluations . . . 146
8.7.1 Generation Time . . . 147
8.7.2 Physical Stability . . . 149
8.7.3 Intended Solvability . . . 150
8.7.4 Intended Unsolvability . . . 151
8.7.5 Solution Switch . . . 151
8.7.6 Accidental Solvability . . . 152
8.8 Conclusion and Future Work . . . 152

9 Conclusion and Future Work 155
9.1 Future Work . . . 158
9.2 Concluding Remarks . . . 160

A Appendix: Science Birds Objects 163

B Appendix: Phy-Q 165
B.1 Phy-Q Task Representation . . . 165
B.2 Phy-Q Tasks . . . 167
B.2.1 Task Templates in Phy-Q . . . 167
B.2.2 Di erences Between the Task Templates in Phy-Q . . . 173
B.2.3 Solution Descriptions for Example Task Templates in Phy-Q . . . 175
B.3 Local Genaralization Results . . . 176
B.4 Broad generalization Training and Testing Split . . . 177
B.5 Human Player Evaluation . . . 177
B.6 2021 AIBIRDS Competition Game Levels Used for the Evaluation . . . 179
B.7 Relation to learning in video games . . . 179
B.8 Computing Resources Used for the Experiments . . . 179

C Appendix: NovPhy 181
C.1 The Objects in NovPhy Tasks . . . 181
C.2 Tasks in NovPhy . . . 183
C.3 Designing Novel Tasks in NovPhy . . . 189

C.4 Novelty Detection Performance 190
C.5 Novelty Adaptation Performance 194
C.6 Human Detection vs Adaptation 200

Bibliography 203

List of Acronyms

AI	Arti cial Intelligence
AP	Asymptotic Performance
AS	Accidental Solvability
AUS	Area Under the Success Curve
CDT	Correctly Detected Trials
DARPA	Defense Advanced Research Projects Agency
DD	Detection Delay
D-DDQN	Double Dueling Deep Q-network
DNF	Disjunctive Normal Form
DQN	Deep Q-network
EBL	Explanation-Based Learning
GAN	Generative Adversarial Network
GPT	Generative Pre-trained Transformer
GVGAI	General Video Game Arti cial Intelligence
IQ	Intelligence Quotient
IJCAI	International Joint Conference on Arti cial Intelligence
JSON	JavaScript Object Notation
NPS	Novelty Performance in Scenarios
OST	Outperforming Solution Tactic
OWL	Open-world Learning
PCG	Procedural content generation

PCG-QD	Procedural Content Generation through Quality Diversity
QSR	Qualitative Spatial Relationship
SAIL-ON	Science of Arti cial Intelligence and Learning for Open-world Novelty
SD	Standard Deviation
SPN	Scenario Performance under Novelties

List of Figures

2.1 Example tasks from visual reasoning environments. 11
2.2 Example tasks from robotic environments. 13
2.3 Example tasks from game-based environments. 14
2.4 Example tasks from competition-based environments. 15
2.5 An example game level in Science Birds. 17
2.6 Example novelties that can be seen in the real world. 20
2.7 Example novelties from novelty-centric environments. 24
2.8 Example game levels generated from an Angry Birds level generator. . . . 31

3.1 Example handcrafted levels for the physical deceptions in Angry Birds. . . 36
3.2 Deceptive level generation methodology. 37
3.3 An illustration of the extraction process for a single physical entity. 38
3.4 Examples of generated levels for the six deception categories. 45
3.5 Level solve rates of three agents for human-created and generated levels. . 48

4.1 15 example tasks in Phy-Q representing the 15 physical scenarios. 58
4.2 Task templates of the relative height scenario and tasks generated using one of those task templates. 60
4.3 The local generalization and the broad generalization evaluation settings. 61
4.4 An illustration of how generalizing a physical rule is evaluated in thebroad generalization evaluation using bouncing scenario as an example. 61
4.5 Performance of human players in Phy-Q. 66
4.6 Broad generalization results of the baseline agents in Phy-Q. 68
4.7 Phy-Q Score of humans and the agents. 69

5.1 The layout constraint graphs for example scenarios 6 and 9. 82
5.2 Illustrations of qualitative spatial relations. 83
5.3 Dimension graphs and a spatial con guration for an exmaple conjunctive constraint. 85
5.4 Generated tasks for the 16 example scenarios. 87
5.5 Four tasks generated for the example scenario 8. 88

6.1 Overview of the proposed novelty generation framework. 92
6.2 Architecture of the novelty generation framework in Unity. 96
6.3 Variation in the four novelty outcomes as the linear drag of wood blocks changes. 102
6.4 A base game level and the corresponding generated game level with category 1 novel game objects 103
6.5 A base game level and the corresponding generated game level with category 2 novel game objects after the same bird shot. 103
6.6 Evaluation process of the DQ-Birds agent. 105

7.1 Example tasks of the rolling scenario for the eight novelties. 115
7.2 Example tasks of the inverted gravity novelty applied to the ve physical scenarios. 117
7.3 The trial-based evaluation protocol used in NovPhy. 118
7.4 An example pass rate curve of an agent that played a set of trials. 120
7.5 CDT and DD results of the human players in NovPhy. 123
7.6 AP and AUS results of the human players in NovPhy. 124
7.7 The pass rate of the agents per novelty in NovPhy. 128
7.8 The pass rate of the agents per scenario in NovPhy. 129

8.1 The layout constraint graph of example scenario 1. 143
8.2 The QSR graph of example scenario 1. 143
8.3 Simulation phase steps of the generation process. 145
8.4 Generated tasks for the 12 example scenarios. 147
8.5 Continuation of Figure 8.4. 148
8.6 Four tasks generated for the example scenario 1. 149
8.7 A visual representation of the intended solvability, intended unsolvability, and solution switch measures. 150
8.8 The results of the intended solvability, intended unsolvability, solution switch, and accidental solvability evaluations. 151

A.1 Game objects in Science Birds that are used in this study. 163

B.1 An example task and its symbolic representation. 166
B.2 Task templates from the single force scenario. 167
B.3 Task templates from the multiple forces scenario. 167
B.4 Task templates from the rolling scenario. 168
B.5 Task templates from the falling scenario. 168
B.6 Task templates from the sliding scenario. 168
B.7 Task templates from the bouncing scenario. 169
B.8 Task templates from the relative weight scenario. 169
B.9 Task templates from the relative height scenario. 169
B.10 Task templates from the relative width scenario. 170
B.11 Task templates from the shape di erence scenario. 170

B.12 Task templates from the non-greedy actions scenario. 170
B.13 Task templates from the structural analysis scenario. 171
B.14 Task templates from the clearing paths scenario. 171
B.15 Task templates from the adequate timing scenario. 171
B.16 Task templates from the manoeuvring scenario. 172
B.17 Tasks generated from the same task template. 172
B.18 Passing rates of the agents for the within template evaluation in Phy-Q. . 176
B.19 The average pass rate, ve attempt pass rate, the maximum number of attempts made, and the total thinking time taken by human participants. 178
B.20 Performance of humans, the performance of heuristic agents, and local generalization of learning agents on the sample of task templates used for the human evaluation. 178
B.21 The game levels from the 2021 AIBIRDS competition used for the AIBIRDS heuristic agent evaluation. 179

C.1 The Air Turbulence external agent 181
C.2 Objects associated with the novelties in NovPhy 182
C.3 Task templates of the single force scenario with eight novelties applied to them. 183
C.4 Task templates of the multiple forces scenario with eight novelties applied to them. 184
C.5 Task templates of the rolling scenario with eight novelties applied to them. 185
C.6 Task templates of the falling scenario with eight novelties applied to them. 186
C.7 Task templates of the sliding scenario with eight novelties applied to them. 187
C.8 Example tasks generated from the same task template. 188
C.9 The CDT of the agents under the two algorithms: simple and pre-assumed moving average. 190
C.10 The CDT and DD results of the Datalab. 191
C.11 The CDT and DD results of the Eagle's Wing. 191
C.12 The CDT and DD results of the Pig Shooter. 192
C.13 The CDT and DD results of the Random agent. 192
C.14 NPS_{CDT} of the agents for the eight novelties. 192
C.15 NPS_{DD} of the agents for the eight novelties. 193
C.16 SPN_{CDT} of the agents for the ve scenarios. 193
C.17 SPN_{DD} of the agents for the ve scenarios. 193
C.18 The pass rate of the agents in all novelty-scenarios. 194
C.19 The AP and AUS results of the DQN O ine. 195
C.20 The AP and AUS results of the DQN Online. 195
C.21 The AP and AUS results of the DQN Adapt. 195
C.22 The AP and AUS results of the Relational O ine. 196
C.23 The AP and AUS results of the Relational Online. 196
C.24 The AP and AUS results of the Relational Adapt. 196
C.25 The AP and AUS results of the Naive Adapt. 197

C.26 The AP and AUS results of the Datalab. 197
C.27 The AP and AUS results of the Eagle's Wing. 197
C.28 The AP and AUS results of the Pig shooter. 198
C.29 The AP and AUS results of the Random agent. 198
C.30 NPS_{AP} of the agents for the eight novelties. 198
C.31 NPS_{AUS} of the agents for the eight novelties. 199
C.32 SPN_{AP} of the agents for the ve scenarios. 199
C.33 SPN_{AUS} of the agents for the ve scenarios. 199
C.34 The adaptation performance vs CDT performance of humans. 200
C.35 The correlation of adaptation performance and detection delay of humans. 201

List of Tables

3.1 Evaluation results for the stability rate, solvability rate, the average deceptive score, and the average di erence of the solve rates for human-created and generated levels. 46

4.1 Comparison of Phy-Q with related physics benchmarks and competitions 54
4.2 Comparison of agents fo local generalization and broad generalization in Phy-Q. 67
4.3 Comparison of AIBIRDS competition performance with Phy-Q score of heuristic agents. 70

5.1 Object grammar. 77
5.2 Interaction grammar. 77
5.3 Restriction grammar. 78
5.4 Layout grammar. 78
5.5 De nitions of 16 example physical scenarios using the grammar. 79
5.6 Continuation of Table 5.5 80
5.7 Inferred layout constraints from the interactions and restrictions. 82
5.8 Inferred QSRs from the layout constraints. 84
5.9 The results of the intended solvability, accidental solvability, and generation time evaluations. 89

6.1 Results of the three agents for the two experiments with two Novelty Objectives 104
6.2 Mean game scores for di erent versions of the DQ-Birds agent. 105

7.1 Representative novelties in NovPhy for each hierarchy level of the DARPA SAIL-ON open-world novelty hierarchy. 113
7.2 Results of the NPS measures of the agents and humans for the eight novelties. The best results of each row are highlighted in blue. 126
7.3 Results of the SPN measures of the agents and humans for the ve scenarios. The best results of each row are highlighted in blue. 127

8.1 Disruption grammar. 137

8.2 Construction grammar. 137
8.3 De nitions of 12 example physical scenarios with novelties. 141
8.4 Continuation of Table 8.3. 142

B.1 The di erences between the task templates of the scenarios in Phy-Q . . . 173
B.2 Continuation of Table B.1. 174
B.3 Solution descriptions for 15 example task templates in Phy-Q. 175
B.4 The train-test split of the task templates for the within scenario evaluation in Phy-Q. 177

C.1 The physical interaction phases of the solution that were a ected by the novelty in NovPhy. 189

Chapter 1

Introduction

1.1 Overview

Working in the physical world is a fundamental aspect of human intelligence, which necessitates reasoning about the behaviours of objects under the in uence of physics. For instance, when attempting to stack a set of tableware on a cupboard, an individual may consider the weight, shape, and balance of each item and anticipate their interactions when arranged together. Employing their physical reasoning capabilities, they can make informed decisions on how to arrange the tableware to prevent toppling. Similarly, when a child builds a sandcastle on the beach, they understand the sand's behaviour, including concepts like cohesion and stability, to ensure the castle retains its shape. This signi cance of physical reasoning extends to a myriad of day-to-day tasks, encompassing activities as simple as pouring a glass of water to complex activities like driving a car. In these tasks, we often lack precise knowledge of the objects' speci c physical properties or their precise reactions to our actions. Nevertheless, we consistently demonstrate the ability to execute these actions successfully on each occasion.

Researchers have extensively investigated the developmental trajectory of physical reasoning capabilities in humans, demonstrating that these skills emerge from infancy. Studies have highlighted that shortly after birth, infants already display an awareness of object solidity [Valenza et al., 2006]. Throughout their initial year of life, they begin to comprehend essential concepts, including object permanence [Baillargeon and DeVos, 1991], stability [Baillargeon et al., 1992], support [Baillargeon and Hanko-Summers, 1990], and shape constancy [Day and McKenzie, 1973]. These early cognitive achievements lay the foundation for the sophisticated physical reasoning capabilities that humans demonstrate throughout their lives. One prevailing viewpoint among cognitive psychology researchers is that humans possess forward physics prediction models, which play a vital role in supporting their physical reasoning abilities [McCloskey, 1983]. Consequently, equipped with these innate physical reasoning capabilities, humans adeptly navigate and operate

in the physical world, a testament to the signi cance of physics-based intelligence in our everyday lives.

In addition to the physics-based challenges faced in the real environment, the open-world nature of reality often introduces novel situations that demand adaptability. For instance, consider a person driving through an unfamiliar city and encountering a complex intersection with multiple lanes, tra c lights, and road signs they have never seen. Despite the novelty of the situation, the driver successfully navigates the intersection by leveraging their ability to adapt and make decisions based on previous driving experiences and knowledge of tra c rules. Similarly, consider someone accustomed to preparing familiar recipes in their well-equipped kitchen. However, when cooking at a friend's house with unfamiliar appliances and utensils, they con dently adapt to the situation. They quickly assess the available tools, determine their functions, and utilize them for cooking. Throughout the process, they may adjust their cooking techniques and overcome novel challenges, such as adapting to measure ingredient quantities based on di erent measurements. These examples demonstrate how the human species' physical reasoning and novelty-detecting and adapting capabilities have been crucial for their successful survival in the ever-changing physical world.

In arti cial intelligence (AI) research, a longstanding pursuit has been the development of intelligent agents capable of operating e ectively in the physical environment to enhance human lives. While working in the real world seems e ortless to humans, it presents formidable challenges for AI systems. For human-like performance, AI agents must possess essential attributes such as physical reasoning capabilities and adaptability to novelty, which play a crucial role in achieving success in real-world scenarios. While AI has demonstrated exceptional achievements, surpassing human-level performance in tasks like Chess [Campbell et al., 2002], Go [Silver et al., 2016], image recognition [Krizhevsky et al., 2012], and speech recognition [Xiong et al., 2016], it still faces obstacles in mastering physical reasoning tasks. Furthermore, current AI systems exhibit limitations when confronted with situations that were either not encountered during their training phase or not anticipated by their developers. Consequently, the incorporation of physical reasoning capabilities and the ability to detect and adapt to novel situations have emerged as active and essential research areas in the pursuit of more robust AI agents in real-world settings.

To enable the development of AI agents equipped with the aforementioned physical reasoning capabilities and novelty detection and adaptation skills in physical environments, it is essential to establish a robust infrastructure. This infrastructure includes evaluation measures to quantify and assess the agents' capabilities, testbeds for agent training, testing, and comparison, and, at its core, a high-quality set of tasks that con dently support the experimentation of AI agents. In this thesis, our primary objective is to introduce methodologies for designing and generating physics-based tasks that e ectively aid in the advancement of AI agents functioning in open-world physics-based settings. These tasks are carefully crafted to meet two crucial requirements: rst, they demand physical reasoning capabilities for successful completion, and second, they necessitate

novelty detection and adaptation capabilities in the context of physical environments. By leveraging these tasks, we construct evaluation protocols, measures, and testbeds that facilitate the evolution of AI agents in real physical environments.

In AI research, physics-simulating environments are commonly employed for developing real-world AI systems due to their numerous advantages over direct experimentation in the real world. These environments o er controlled variables, ensuring consistency and reproducibility of experiments while providing the exibility to design scenarios, including extreme and rare conditions, that may not frequently occur in reality. The ability to rapidly iterate and experiment with a wide range of settings e ciently, coupled with cost-e ectiveness and safety, makes physics-simulating environments an invaluable tool for researchers. Video games, in particular, serve as one such physics-simulating environment frequently utilized by researchers. By designing tasks in these games that require sophisticated physical reasoning capabilities to solve, similar to real-world challenges, researchers can e ectively assess the physical reasoning performance of AI agents. Furthermore, the ease of introducing novel situations across a wide spectrum within these game environments adds to their appeal. In this research, we adopt a physics-based game environment, Angry Birds, due to the aforementioned reasons. This enables the development of comprehensive content that contributes to the advancement of AI systems, which might not have been practically feasible to cover through direct implementation in the real world.

1.2 Motivation

The motivation behind this thesis is rooted in recognizing the signi cant challenges that AI agents face in tasks involving physical reasoning and novelty detection and adaptation within real-world physical environments. Despite the remarkable progress made in AI research, current AI systems still struggle to match human-level performance in tasks requiring physical reasoning capabilities. Moreover, when confronted with novel situations, AI agents often exhibit limited adaptability compared to their human counterparts.

Addressing these shortcomings is of utmost importance for several reasons. Firstly, achieving human-level physical reasoning capabilities in AI systems is crucial for their successful deployment in real-world applications. Many real-world tasks demand a deep understanding of physical interactions, and without adequately addressing this aspect, AI agents may fall short in critical domains such as robotics, autonomous vehicles, and environmental monitoring. Secondly, the ability to adapt to novel situations is a fundamental aspect of human intelligence that we aim to replicate in AI systems. Humans exhibit remarkable exibility in navigating unfamiliar scenarios, leveraging their ability to generalize knowledge and experience from known situations to new ones. By empowering AI agents with similar novelty detection and adaptation capabilities, we can enhance their versatility and overall performance in diverse and dynamic environments.

This thesis aims to address the fundamental challenges of physical reasoning and novelty

detection and adaptation in AI systems by developing comprehensive content for their evaluation, developing automatic content generation techniques, and proposing evaluation testbeds. Our objective is to make signi cant contributions to the advancement of AI systems, enabling them to navigate and excel in complex, dynamic physical environments. Through the developed content and rigorous evaluations, we strive to empower AI agents with enhanced physical reasoning capabilities and the ability to adapt to novel scenarios, bringing them closer to achieving human-level performance in these crucial aspects.

1.3 Research Problems

In this thesis, our primary focus is on addressing three fundamental research problems that revolve around facilitating the advancement of the capabilities of AI agents in physical reasoning and handling novelties within open-world physics-based environments.

- Can we design tasks in physics-based environments that e ectively evaluate the physical reasoning capabilities of AI agents? Additionally, can we develop techniques to generate such tasks automatically?
- Can we introduce open-world properties into these tasks by incorporating novelties that challenge AI agents' ability to detect and adapt to novel situations in physics-based environments? Moreover, can we devise techniques for the automatic generation of tasks with novelties?
- Can we utilize these tasks to create comprehensive evaluations and testbeds for advancing AI systems operating in open-world physics-based environments?

If we can e ectively address these three research problems and make substantial progress, we rmly believe that it will be a signi cant step towards achieving AI agents with the above-mentioned capabilities.

1.4 Methodology

In this section, we delve into the methodologies utilized in this study to address the three research problems outlined earlier.

As previously indicated, our study employs the physics-simulating game Angry Birds as the experimental framework. In order to address the rst research problem, our approach involves an initial exploration within the Angry Birds domain to identify the speci c types of game levels that pose challenges for AI agents. Subsequently, we develop methods aimed at generating these challenging game levels. These generation techniques facilitate the creation of an extensive array of training and testing scenarios for AI agents. We also take into account the diversity of the generated levels, acknowledging that creating levels that look visually similar or share the same solving strategy may not provide varied challenges for the agents. Furthermore, these levels are bound

by the constraints of the physics-based environment, mandating the prevention of object overlap, the requirement for objects to be arranged in a physically stable con guration, and compliance with other constraints dictated by the underlying physics. These considerations collectively ensure the pertinence and suitability of the levels within the context of the physics-based domain.

Continuing our exploration beyond the challenges faced by Angry Birds agents, we delve into physical scenarios that human infants naturally grasp from an early age, as well as those essential for equipping robots with real-world operational capabilities. We meticulously design tasks tailored to these distinct physical scenarios, subjecting AI agents to isolated evaluations on each scenario to reveal their inherent strengths and limitations. Furthermore, we introduce a novel technique aimed at generating physics-based tasks relevant to these physical scenarios. This technique draws its foundation from the interconnectedness of causal physical interactions of objects in physics-based environments. Through this approach, we facilitate a nuanced comprehension of the intricate physics mechanics pivotal for solving these tasks, thus enabling a comprehensive evaluation of agents' performance in a granular manner.

In addressing the second research problem, our approach commences by exploring avenues for the automatic generation of novelties that can exert pre-de ned e ects within the Angry Birds game levels while ensuring that these novelties remain detectable by AI agents. Subsequently, we delve into introducing novelties into the aforementioned physical scenarios. The objective here is to design tasks that necessitate the agents' comprehension of these novelties and their ability to adapt accordingly to successfully accomplish the tasks. Through this endeavour, we not only evaluate an agent's capability to detect and adapt to a novelty across various physical scenarios but also gain insights into its performance on a physical scenario when confronted with di erent novelties. Moreover, extending the framework of the aforementioned physical interaction-based task generation technique, we explore the automatic generation of tasks augmented with novelties. This extension is based on disrupting existing physical interactions among objects while simultaneously constructing new interactions using novelties. By adopting this extended approach, we gain a deeper understanding of the underlying physics interactions pivotal for solving these tasks with novelties. Such insights are instrumental in comprehending agents' performance dynamics and subsequently facilitating their enhancement and advancement.

When investigating the second research problem, our investigation focuses on novelty types commonly encountered within physical environments. These novelties encompass visually detectable changes, such as alterations in colour, shape, or size of objects within the environment. Additionally, we explore novelties that are not visually apparent but require interaction to be detected, such as variations in physical parameters like mass, friction, or bounciness of objects. Furthermore, we emphasize the generation of tasks and the evaluation of agents on tasks that necessitate adaptive actions in response to novelty. For instance, in scenarios where an object's friction coe cient has changed, agents must adjust the applied force to move the object to its intended destination successfully.

Addressing the third research question entails harnessing the tasks derived from the preceding investigations to construct extensive evaluation settings and testbeds. These evaluation settings are designed with the objective of enabling a comprehensive and con- dent assessment of AI agents' competencies in terms of physical reasoning and novelty detection and adaptation. Moreover, these testbeds serve as a guiding framework for the continued advancement of these agents. In addition, we augment our evaluation approach by conducting experiments involving human participants. This human-centric evaluation serves a dual purpose: it not only establishes baseline performance bench- marks for human intelligence within the de ned tasks but also o ers pivotal insights that help contextualize the capabilities of AI systems operating within open-world physics- based environments.

1.5 Thesis Outline

The thesis is structured into nine chapters. The next chapter discusses the background and related work, and the three chapters after that (Chapters 3, 4, and 5) delve into the generation of physical reasoning tasks. Subsequently, the following three chapters (Chapters 6, 7, and 8) build on top of the previous three chapters to integrate novelty into those physical reasoning tasks. The last section covers the conclusion and outlines future work. Below, we provide a brief overview of the content of those chapters.

Chapter 2 delves into the foundational aspects and related research within the elds of physical reasoning, open-world novelty, and procedural content generation. Within the physical reasoning domain, we provide an overview of the research area and examine environments utilized for AI agents' physical reasoning experiments. Additionally, we introduce our experimental domain, Angry Birds, and provide a brief overview of agents possessing physical reasoning capabilities, which we use in this study for our experiments. Transitioning to open-world novelty research, we delve into the concept of novelty for AI agents and discuss novelty-centric environments designed to foster agent development. This chapter concludes with a focus on procedural content generation, encompassing a discussion on di erent content that is automatically produced in the context of computer games and physics-simulating environments. The subsequent chapters systematically describe our thesis's research contributions, each building upon the preceding ones to progressively realize our research objectives.

The act of mastering Angry Birds game levels inherently demands the utilization of physical reasoning capabilities. Despite an enduring competition spanning over a decade aimed at fostering the evolution of AI agents capable of mastering the game, these agents have so far fallen short of achieving human-level performance. In Chapter 3, we delve into a facet of this challenge by focusing on game levels strategically designed to deceive agents into making suboptimal decisions within the physics-based environment. Tradi- tionally, these levels have been painstakingly crafted by humans due to their demand for meticulous design. In response, we introduce a procedural methodology that automates the generation of such deceptive game levels, with the aim of empowering AI agents

to excel in the face of deception through extensive training and testing datasets. This chapter's contents have been published in the research paper [Gamage et al., 2021a].

Progressing beyond the realm of deceptive levels within Angry Birds, our inquiry draws inspiration from the foundation of physical knowledge that humans acquire in their infancy, coupled with the requisite skills that robots must possess to function seamlessly within real-world environments. This inspiration guides us to identify a set of essential physical scenarios that hold signi cance for AI agents to master when operating within the real physical world. Translating these scenarios into Angry Birds, we crafted tasks (i.e., game levels) that e ectively replicate these real-world situations. Inclusive of these tasks, we developed a specialized testbed tailored to AI agents, enabling the concurrent training of multiple agent instances with expedited gameplay. This testbed serves as a pivotal platform for assessing the local and broad generalization capabilities of AI agents, particularly when confronted with tasks that demand physical reasoning capabilities. Mimicking the foundational principles underlying the human Intelligence Quotient (IQ) assessment, we introduce a novel metric: the Physical Reasoning Quotient (Phy-Q score). This metric serves as a quanti able measure of an agent's physical reasoning intelligence, gauged through its performance across the identi ed physical scenarios. To contextualize AI agent performance, we benchmark it against human pro ciency levels established within the testbed tasks. This work is presented in Chapter 4 and is based on the work published in the research paper [Gamage et al., 2023c].

The approach outlined above involves the representation of the physical scenarios as abstract physical rules, though informative, faces limitations in capturing the intricate mechanics essential for a comprehensive understanding of the requisite capabilities for successfully performing in those scenarios. To address this limitation and foster a deeper understanding of the underlying mechanics, Chapter 5 introduces a grammar-based methodology that can be used to systematically de ne physical scenarios by representing a causal sequence of physical interactions between objects. Subsequently, a methodology for task generation within physics-simulating environments is proposed, which utilises these scenarios de ned as causal sequences as foundational inputs. This approach not only enriches the understanding of the nuanced mechanics necessary for solving physics-based tasks but also facilitates a more accurate evaluation of AI systems' pro ciency in physical reasoning. This chapter has been adapted from the research published in the paper [Gamage et al., 2023a].

Subsequently, our focus shifts to the integration of novelties into physics-based tasks. Initially, we strive to develop methodologies for generating novelties and seamlessly integrating them into physics-based tasks. This entails automatically generating novelties that bring about prede ned e ects within the tasks, ensuring their detectability by AI agents. This framework for novelty generation is constructed using Angry Birds. Our investigation extends to experimental validation, wherein we assess the e cacy of the devised methodology in generating novelties capable of producing the intended e ects. These novelties serve as tools for the experimentation of AI agents, explicitly facilitating the re nement of their novelty-detection capabilities. Chapter 6 of this thesis compre-

hensively expounds upon this work, and the content included in this chapter is based on the publication [Gamage et al., 2021b].

Building upon the physical reasoning testbed detailed in Chapter 4, our inquiry extends to the integration of novelties within the established physical scenarios explored in that chapter. We delve into the methodology of introducing novelties into physics-based tasks in a manner that necessitates an agent's comprehension of the novelty and subsequent adjustment of its actions to accomplish these tasks pro ciently. We develop a novel testbed tailored to require agents to engage with physical scenarios augmented by novelties. This testbed serves as a litmus test for agents' capacities to reason through intricate physical scenarios in the presence of novelties and respond competently. The testbed o ers a dual evaluation, enabling the assessment of an agent's performance when confronted with a novelty across various physical scenarios, as well as its performance when tackling di erent novelties within the same physical scenario. This testbed is further enriched by the inclusion of human performance benchmarks, which aspire to elevate these agents to a level of competency akin to, or surpassing, human capabilities as they navigate the open-world physical environments. Chapter 7 delves into the comprehensive exploration of this work. The content presented in this chapter is drawn from the research paper [Gamage et al., 2023b].

Emerging from the exploration conducted in the above chapter, which delves into the integration of novelties into physical scenarios, we recognize the inherent labour-intensiveness of manually crafting such tasks. To address this, Chapter 8 takes a step forward by proposing a method for the automatic generation of such tasks. Building upon the foundational concepts introduced in Chapter 5, the approach revolves around de ning physical scenarios through causal sequences of objects' physical interactions. We expand that task generation approach to integrate the generation and introduction of novelties into physical scenarios. This process involves disrupting existing causal interactions between objects by introducing novelties while simultaneously constructing new interactions. This automated approach not only alleviates the manual burden of task creation but also o ers two key advantages: rst, it ensures agents' pro ciency in adapting to novelties can be systematically measured via task performance; second, it grants insight into the intricate physics mechanics underpinning both novelty adaptation and task solution, thus facilitating a comprehensive evaluation of AI agents. This work, discussed in Chapter 8, is from the research presented in the paper [Gamage et al., 2024].

In the nal chapter, we bring together the contributions and discoveries from the research conducted in this thesis. We also suggest possibilities for further exploration in the future. The individual research sections and their discoveries discussed throughout this thesis have their own value. Yet, when seen as a whole, they come together to achieve a larger aim: enhancing intelligent agents that excel in open-world physics-based environments. Speci cally, these agents are expected to perform like humans, skillfully navigating through challenges involving tasks based on physical reasoning and unexpected situations. We view the contributions made in this thesis as signi cant steps toward making this ambitious goal a reality.

Chapter 2

Background and Related Work

In this chapter, we explore the existing research that underpins the investigations conducted in this thesis. In Section 2.1, we delve into the domain of physical reasoning, encompassing analysis of prevailing physics-based environments that facilitate the development of AI agents pro cient in physical reasoning. This section also presents a detailed introduction to our chosen experimental domain, Angry Birds, highlighting its signi cance in AI research. Further, we also discuss the AI agents specializing in physical reasoning, which we use for the experiments in this study. Following this, in Section 2.2, we shift our focus to the eld of novelty-centric AI research, encompassing diverse concepts of novelties and the distinct domains tailored to the experimentation of AI agents on novelties. Finally, in Section 2.3, we explore procedural content generation research, particularly within the context of video games and other physics-based environments.

2.1 Physical Reasoning

Physical reasoning is the fundamental cognitive ability to comprehend, predict, and reason about object behaviours and the physical world by applying foundational principles of physics [Davis, 2006]. This involves analyzing how objects interact within the framework of forces, motion, gravity, friction, and other physical laws, allowing individuals to anticipate outcomes of actions in the tangible world. Piaget et al.'s pioneering work on cognitive development underscores the signi cance of physical reasoning, particularly in early childhood, highlighting its pivotal role in human cognition. Infants grasp concepts like object solidity within days of birth [Valenza et al., 2006], and over their rst year, they develop understandings of object permanence [Baillargeon and DeVos, 1991], spatiotemporal continuity [Leslie, 1984], stability [Baillargeon et al., 1992], support [Baillargeon and Hanko-Summers, 1990], causality [Saxe and Carey, 2006], and shape constancy [Day and McKenzie, 1973]. In activities like stacking items, handling liquids, catching objects, or driving vehicles, humans employ physical reasoning to fore-

see object movements, interactions, and responses to external in uences. This cognitive skill underpins daily tasks, problem-solving, and decision-making in physical domains.

In the eld of AI research, cultivating physical reasoning capabilities in AI agents entails enabling machines to accurately simulate, understand, and predict physical interactions. These capabilities are crucial for AI systems to navigate the real world, make informed choices, and execute tasks necessitating a profound grasp of underlying physics. The pursuit of robust AI physical reasoning is an ongoing research endeavour with substantial implications for elds like robotics, autonomous systems, and human-AI collaboration. Notably, research by Battaglia et al. [2013] underscores the signi cance of infusing AI agents with physical comprehension to enable them to deduce object interactions and dynamics. E orts in this domain aim to construct machines capable of achieving physical reasoning on par with human capabilities [Lake et al., 2017]. By advancing the understanding of physical reasoning in AI, researchers strive to engender agents capable of thriving in real-world physical contexts.

2.1.1 Physical Reasoning Environments

In this section, we explore the environments developed for evaluating the physical reasoning capabilities of AI agents. Various researchers have created testbeds, benchmarks, and competitions to support the examination, assessment, and enhancement of AI agents' physical reasoning capabilities. First, we discuss physics-based visual reasoning environments, followed by an exploration of environments employing physics simulators to simulate object behaviour and motion. This encompasses robotic, game-based, and AI competition-based environments.

Visual Reasoning Environments

There are video-based benchmarks, including Physion [Bear et al., 2021], IntPhys [Riochet et al., 2020], CLEVRER [Yi* et al., 2020], and CATER [Girdhar and Ramanan, 2020], and image-based benchmarks such as CoPhy [Wolf, 2020] designed to gauge the physical reasoning abilities of AI agents through visual content. Figure 2.1 shows example tasks from these environments.

The Physion benchmark comprises realistic video examples portraying eight distinct physical scenarios. These scenarios encompass object collisions, support, containment, and attachment, as well as projectiles, rolling, and falling motions. Additionally, the behaviour of soft materials, such as cloth, is represented. Within Physion, agents are tasked with predicting whether two speci ed objects will come into contact as a scene unfolds. IntPhys, assesses an agent's grasp of intuitive physics by evaluating its ability to discern viable events from impossible ones within videos. This benchmark scrutinizes three fundamental aspects of intuitive physics: 1) Object permanence, encompassing the continuous existence of objects over time without abrupt appearances or disappearances; 2) shape constancy, highlighting the capacity of rigid objects to retain their forms over time; and 3) object trajectories, emphasizing the uninterrupted spatial movement of

(a) Example frames from the eight scenarios in Physion.

(b) Example frames from videos in IntPhys.

(c) Example frames from a video in CLEVRER.

(d) Example frames from a video in CATER.

(e) Example images from the three scenarios in CoPhy.

Figure 2.1: Example tasks from the visual reasoning environments Physion [Bear et al., 2021], IntPhys [Riochet et al., 2020], CLEVRER [Yi* et al., 2020], CATER [Girdhar and Ramanan, 2020], and CoPhy [Wolf, 2020].

objects. Moreover, CLEVRER introduces a video dataset containing collision events, wherein agents are tasked with addressing four types of queries: 1) descriptive questions, centred on identifying video content; 2) explanatory inquiries, requiring the delineation of event causes; 3) predictive elements, necessitating forecasts of future occurrences; and 4) counterfactual scenarios, encouraging contemplation of alternative outcomes. CATER, is also a video dataset that challenges agents to exhibit spatiotemporal comprehension. Within this context, three tasks demand attention: detection of all active actions in the video, identi cation of active compositional actions, and recognition of the spatial location of the `snitch' object at a video's conclusion.

CoPhy serves as an image-centric benchmark designed to facilitate the acquisition of counterfactual knowledge concerning physical dynamics. It encompasses tasks associated with three distinct physical scenarios: balls rebounding against surfaces, the collision of objects, and a tower of blocks descending. The agent's objective within this benchmark is to predict the alternative three-dimensional positions of objects following interventions, as inferred from images.

Robotic Environments

CausalWorld [Ahmed* et al., 2021] and RLBench [James et al., 2020] represent two notable robotic benchmarks in this eld. CausalWorld capitalizes on a simulated robotic platform speci cally designed for causal structure and transfer learning. Within CausalWorld, a collection of eight distinct task types is available, encompassing actions like pushing, picking, placing, and stacking. The benchmark tasks require an agent to manipulate blocks within the environment to attain speci c goal arrangements, starting from initial block con gurations. RLBench, on the other hand, o ers a comprehensive benchmark tailored for robotic learning, featuring an array of 100 meticulously designed tasks. These tasks span activities such as reaching designated targets, opening doors, and placing trays in ovens. RLBench also poses a few-shot learning challenge, with the intention of enhancing robots' capacity to swiftly assimilate knowledge from previous tasks to e ciently learn new ones in unfamiliar settings. Figure 2.2 shows example tasks from these two environments.

Game-Based Environments

Several game-based benchmarks aligning with the investigations undertaken in this research include PHYRE [Bakhtin et al., 2019], Virtual Tools game [Allen et al., 2020b], and OGRE [Allen et al., 2020a]. These benchmarks require agents to navigate physics-simulating game environments, demanding pro cient physical reasoning capabilities.

The PHYRE benchmark presents an assortment of two-dimensional classical mechanics puzzles situated in a physical environment. In PHYRE, an agent is tasked with taking a single action, which involves placing one or more new dynamic bodies into the environment, with the objective of reaching a pre-de ned terminal goal state. Its primary aim is to foster the development of learning algorithms characterized by sample e ciency and

(a) Example tasks from the eight scenarios in CausalWorld. A robot is tasked with constructing the goal shape depicted in red using the objects available in the scene, which are shown in blue.

(b) Ten example tasks from RLBench. For instance, in the top left task, the robot needs to stack a set of six red coloured blocks into a pyramid.

Figure 2.2: Example tasks from the robotic environments CausalWorld [Ahmed* et al., 2021] and RLBench [James et al., 2020].

e ective generalization across diverse puzzles. PHYRE encompasses tasks that assess agents' capacity for two levels of generalization in physical reasoning tasks. These levels gauge the agents' ability to generalize solutions within a task template (within-template) and across various task templates (cross-template). Conversely, the Virtual Tools game serves as a platform to explore priors, representations, and planning and learning algorithms relevant to physical problem-solving. This game engages players in the task of strategically selecting and placing various objects within a dynamic two-dimensional physical environment to move a speci c object into a goal area. With 30 distinct levels, the game o ers a testing ground for concepts like launching, blocking, and supporting. Figure 2.3 displays example tasks from PHYRE and Virtual Tools. Using the tasks drawn from those two benchmarks, OGRE proposes a framework to evaluate object generalization in the context of active physical reasoning. OGRE places particular em-

(a) Three example tasks from PHYRE (left) and one example solution (right). Players place a new ball within the scene with the objective of achieving the speci ed goal.

(b) Twenty example tasks from Virtual Tools. Players pick a tool (shown on the right of each task) and place it in the scene, aiming to move a red object into the green goal area.

Figure 2.3: Example tasks from the game-based environments PHYRE [Bakhtin et al., 2019] and Virtual Tools [Allen et al., 2020b].

phasis on assessing agents' adeptness in tackling novel physical reasoning tasks, gauged through two levels of generalization: generalization over reasoning strategies with familiar objects and generalization across new object types sharing comparable material properties with those encountered during training.

AI Competition-Based Environments

AI competitions such as Computational Pool [Archibald et al., 2010], Geometry Friends [Prada et al., 2015], and AIBIRDS [AIBIRDS, 2023] also function as platforms that demand pro cient physical reasoning capabilities from participating agents. These competitions are based on games and hence can also be categorized under the above category. Example tasks from these environments are shown in Figure 2.4.

(a) The pool table at the beginning of the Computational Pool competition with 15 object balls and a cue ball.

(b) A game level from AIBIRDS. The player's task is to destroy the green pig by launching birds from the slingshot.

(c) A solution for a game level in Geometry Friends. The objective is to collect the purple Diamond, requiring Rectangle and Circle to execute a cooperative skill-based maneuver in order to reach the Diamond.

Figure 2.4: Example tasks from the competition-based environments Computational Pool [Archibald et al., 2010], AIBIRDS [AIBIRDS, 2023], and Geometry Friends [Prada et al., 2015].

The Computational Pool tournament fosters the creation of AI agents capable of excelling at the game of pool. This competition employs a virtual pool table featuring eight balls and leverages a physics simulator for its development. This game challenges AI agents with various intricacies, comprising unbounded state and action spaces, the uctuating nesse of execution skills that introduce uncertainties in control, the dynamic adjustment of turn allocation contingent on state transitions, and the intrinsically adversarial landscape they navigate. On the other hand, Geometry Friends centres around a physics-based cooperative puzzle-platform game. In this game, agents must formulate collaborative plans and execute actions within a physics-driven environment incorporating forces like friction and gravity. The game presents agents with multifaceted challenges, encompassing coordination across diverse strata, from precise motion control to synchronized planning, dexterous handling of ne-grained physics-based actuation, and e ective puzzle solving.

Additionally, the AIBIRDS competition promotes the development of AI agents pro- cient in solving new levels of the Angry Birds game with the same e ectiveness and e ciency demonstrated by humans. While the act of playing Angry Birds might appear elementary for humans, the AI agents have not yet reached human-level performance in this competition, despite its existence for over a decade. This competition is discussed

in more detail in Section 2.1.2.

All these physical reasoning environments provide varied and challenging situations to assess AI agents' abilities to reason about the physical world. They encompass a spectrum of factors such as dynamic interactions, intuitive comprehension of physics, causal and temporal reasoning, spatiotemporal understanding, broad generalization within physical reasoning tasks, and the capacity for counterfactual reasoning in physical settings. We will delve deeper into the connections between these physical reasoning environments and the investigations carried out in this thesis within the respective chapters, providing a comprehensive comparison and contrast.

2.1.2 Angry Birds

In this thesis, we employ the Angry Birds game as the experimental domain for our investigations. This section is focused on introducing this particular domain and exploring the rationale behind our selection of it, as well as highlighting its signi cance in the eld of AI research.

Angry Birds, later known as Angry Birds Classic, emerged as a physics-based puzzle video game developed by Rovio Entertainment [Rovio, 2023] and initially launched in 2009. The primary objective in Angry Birds involves players employing a slingshot to launch a designated number of birds, with the aim of destroying pigs present within the game level. The success of the Angry Birds series is remarkable, with a collective count of over ve billion downloads spanning various platforms as of April 2022, including specialized editions [Rovio, 2022]. Following its debut, the original game garnered descriptions such as being among the most prominent games of 2010 [Camp, 2010], a standout hit in 2010 [Shaer, 2010], and attaining the distinction of being the most signi cant mobile app achievement in the world by 2011 [Eriksen and Abdymomunov, 2011].

We have chosen Angry Birds as the experimental domain for this study for several compelling reasons. Foremost, its semi-realistic physics environment enables us to emulate real-world physical reasoning scenarios, faithfully replicating object behaviours governed by physics principles. This authenticity extends to our capacity to introduce realistic physics-based novelties within the game, e ectively transforming it into an open-world setting well-suited for our research pursuits. Additionally, Angry Birds holds a prominent position within the realm of AI research, particularly in elds like physical reasoning [Zhang and Renz, 2014; Walega et al., 2016], game-playing AI agent development [Renz et al., 2015, 2019], and procedural content generation [Stephenson et al., 2019; Renz et al., 2021]. The extensive array of AI agents speci cally designed for this game [Stephenson et al., 2018], coupled with the diversity of techniques developed for generating game levels [Stephenson et al., 2019], underscores the rich landscape for exploration. Subsequent sections of this chapter will delve into these areas in comprehensive detail. Furthermore, the game's widespread presence within research circles has led to the development of associated research software and tools [Ferreira and Toledo, 2014b; Xue et al., 2022] that signi cantly expedite investigative e orts. Beyond its research signi cance, the game's

Figure 2.5: An example game level in Science Birds.

widespread appeal among the general public, attributed to its engaging gameplay, aids in recruiting participants for human-based experiments within this study.

For the investigations conducted within this thesis, we utilize a research version of the game named Science Birds [Ferreira and Toledo, 2014b], developed in Unity and incorporating the box2D physics engine [Catto, 2023]. An illustrative example of a game level within Science Birds is presented in Figure 2.5, accompanied by a depiction of standard game objects in Science Birds used in this study, in Appendix A. Comparable to its counterpart, Angry Birds, a typical game level in Science Birds encompasses a variety of elements: birds, blocks, pigs, platforms, TNT explosives, and a slingshot. Of these, birds, blocks, and pigs are dynamic objects governed by Newtonian physics principles, while platforms remain static due to their imperviousness to external forces. TNTs exhibit an explosive response upon impact by other objects, causing damage to surrounding dynamic elements. The slingshot serves as the means to shoot the birds. Dynamic objects feature health points that diminish upon collisions and explosions; when reaching zero health, the object is eliminated. The blocks are fashioned from three distinct materials: ice, wood, and stone. Material composition in uences the block's health and mass, with ice blocks being the least sturdy and heaviest, followed by wood and then stone. The game features ve bird types, with four of them endowed with distinctive powers, excluding the standard red bird. The deployment of these powers occurs through screen-tapping during ight: the blue bird divides into three, the yellow bird accelerates, the black bird detonates, and the white bird lays an explosive egg.

In the context of AI agents playing Angry Birds, agents possess a limited set of actions at their disposal. These actions encompass the speci cation of the bird's release coordinates (x, y), thereby determining the trajectory by drawing the bird backward within the slingshot. Subsequently, agents are required to tap the screen at time t after the release to activate the bird's unique power, if available. Given the continuous nature of the

action space, agents are confronted with an in nite array of potential actions denoted as (x, y, t). When multiple birds are present within a game level, agents must adhere to a sequential order, launching the birds one after the other. Also, the agent is a orded the privilege of observing the aftermath of each preceding launch prior to the subsequent launch. Upon e ectively eliminating all pigs within a given level utilizing the provided birds, the agent achieves level completion and accrues points based on factors such as the remaining number of birds and the extent of destruction incurred upon game-level objects. Conversely, if the agent is unable to eradicate all pigs following the utilization of all available birds, the level is regarded as unsuccessful.

In contrast to the intricate and unstructured nature of the real physical world, Angry Birds serves as a simpli ed and regulated platform. Nevertheless, despite its controlled setting, no AI system has thus far emerged that can rival human performance within this domain. To catalyze the advancement of AI agents endowed with human-like physics reasoning capacities, the AIBIRDS competition [AIBIRDS, 2023] has been a recurring event since 2012, primarily hosted at the International Joint Conference on Arti cial Intelligence (IJCAI). Over the years, a plethora of AI methodologies have been proposed, spanning from contemporary deep reinforcement learning techniques to more traditional heuristic approaches such as qualitative physical reasoning methods. Despite this diversity, none of these strategies has achieved the notable benchmark of attaining human-level performance. Plausibly, one contributing factor to this outcome is the absence of intricate levels and associated testbeds tailored to hone and evaluate agents' physical reasoning capacities, thereby restraining their potential advancement.

2.1.3 Physical Reasoning Agents

This section provides a brief overview of AI agents from existing literature that exhibit physical reasoning capabilities and are utilized in the explorations within this thesis. Speci cally, our focus is on agents operating within action-oriented physical environments. These environments necessitate agent actions to accomplish tasks within physics-based settings, distinct from tasks such as visual question answering. Our discussion centres around deep reinforcement learning agents, encompassing the Deep Q-network (DQN) agent, Relation DQN, Policy learners, and four heuristic agents specializing in playing the Angry Birds game. The four heuristic agents are participants of the AIBIRDS competition, namely Datalab, Bambirds, Eagle's Wings, and Pig Shooter. Detailed elaborations on the speci cs of the learning agents are provided in the respective chapters where they are utilized for our experiments.

- DQN agent : The deep Q-network agent [Mnih et al., 2015] operates by gathering quadruplets of state, action, reward, and next state during its training phase, employing a decaying epsilon-greedy strategy. This approach has proven e ective in the assessment of action-centric physical reasoning environments like PHYRE and OGRE. Notably, DQN has showcased the best performance in PHYRE and in within-dataset evaluations in OGRE while remaining moderate in cross-dataset evaluations in OGRE. Variants of this agent have emerged, including the Double

Dueling DQN [Wang et al., 2016; Van Hasselt et al., 2016], and adaptations involving distinct input types such as images and symbolic representations. Additionally, both online learning and o ine versions of the DQN agent have been developed, contributing to its versatility and adaptability across di erent scenarios.

- Relational DQN agent : The Relational DQN agent extends the DQN agent by integrating a relational module [Zambaldi et al., 2018]. Its design focuses on improving generalization across various scenarios through structured perception and relational reasoning. This agent is primarily designed to recognize and grasp signi cant relationships among objects in di erent templates or events. This method allows the agent to learn and internalize these relationships, facilitating its ability to apply this knowledge in di erent situations.

- Policy learner agents : Policy learners encompass agents that specialize in learning optimal decision-making strategies directly through policy-based reinforcement learning approaches. These approaches involve training a policy, a mapping from states to actions, to maximize cumulative rewards over time. Among these, the A2C [Mnih et al., 2016] and PPO [Schulman et al., 2017] agents stand out as notable examples of policy learners. Our experimentation involved these agents in scenarios with both discrete and continuous action spaces. However, in alignment with observations made in PHYRE, it became evident that policy-gradient methods encountered limitations when compared to DQN, struggling to achieve optimal policy convergence within a reasonable duration.

- Datalab: Datalab [Borovička et al., 2014] is an Angry Birds playing agent that employs a set of four pre-de ned strategies for planning one move at a time. These strategies involve tactics like destroying structures, eliminating as many pigs as possible, displacing round blocks to cause damage, and destroying explosives. Each strategy aims to maximize damage, with the choice determined by the environment, potential trajectories, current bird type, and remaining birds. Datalab secured victory in the 2014 and 2015 AIBIRDS competitions.

- Bambirds: Bambirds [Felix Haase, 2021], as an Angry Birds playing agent, operates by generating a qualitative level representation and subsequently selecting a strategy from nine distinct strategies based on the current state of the level. These strategies encompass tactics such as initiating a domino e ect, targeting load-bearing blocks, maximizing structure penetration, prioritizing protective blocks, focusing on pig targets, and leveraging speci c bird powers. Each strategy is assigned a score re ecting its anticipated damage potential for the current bird type. Random selection of a strategy ensues, with the associated score determining the likelihood of its choice. Notably, this agent retains a memory of prior shots and strategies, which aids in its approach to level retrying. Bambirds clinched victory in both the 2016 and 2019 AIBIRDS competitions.

- Eagle's Wing: Eagle's Wing [Wang, 2017], is also an Angry Birds playing agent that employs a set of ve distinct strategies. These strategies encompass actions

Figure 2.6: Example novelties that can be encountered in the real world. On the left: The emergence of self-checkout machines in supermarkets post-early 2000s represented a novel experience for customers accustomed to traditional checkout methods. On the right: tra c accidents, relatively uncommon in training data, presenting novel scenarios to self-driving cars due to their infrequency and unique visual characteristics in each occurrence. Image credits: [Wikipedia contributors, 2023d,a].

such as targeting pigs, activating explosives, achieving maximum block destruction, aiming at high-round objects to induce their fall and subsequent damage, and directing shots at bottom building blocks to disrupt structures and undermine stability. Strategy selection relies on assessing the anticipated utility of each approach in conjunction with the currently chosen bird. This agent secured victory in both the 2017 and 2018 AIBIRDS competitions.

ˆ Pig Shooter: The Pig Shooter (also referred to as the Naive Agent) [Stephenson et al., 2018], introduced in the AIBIRDS competition, serves as a foundational agent. It integrates a computer vision module for object recognition and a trajectory planning module that computes two feasible bird release points to hit a reachable target. Notably, the Pig Shooter consistently launches the current bird at a randomly selected pig via an arbitrarily selected trajectory without taking other factors into account.

2.2 Open-World Novelty

In our everyday lives, we frequently encounter scenarios like unforeseen road closures due to constructions or accidents, demanding rapid adjustments to our routes; sudden changes in weather, such as unexpected rainstorms, prompting modi cations to plans like carrying an umbrella or altering outdoor activities; or natural disasters like earthquakes that call for evacuations and adaptation to new living conditions. Real-world examples of such novelties are depicted in Figure 2.6. Human beings exhibit adeptness in adapting

to such circumstances. To function e ectively in the real world, in addition to possessing physical reasoning abilities, the capacity to adapt to novel situations is crucial. As the use of AI systems in open-world environments, like the real world, continues to expand, researchers are working towards incorporating these adaptive capabilities into AI agents as well.

Within the AI context, the term `open world' pertains to an environment in which an AI system confronts and engages with situations that may not have been explicitly covered during its training phase. Di ering from a `closed world' scenario, where an AI system functions within a con ned and predetermined scope, an open-world AI system is anticipated to navigate novel and unexpected situations by extrapolating its knowledge and adjusting its actions based on available information. This places a demand on the AI system to demonstrate resilience, adaptability, and the competence to make well-informed decisions even amidst unfamiliar and ever-changing conditions. Open-world learning (OWL) is a contemporary eld of study focused on creating such AI systems that can autonomously respond to unexpected alterations in their surroundings [Langley, 2020]. OWL researchers aim to empower agents capable of both detecting novelties in the environment and e ectively adapting to those novelties.

It is noteworthy that several research domains exhibit similarities with the principles underlying OWL. For instance, incremental learning [Ade and Deshmukh, 2013] involves updating the learning process as new examples are introduced, thereby re ning the acquired knowledge. Another related area is learning under concept drift [Lu et al., 2018], which entails developing methodologies to detect, comprehend, and adapt to unforeseen alterations in the distribution of streaming data over time. Furthermore, few-shot learning [Lu et al., 2018] focuses on swiftly generalizing to novel tasks with limited supervised information, while zero-shot learning deals with scenarios where the classes represented by training instances are disjoint from the classes the agent aims to classify. Additionally, transfer learning [Pan and Yang, 2010] endeavours to transfer knowledge from one domain to another, particularly when training data is available only in one domain, while the task pertains to another domain of interest.

Further insight into OWL emerges from cognitive science, which underscores the significance of common sense reasoning. This involves an intuitive understanding of physics, everyday world knowledge, and an inherent grasp of cause-and-e ect relationships [Stenning and Van Lambalgen, 2012]. Humans adeptly integrate these facets to comprehend their surroundings, predict outcomes, and adapt to new situations. In contrast, AI systems often lack comprehensive common-sense reasoning, hindering their ability to navigate complex environments e ectively [Marcus, 2020]. Machine common sense research endeavours to narrow this disparity by equipping AI with foundational knowledge about the world, akin to human intuition. Drawing from psychology and cognitive science, researchers strive to imbue AI models with robust common-sense reasoning abilities [Davis, 2023]. Integrating physics-based reasoning is a pivotal component of this eld [Ye et al., 2018; Agrawal et al., 2016; Mottaghi et al., 2016].

In the rest of this section, we explore the concept of novelty in the eld of OWL research and also investigate the environments that have been created to enable the experimentation of AI agents' abilities to detect and adapt to novel situations.

2.2.1 Concept of Novelty

In various AI studies, researchers have explored diverse approaches to formally conceptualize the concept of novelty. Certain scholars have referred to novel scenarios as anomalies or instances of out-of-distribution data [Boult et al., 2021; Feeney et al., 2022]. Other researchers de ne novelty in terms of situations that deviate from either implicit or explicit assumptions within an agent's model of the external world, which encompasses both other agents, the environment, and their interactions [Doctor et al., 2022; Kejriwal and Thomas, 2021]. An alternative perspective is o ered by Langley [2020], who characterizes novelty as transformations applied to the elements existing within the environment. These transformations encompass various forms, such as spatio-temporal, structural, process-oriented, and constraint-based transformations. In further support of this conceptualization, Molineaux and Dannenhauer [2022] formally establish a range of environmental transformations. Conversely, Boult et al. [2021] have introduced a comprehensive framework that uni es the notion of novelty within the domain of AI. Their approach revolves around the world space, observation space, and the agent's state to rigorously de ne various categories of novelties. In contrast to other research domains, such as incremental learning and learning under concept drift, as discussed in the preceding section, the novelties examined in our study encompass open-world elements characterized by structural and parametric alterations. In this context, agents must actively detect and swiftly adapt to sudden changes to e ectively operate in the environment.

The Defense Advanced Research Projects Agency (DARPA) has initiated the Science of Arti cial Intelligence and Learning for Open-world Novelty (SAIL-ON) program with the aim of investigating and developing the fundamental scienti c principles, general engineering methodologies, and algorithms necessary for the creation of adaptive AI systems that respond e ectively to novel situations [Russell, 2019; SAIL ON BBA, 2019]. As part of the SAIL-ON program, a working group known as the novelty working group (for a list of participants, refer to [Doctor et al., 2022]) has devised a novelty hierarchy. This hierarchy o ers a framework for categorizing di erent types of novelties based on their characteristics. This categorization proves invaluable for robust novelty assessment, enabling the design of novelties that span a broad spectrum and aiding in the identi cation of distinct novelty categories where an agent model might falter. In this thesis, during the investigations pertaining to open-world novelties, we employ this novelty hierarchy as a guiding framework. The hierarchy encompasses levels such as objects, agents, actions, relations, interactions, environments, goals, and events. Each level's de nition, as provided in [Doctor et al., 2022], is outlined below alongside an example involving a self-driving car.

1. Objects: New classes, attributes, or representations of non-volitional entities

- Encountering an unfamiliar road sign or a recently installed tra c signal that the self-driving car has not previously encountered.

2. Agents: New classes, attributes, or representations of volitional entities
   - Engaging with an autonomous drone that monitors tra c conditions and communicates information to the self-driving car's navigation system.
3. Actions: New classes, attributes, or representations of external agent behaviour
   - Observing a pedestrian using a unique hand signal to indicate intent to cross the road, necessitating the self-driving car to recognize and respond.
4. Interactions: New classes, attributes, or representations of dynamic properties of behaviours impacting multiple entities
   - Navigating through a construction zone with workers directing tra c using hand signals, demanding the self-driving car's understanding and appropriate response.
5. Relations: New classes, attributes, or representations of static properties of the relationships between multiple entities
   - Navigating a right-hand tra c area when the car is trained for left-hand side tra c.
6. Environments: New classes, attributes, or representations of elements independent of speci c entities
   - Driving through a region with unfamiliar ora and fauna potentially a ecting the self-driving car's sensors and decision-making.
7. Goals: New classes, attributes, or representations of external agent objectives
   - Interpreting the sudden acceleration of a nearby vehicle as a signal for a lane change, necessitating the self-driving car to adjust its trajectory accordingly.
8. Events: New classes, attributes, or representations of series of state changes
   - Unexpected road closures due to a local event or parade, prompting the self-driving car to reroute and adapt to changing tra c conditions.

2.2.2 Novelty Centric Environments

With the aim of advancing agents capable of functioning in open-world settings, researchers have created di erent environments for experimenting with and evaluating open-world AI systems. These environments revolve around introducing novelties into the tasks the agents have to perform, allowing for the assessment of agents' abilities to detect and adapt to such novelties.

(a) A diagrammatic representation of the NovelGridworlds domain with three example novelties. The default setting features the agent facing a crafting table and a surrounding wall encompassing the grid world arena.

(b) Example content from the NovelCraft dataset. The rst row displays a standard episode, while the following two rows depict two novel episodes. In these episodes, novel objects are highlighted in orange when they are visible.

Figure 2.7: Example novelties from the novelty-centric environments NovelGridworlds [Goel et al., 2021] and NovelCraft [Feeney et al., 2022].

GNOME [Kejriwal and Thomas, 2021] is a simulation platform designed to enhance the development and evaluation of AI systems within complex multi-agent scenarios like strategic board games. Speci cally, GNOME serves as a valuable resource for crafting, training, and assessing AI agents specialized in playing the classic strategic board game, Monopoly. The simulator introduces novelties across three distinct categories. First, it

enables the inclusion of class novelties, involving the introduction of new object types previously unseen in the game. For instance, a new card might be introduced that, when drawn, entails a substantial fee for each owned house and hotel. Second, attribute novelties are achievable, which entail altering the attributes of existing game elements. For example, adjustments could be made to the rent values of properties. Third, representation novelties are possible, leading to changes in the visual or structural representation of the game. An instance could be shu ing the order of slots on the board.

The NovGrid toolkit [Balloch et al., 2022] has been crafted to introduce novelty into the MiniGrid environment [Chevalier-Boisvert et al., 2018], thus enhancing its capabilities for novelty generation. This augmentation involves expanding the functionalities of existing elements within the MiniGrid environment, empowering agents to both detect and adapt to novel scenarios. Notably, the MiniGrid environment is designed as a grid-world platform that simpli es the integration of reinforcement learning algorithms into the environment, thereby promoting e cient iteration and testing of algorithms. NovGrid further extends this environment by enabling a more intricate interaction between the grid world and the agent, facilitating the seamless injection of novelties into the environment. The primary purpose behind the development of the NovGrid toolkit is to expedite the creation and assessment of reinforcement learning techniques that are equipped to adapt to novelties. The toolkit o ers a collection of 11 example novelties. These examples include modi cations such as relocating the goal object, altering the assumed locked or unlocked state of a door, adjusting the requisite number of keys for unlocking a door, and more.

The NovelGridworlds benchmark [Goel et al., 2021] has also been created as a tool to evaluate novelty detection and adaptation within a grid-based environment. Leveraging an OpenAI Gym environment [Brockman et al., 2016], this benchmark o ers a versatile platform for algorithm prototyping in both planning and machine learning contexts. The framework encompasses a set of 12 distinct novel scenarios, e ectively spanning three categories of novelties: objects, attributes, and actions. Under the objects category, novel elements are introduced into the environment, such as a new fence that restricts the agent's access to certain objects. Meanwhile, within the attributes category, modi- cations are made to the properties of existing items in the environment. For instance, the number of items the agent gains upon breaking an object might increase. Lastly, the actions category involves adjustments to the action space of the environment. As an example, a new action like `jump' could be introduced to the agent's available actions. This environment and sample novelties are shown in Figure 2.7a.

NovelCraft [Feeney et al., 2022] serves as a benchmark dataset speci cally designed for the purposes of novelty detection and discovery. This dataset operates within a modi ed version of the Minecraft game [Mojang Studios, 2023b], known as PolyCraft POGO [Goss et al., 2023]. PolyCraft POGO introduces a 3D environment where the agent's objective revolves around orchestrating a sequence of actions to transform available resources into a pogo stick. The NovelCraft dataset encompasses a collection of pre-rendered images and symbolic world-states, which o er insights into the perspective of an agent engaged

in the assembly task of creating a pogo stick. Examples of some novelties in NovelCraft include the introduction of new fences that obstruct access to trees and the introduction of novel tree species characterized by distinct bark properties, which in turn facilitates more e cient harvesting. These modi cations are visually identi able, enhancing the dataset's utility for evaluating the detection and discovery of novelties. Figure 2.7b displays representative images from the NovelCraft dataset.

The Smart Home GAN (SHGAN) system, introduced by Briscoe et al. [2022], is a tool aimed at producing smart home data with a particular emphasis on facilitating novelty detection. This system models the intricate dynamics inherent in a smart home environment. Through this modelling, SHGAN is capable of generating a multitude of lifelike sensor readings, which can be harnessed as an experimental platform for AI systems. The authors adopt a conditional Wasserstein Generative Adversarial Network (GAN) approach for the data generation process. Notably, this approach incorporates various essential factors, including the temporal sequence of sensor readings, human behaviours within the home, and the inclusion of corresponding activity labels. To validate their methodology, the authors draw upon data originating from real-world smart home contexts.

The Cartpole with novelty domain, as presented in [Holder, 2023], serves as a physics-based platform that places a signi cant emphasis on novelty. This domain is founded upon the well-known CartPole problem, also referred to as the inverted pendulum scenario [Wikipedia contributors, 2023b]. The core objective here revolves around the agent's task of maintaining the balance of a pole by strategically moving the associated cart in various directions: forwards, backwards, left, or right. The environment encompasses additional objects that exhibit movement and hold the potential for collisions with the cart, pole, walls, or one another. Within this setting, a range of novelties is embedded. These novelties encompass variations such as alterations in the cart's mass, shifts in gravity that impact both the cart and the pole, as well as modi cations to the dimensions of the objects within the environment.

Lastly, the Science Birds Novelty framework [Xue et al., 2022] stands as another noteworthy inclusion in the eld. Built upon the foundation of the previously discussed Angry Birds clone, Science Birds [Ferreira and Toledo, 2014b], this framework takes a central role in our novelty-based explorations within this thesis. It e ectively integrates novelties into the game levels of Science Birds and serves as the essential infrastructure for conducting experiments involving AI agents. It is noteworthy to highlight that the previously discussed AIBIRDS [AIBIRDS, 2023] competition has established a dedicated Novelty Track [AIBIRDS Novelty Track, 2023] that operates through this very framework. This competition track, which has been active since 2021, serves as a catalyst for advancing the development of AI agents capable of operating as pro ciently as humans in open-world physics-based settings. The 2022 iteration of the Novelty Track [AIBIRDS Novelty Track 2022, 2022] presented ten distinct novelties, each falling into ve distinct novelty categories:

- Class: Unseen objects or entities - 1) a new egg-shaped object sharing the colour of a pig, 2) a new object behaving like a pig that must be eliminated for victory.
- Attribute: Alterations in object features - 1) increasing the bounciness of the red bird, 2) elevating the slingshot's placement.
- Representations: Modi cations in object representations - 1) objects encircling pigs adopting the same colour as pigs, 2) transforming object shapes into circles instead of their true forms.
- Static Relations: Changes in static relationships - 1) relocating the slingshot to the right, 2) allowing wooden blocks to oat above ice blocks, causing wooden blocks to plummet when ice blocks are destroyed.
- Dynamic Interactions: Changes in dynamic interactions - 1) birds gliding through pigs without eliminating them, necessitating alternative methods to neutralize pigs, 2) enabling birds to slide along stone blocks due to reduced friction.

2.3 Procedural Content Generation

Procedural content generation (PCG) entails the algorithmic creation of content within virtual environments, particularly in the context of games. The scope of such content encompasses a wide array of elements, including game objects like characters and buildings, maps, road networks, music, puzzles, stories, game levels, and even complete games themselves. The ultimate aspiration in developing PCG algorithms often revolves around achieving content that rivals or surpasses human-designed creations in terms of quality and authenticity.

PCG has gained popularity among researchers and within the game industry for various reasons. One notable advantage is its ability to reduce the heavy reliance on human designers and artists, whose involvement can be time-consuming and expensive. As game development becomes more complex, requiring substantial human e ort, the potential to simplify this process through algorithmic generation becomes evident. Initially, PCG was valued for enabling game creation within memory limitations, but it also extends its utility by o ering a broader array of content than manual creation allows. Intelligent design tools with PCG capabilities could enhance the creativity of human creators, aiding small teams and enthusiasts in developing content-rich games without getting bogged down in details. Many PCG algorithms can swiftly produce new content that supports ongoing gameplay, ensuring a constant stream of new levels, characters, and areas, thereby fostering novel gaming experiences. Additionally, PCG's integration with player modelling holds the promise of player-adaptive games, enhancing player satisfaction, learning outcomes, and engagement.

Furthermore, PCG also contributes signi cantly to advancing AI research by providing a valuable content source for the development of learning algorithms. One challenge in AI research is the scarcity of datasets, a common issue. Gathering substantial amounts of

varied and pertinent data can be time-consuming and resource-intensive across di erent elds. Here, PCG o ers a scalable and e cient method to generate ample training data, ensuring that learning algorithms have access to numerous examples to re ne their capabilities. In the domain of AI, the quality and diversity of training data play crucial roles in in uencing the performance and adaptability of learning algorithms. PCG o ers an innovative approach to creating diverse datasets that can expose AI systems to a wide array of contexts and challenges.

2.3.1 Procedurally Generated Content and Techniques

PCG has been harnessed across a diverse spectrum of games to generate varied content utilizing various techniques. Hendrikx et al. [2013] present a classi cation of PCG-generated content, encompassing game bits, game space, game systems, game scenarios, game design, and derived content, while also outlining a taxonomy of common content generation methods, including pseudo-random number generators, generative grammars, image lltering, spatial algorithms, modelling and simulation of complex systems, and arti cial intelligence. Additionally, Togelius et al. [2011] categorizes PCG approaches into three categories: constructive, generate and test, and search-based.

The application spectrum of PCG extends to generating diverse elements within games. PCG has been harnessed for producing vegetation [Deussen et al., 1998; Dapper, 2023], terrains [Smelik et al., 2009; Olsen, 2004], caves [Lee et al., 2017], dungeons [Baldwin et al., 2017], road networks [Banf et al., 2010; Kelly and McCabe, 2007; Parish and Müller, 2001], buildings [Müller et al., 2006; Mirahmadi and Shami, 2012], creatures [Hudson, 2013], simulated motions [Karim et al., 2013; Horswill, 2009], non-player character interactions [Strong and Mateas, 2008; Gratch et al., 2002], quests [de Lima et al., 2019], stories [Riedl and Young, 2003; Mason et al., 2019; Jaschek et al., 2019], and levels [Gao et al., 2022]. Even beyond direct gameplay, PCG has extended to generating content such as music [Edwards, 2011; Boenn et al., 2011], leaderboards [Iosup et al., 2010], and in-game news and broadcasts [Rossignol, 2010], contributing to enhanced player immersion.

When examining games that employ PCG to create fundamental gameplay elements, noteworthy examples emerge. Elite [Braben and Bell, 2023], for instance, features eight 3D wireframe galaxies with each galaxy comprising 256 procedurally generated star systems. Microsoft Minesweeper [Microsoft, 2023] employs PCG to establish the grid layout of concealed mines. The game .kkrieger [Farbrausch, 2023] employs PCG not only for generating meshes and textures but also for music. Dwarf Fortress [Bay 12 Games, 2023] extensively employs PCG, encompassing narrative, history, creatures, and more, thereby shaping a substantial portion of its game content. In the case of Spelunky [Mossmouth LLC, 2023], tile-based levels are procedurally generated. The iconic Minecraft [Mojang Studios, 2023b] constructs the three-dimensional world composed of cube blocks through PCG techniques. RimWorld [Ludeon Studios, 2023] utilizes PCG to fashion narratives, events, and world map customizations. Banished [Shining Rock Software, 2023] generates terrain maps through PCG. No Man's Sky [Hello Games, 2023] pioneers procedurally

generated galaxies replete with planets and their diverse ecosystems. Minecraft Dungeons [Mojang Studios, 2023a] crafts dungeons imbued with monsters, traps, puzzles, and treasures through PCG. Valheim [Iron Gate Studio, 2023] engenders maps featuring an archipelago of islands set within an oceanic expanse. Meanwhile, Core Keeper [Pugstorm, 2023] presents an underground world shaped by procedural generation. For a comprehensive compilation of games utilizing PCG, refer to [Wikipedia contributors, 2023c].

One of the widely adopted approaches in PCG is the use of evolutionary algorithms, which mimic the process of natural selection to evolve content iteratively based on prede ned tness criteria. These algorithms have been employed to generate various game elements, including game maps [Ra e et al., 2014; Lara-Cabrera et al., 2014], platform game levels [Thakkar et al., 2019], dungeon-type game levels [Font et al., 2016], game character mechanics [Skj rseth and Vinje, 2020; Pereira et al., 2021], and racing tracks [Loiacono et al., 2011], leading to the creation of dynamic and engaging gameplay experiences. Coevolutionary techniques, on the other hand, involve the simultaneous evolution of multiple components of a game system, such as AI players and level generators, where the level generators are evaluated by adapting AI players, and vice versa. This approach yields an AI player trained on gradually more di cult levels and a sequence of level generators with gradually increasing di culty [Nogueira-Collazo et al., 2015; Flimmel et al., 2021], fostering emergent complexity and balance within the game environment.

In recent years, adversarial design has signi cantly impacted PCG by introducing a competitive element to the generation process. Adversarial design pits two AI systems against each other: a generator tasked with generating content and an evaluator responsible for assessing its quality. This adversarial dynamic encourages the generator to produce content that challenges the evaluator's ability to discern between generated and human-created content, ultimately leading to the generation of high-quality game assets [Mittalella et al., 2021]. Moreover, signi cant advancements are occurring in the eld of PCG with the emergence of foundation models, such as GPT (Generative Pretrained Transformer) architectures. One such recent example is MarioGPT [Sudhakaran et al., 2024], a ne-tuned GPT2 model trained to generate tile-based game levels. MarioGPT enables controllable level generation through text prompts, leveraging large-scale pretraining on diverse datasets to learn rich representations of game content. These advancements enable foundation models to generate coherent and contextually relevant game elements across various genres and styles.

2.3.2 PCG in Physics-Based Environments

Creating content through procedural generation is a complex endeavour involving several challenges. These include achieving satisfactory generation speed, which varies depending on whether content is generated during gameplay or game development. Ensuring content reliability entails satisfying prede ned quality criteria; for instance, a dungeon must possess both entry and exit points. It is also important for the content to be diverse in order to prevent it from becoming monotonous. Additionally, the

content should appear inventive rather than being obviously machine-generated. When applied to physics-based settings, PCG faces additional complexities due to the in u-ence of physics. In such environments, devising content necessitates algorithms that seamlessly merge physics simulations with essential design principles. These algorithms must acknowledge the physical limitations of the environment, generating content that aligns with these constraints for practical feasibility. Furthermore, precise determination of game object positions is imperative to achieve intended outcomes during gameplay. Many of the challenges encountered in generating content for physics-based environments are relevant to the challenges faced in real-world physics scenarios [Stephenson, 2019].

Physics-based puzzle games have gained substantial attention within the PCG research due to the intriguing physics-related challenges they present when generating content, as previously discussed. Researchers have employed well-known physics puzzle games like Angry Birds [Rovio, 2023] and Cut the Rope [ZeptoLab, 2023] as experimental platforms to explore techniques for generating physics-based content. For instance, in the case of Cut the Rope, investigations have revolved around the development of an authoring tool for crafting and evaluating levels. This tool seamlessly merges automated content design with the creative input of the designer [Shaker et al., 2013c]. Furthermore, the application of evolutionary computation, particularly grammatical evolution, has been examined for level creation [Shaker et al., 2013a]. Additionally, researchers have adopted simulation-based strategies to assess playability [Shaker et al., 2013b], along with a progressive method for generating levels that synergistically combines features of search-based and constructive approaches [Shaker et al., 2015].

The researchers have devoted substantial attention to the exploration of level generation in Angry Birds [Stephenson et al., 2019]. This e ort has been strengthened by the Angry Birds level generation competition, typically conducted at the IEEE Conference on Games [Renz et al., 2021]. This competition serves as an incentive for the development of level generators designed for the game. In this research landscape, various approaches have been pursued to address the multifaceted challenges of level generation in Angry Birds. Noteworthy methodologies encompass employing genetic algorithms for level generation [Ferreira and Toledo, 2014a,c; Kaidan et al., 2016; Ferreira and Toledo, 2017; Calle et al., 2019b,a; Salinas-Hernandez and Garcia-Valdez, 2020], leveraging machine learning techniques to optimize the runtime of genetic algorithm-based level generation [Pereira et al., 2016], devising strategies for generating stable structures within the game's physical environment [Stephenson and Renz, 2016a,b], employing Monte Carlo Tree Search for level creation [Graves, 2016], implementing constructive grammar for generating architectural-style structures [Jiang et al., 2016], achieving variation, stability, and solvability in generated levels [Stephenson and Renz, 2017] (Figure 2.8 illustrates example levels generated by this generator), generating fun levels through pattern-struct and preset-model approaches [Jiang et al., 2017], introducing a mixed-initiative system for swift level prototyping [Campos et al., 2017], generating engaging physical structures using image silhouettes [Zhou et al., 2018], dynamically adapting game level di culty via agent-based adaptive generation [Stephenson and Renz, 2019], generating levels in-

Figure 2.8: Example Angry Birds game levels generated from the level generator presented by Stephenson and Renz [2017].

spired by Rube Goldberg Machine mechanisms [Abdullah et al., 2019, 2020], employing deep generative models with sequential encoding and latent variable evolution for level generation [Tanabe et al., 2021], creating block structures based on hand-drawn sketches [Stephenson et al., 2021], generating physical structures for the game using generative adversarial networks [Abraham, 2022] and utilizing prompt engineering for generating levels [Taveekitworachai et al., 2023].

PCG also nds application within certain physical reasoning environments that are discussed in Section 2.1.1. The designing and creation of tasks within these environments predominantly rely on task templates developed by developers, complemented by automated generation techniques that produce modest variations of those handcrafted task templates. For instance, PHYRE employs the strategy of task generation by manipulating object positions within 25 handcrafted task templates. Similarly, Virtual Tools employs a comparable approach, generating random tasks for reinforcement learning agents by adjusting object dimensions or positions within prede ned templates while maintaining the original task solution. OGRE builds upon PHYRE and Virtual Tools, adopting a similar generative process involving modi cations to the shape, size, and location of the baseline template. In the CausalWorld benchmark, the agent's task involves rearranging blocks to attain a speci ed goal con guration of blocks. This benchmark encompasses eight distinct task types, each equipped with a dedicated generator that derives tasks by sampling new goal shapes from a prede ned goal shape template.

Chapter 3

Deceptive Level Generation for Angry Birds

The AIBIRDS competition has spanned several years, aimed at fostering the advancement of AI agents capable of outperforming human players in Angry Birds game levels. Throughout the competition's history, diverse agents with varying approaches have been employed to tackle this challenge. Despite considerable progress achieved by these agents in recent years, they still face signi cant limitations when confronted with deceptive levels. This arises from the agents' struggles in grasping the intricacies of deceptive setups within the physics-based environments, as well as their inclination to focus on selecting optimal individual shots rather than devising e cient shot sequences. In order to facilitate progress in addressing these shortcomings, we introduce an automated methodology for crafting deceptive game levels in Angry Birds, thereby equipping agents with ample experimental data for enhancement. While numerous content generators for Angry Birds exist, they largely overlook the generation of deceptive levels. Within this chapter, we present a procedural framework for generating deceptive levels across six distinct deception categories aimed at confounding state-of-the-art Angry Birds playing AI agents. Our evaluations illustrate that the generated deceptive levels exhibit qualities akin to those crafted by human creators. Additionally, we establish metrics to assess the stability, solvability, and degree of deception inherent in the generated levels.

3.1 Introduction

Procedural Content Generation (PCG) is a key area of investigation that focuses on the algorithmic generation of game content in video game research, as we discussed in Section 2.3. Game-playing agent development has bene ted from PCG since PCG can be used to generate a large amount of training data in a short period of time [Togelius, 2016]. Most of the current learning-based approaches embedded in AI agents require a

huge amount of training data to be able to perform well [Zhu et al., 2016].

Video games are frequently used by AI researchers as testbeds for their research [Justesen et al., 2019] and Angry Birds is one such example that provides interesting challenges for AI agents when solving the game levels. PCG in Angry Birds is non-trivial due to the physics constraints in the game. Any PCG algorithm should adhere to the physical constraints of the game, and the positions of the game objects should be determined with greater precision to ensure the expected outcomes can be obtained when playing. Similar to an agent in a physical environment, Angry Birds playing agents have a continuous action space. Hence, generating levels that can only be solved by intended actions is challenging, especially when the levels are complex and require appropriate reasoning capabilities to solve.

A deception for an AI agent can be seen as a characteristic of a task that `tricks' the agent into making poor actions by exploiting its biases or limitations [Anderson et al., 2018]. A deceptive game level has a reward structure that can lead the agent away from the optimal strategy [Bontrager et al., 2019]. Previous work on Angry Birds presented six categories of deceptions for the existing Angry Birds playing agents [Stephenson and Renz, 2018]. Using handcrafted levels for those deception categories, they showed the vulnerabilities of the state-of-the-art Angry Birds playing agents. The drawbacks of existing agents in handling deceptions show that there is room for further advancements of agent capabilities. This urges the necessity of creating a su cient number of challenging deceptive levels to satisfy the data requirement of current learning approaches to facilitate research in developing advanced AI agents that can handle deceptions.

In this chapter, we present a methodology to generate deceptive levels in Angry Birds. We consider the six deception categories presented by Stephenson and Renz [2018] and de ne level templates that enable the automatic generation of varied levels for each of these deception categories. In addition to the level itself, a salient feature of this approach is that the solution for the level is also generated and can be utilized by the agents in the learning process. To evaluate our methodology, we de ne metrics to measure the stability, solvability, and deceptiveness of the generated levels. Moreover, to measure the characteristics of the generated levels compared to human-created levels, we evaluate AI agents' behaviour on both generated levels and handcrafted levels that require similar capabilities to solve.

3.2 Background and Related Work

3.2.1 PCG in Angry Birds

Researchers have previously used several diverse approaches for procedurally generating game levels for Angry Birds, as discussed in Section 2.3.2. This prior work mainly focuses on preserving the stability of the physical structures in the levels, adjusting the di culty and enjoyability of the levels, and ensuring the levels are solvable. None of them focuses on generating deceptive game levels that require challenging reasoning and

planning capabilities to solve. In this work, we focus on generating game levels that AI agents are not capable of solving through simple, intuitive approaches. Another signi cant di erence from the existing work is our approach generates the solutions for the levels, which can be bene cial for learning AI agents in their training.

3.2.2 Deceptive Games

The concept of deceptive games was rst presented in the study by [Anderson et al., 2018]. They create a suite of deceptive arcade-style games for the General Video Game AI (GVGAI) framework [Perez-Liebana et al., 2016] and explore the e ect of those games on game-playing agents. Building on that idea, an approach to generate deceptive levels for games in the GVGAI platform is discussed in [Zafar et al., 2018] and a methodology to evaluate agents on deceptive levels is presented in [Sabah and Zafar, 2021].

In the context of complex physics simulation games, as already mentioned above, the work on deceptive Angry Birds levels by Stephenson and Renz [2018] suggests six di er-ent categories of deceptions that can trick or exploit the current state-of-the-art Angry Birds playing agents. While these deceptions do not a ect all Angry Birds agents equally, no agent was able to successfully handle all deception categories. We consider these same six deception categories in our generation process. The six deception categories are de-scribed below and examples for those categories are shown in Fig. 3.1.

- Rolling/falling objects : This deception uses the fact that objects of one en-tity can fall or roll onto another entity to create an impact. An agent needs to understand that an object can fall or roll and that object can be used to hit targets.
- Clearing paths : This deception occurs when there are obstacles that need to be removed or destroyed before a target can be reached. To deal with this deception, an agent needs to understand that in order to reach a target, it should rst clear the path to the target.
- Entity strength analysis : This deception requires an agent to analyse the strengths of the entities in a level. The strength of an entity depends on the factors such as its material, shape, and size. An agent should be capable of determining the physical weaknesses/strengths of individual entities and interact accordingly to solve the level.
- Non-greedy actions : In this deception, an action that appears to be less e ective in the short term compared to another possible action, will gain higher advantages in the long term. The agent should look ahead and plan the actions using its knowledge of the environment rather than performing a greedy action that gives a higher short term reward.
- Non- xed tap time : The special powers of the birds are activated by tapping while the bird is in ight. In this deception, an agent needs to activate the special powers of the birds at non- xed times, opposed to tapping at a pre-determined xed time.

(a) Rolling/falling objects

(b) Clearing paths

(c) Entity strength analysis

(d) Non-greedy actions

(e) Non- xed tap time

(f) TNT

Figure 3.1: Six example handcrafted levels for each deception category presented in [Stephenson and Renz, 2018]. The solutions for the levels are, (a) target the structure which collapses and falls on top of the pig, (b) use the rst two birds to clear the path for the third, (c) use blue and yellow birds respectively on ice and wood structures, (d) rst make the non-greedy shot that kills fewer pigs with black bird, (e) tap the yellow bird (accelerate) before hitting the block, and (f) target pig directly and ignore TNTs.

ˆ TNT : The TNT explosives in the game explode when hit, causing damage or pushing nearby objects. In this deception, an agent needs to use TNTs to kill pigs or TNTs are used to distract the agent from the objective of killing pigs.

3.3 Terminology

In this chapter, we use the term *environment* to refer to the space where an agent can sense and act, which is not a part of the agent. A stand-alone, self-contained set of physically stable objects in the environment is termed as a *physical entity*. A *deception* is considered as a characteristic of a task that exploits the cognitive biases of an agent and causes it to make sub-optimal decisions [Anderson et al., 2018]. A *tactic* is a sequence of an agent's actions (i.e., bird release points and tap times) that involves interacting

Figure 3.2: Deceptive level generation methodology.

with the environment. A solution tactic is a speci c tactic that solves a given level when executed. Finally, a strategy is a plan that an agent tries to solve a level (e.g. shooting birds targeting at pigs). For a strategy, there can be multiple tactics that may or may not solve the level.

3.4 Proposed Procedure

In this section, we present the proposed deceptive level generation procedure for Angry Birds. Fig. 3.2 shows the main components of the generation procedure. The four components shown in blue are the modules in the procedure and the four components in red are the inputs/outputs of those modules. The generation procedure does not involve creating physical entities; instead, it extracts physical entities created from existing content generators. This provides access to a wide range of content from various generation methods. Similarly to using existing content generators, customized handcrafted content can be used as well. The generation procedure consists of two stages: an o ine stage and an online stage. The o ine stage is shown inside the grey box (rst row) and the online stage is shown inside the purple box (second row). The o ine stage needs to be executed only once for an extracted set of physical entities. When a level generation request is received, the online stage is executed to generate a level using the o ine stage's stored data. In subsequent subsections, we discuss the four components in our generation procedure: Physical Entity Extraction Module, Physical Entity Analyzer Module, Level Template Matching Module, and Level Validation Module.

3.4.1 Physical Entity Extraction Module

This module takes already generated game instances from Angry Birds content generators and extracts physical entities from those instances. The game instances can be complete game levels or parts of game levels with physical entities. We use a collection of existing Angry Birds level generators [Stephenson et al., 2019] to generate a set of game instances. Entities are copied from these instances using a qualitative reasoning process that iteratively extracts entities. Algorithm 1 shows the physical entity extraction process that can be applied to a game instance with multiple physical entities. When formulating the algorithm, we de ne the term supporter using the support graph similar to how authors have de ned it in [Stephenson et al., 2021]. If the bottom horizontal edge

Algorithm 1 Extract physical entities

```
Input  : Game instance with physical entities
Output  : Set of extracted physical entities
 1: entitiesExtracted =  fg
 2: objectsRemaining = all objects in the instance
 3: while  objectsRemaining is not empty  do
 4:     topObject = topmost object of objectsRemaining
 5:     entityExtracting = topObject and supporters of topObject
 6:     while  True  do
 7:         boundingBox = bounding box of entityExtracting
 8:         entityExtracting = objects inside boundingBox
 9:         if  boundingBox size not increased then
10:             Break
11:     add entityExtracting to entitiesExtracted
12:     remove objects of entityExtracting from objectsRemaining
13: return  entitiesExtracted
```

Figure 3.3: An illustration of the extraction process for a single entity. The leftmost gure (start) shows the entity that needs to be extracted. Each step corresponds to an iteration of the algorithm, and the rightmost gure (end) shows the extracted entity. Selected objects in each step are shown in green, and the bounding box of selected objects is shown in red dotted lines.

of an object O_i is in contact with the top horizontal edge of another object O_j (i.e., O_i is resting on top of O_j), then the support graph contains an edge pointing from object O_i to O_j. In a given support graph, if there exists a path from object O_i to object O_j, then object O_j is considered as a supporter of the object O_i (O_j supports O_i). Objects placed on the ground do not have any supporter and hence has an empty support graph. Fig. 3.3 shows an example of how a physical entity is extracted iteratively using this algorithm.

3.4.2 Physical Entity Analyzer Module

After extracting physical entities, the Physical Entity Analyzer Module is used to analyze those entities individually by interacting with them and observing the outcome. In Angry Birds, interactions with the entities can only be done by shooting birds from the slingshot and tapping to activate the bird's special power. To predict the result of physical interactions, we can use either qualitative methods [Walega et al., 2016; Zhang and Renz, 2014], which are typically faster but less accurate, or simulation-based

methods [Battaglia et al., 2013], which are typically more accurate but slower. Using qualitative methods in Angry Birds for physical predictions has proven to be less accurate and robust for large complex entities [Stephenson and Renz, 2017]. Therefore, we use a simulation-based method to perform interactions and record the outcomes.

We developed a portfolio agent who can play Angry Birds with four di erent strategies adopted from previous Angry Birds playing AI agents discussed in Section 2.1.3. They are, 1) shooting birds (without special powers) targeting pigs in the entity, 2) shooting birds (without special powers) targeting at TNTs in the entity, 3) shooting birds (without special powers) targeting reachable blocks in the entity, and 4) shooting birds with special powers (activated at di erent times) targeting at pigs. The above strategies also have di erent variants (i.e., tactics) based on the order of the targets (if multiple targets exist), the shooting angle, and the activation time of the bird's special power.

Each entity is tested with a predetermined xed set of tactics. When interacting with the entities, the objects in the entity can be subjected to move, collide, damage, or destroy due to the force applied to the entity. For each interaction (i.e., a bird shot), data is recorded in four steps:

- Entity data before the interaction (e.g. the number of objects in the entity, the entity's bounding box size).
- Interaction data (e.g. the magnitude and the location of the force applied on the entity due to the interaction).
- Dynamic data when the objects in the entity are moving as a result of the interaction (e.g. objects that have gone out of the entity's original bounding box, location and the velocity of the objects gone out).
- Entity data after the interaction when all the objects become stationary (e.g. the number of objects in the entity, the entity's bounding box size).

The data gathered through this analysis is attached to the entity as metadata, and a repository of entities with metadata is maintained. This is the end of the o ine stage of the generation process; the next steps are done online.

3.4.3 Level Template Matching Module

The online stage of the physical deceptive level generation process starts with a generation request coming to the Level Template Matching Module, specifying the desired deception category. This module generates a candidate level for that deception category using pre-de ned level templates. For a deception, a level template contains constraints that entities should satisfy and rules for level generation. When generating a level, the level template considers possible interactions that an agent can perform and the outcomes of those interactions obtained from the entities' metadata. The entities that satisfy the constraints in the template are used to generate the level, following the generation rules.

3 Deceptive Level Generation for Angry Birds

When generating levels,Level Template Matching Modulealso creates the solution for the level using the interaction data (bird shots) in the metadata of the entities used to generate the level. The solution has the interaction data that solves the level (i.e., the solution tactic of the level). When designing the level templates, we attempt to make the deceptive level solvable only by using the generated tactic. We assume if an agent uses the solution tactic to solve the level, the agent understands the deception. Restricting the solvability only to the generated solution prevents the agents from solving the levels using other strategies without realizing the deception.

To facilitate the template design discussion, on top of the terminology discussed in Section 3.3, we de ne the following terms. Thegoal of an Angry Birds player is, killing all the pigs in the level using the given number of birds. We refer to the termsolving an entity as killing all the pigs within an entity. An entity is deemed solvable if all the pigs within the entity can be killed using the given birds. An outperforming solution tactic (OST) refers to the solution tactic that uses the minimum number of birds to solve an entity compared to other solution tactics tested. If there are multiple solution tactics with the fewest number of birds needed to solve an entity, then there is no OST for that entity. The level templates designed for the six deception categories are explained below.

Rolling/Falling Objects

A level with this deception is created using two types of entities, called senders and receivers. A sender gives an object out either by rolling or falling when an agent interacts with the entity. A receiver uses the impact of the sender's rolling/falling object towards achieving the goal (i.e., killing the pigs). The locations of the two entities are determined such that the rolling/falling object's impact can be used to replace the impact of an action in the receiver's OST. The placement of the two entities should also ensure that the receiver's OST no longer allows the agent to solve the level. This is done by placing the sender such that it blocks the bird trajectories of the actions in the receiver's OST. The reachability of the target objects for the birds shot from the slingshot is veri ed to ensure the solvability of the level. Finally, considering the interactions needed for the sender and the receiver to solve the level, the number and types of birds are determined and the solution for the level is generated. Algorithm 2 shows the pseudocode of this level template.

Clearing Paths

A level with this deception is created using two types of entities, called obstacle and obstructed. The obstacle blocks the OST of the obstructed entity. The agent should rst substantially collapse the obstacle to clear a path to reach the obstructed entity. To ensure the agent interacts with the obstacle with the objective of collapsing to clear the path, entities that can only be collapsed with a speci c tactic are selected as obstacles (i.e., the agent needs to execute a speci c tactic by analyzing the obstacle to collapse it substantially). Considering the interactions needed to clear the obstacle and execute

Algorithm 2 Rolling/falling objects

```
Input  : Sender entity, Receiver entity
Output  : A level with rolling/falling objects deception
if sender has suitable objects that can be rolled/fallen then
    if receiver has an OST then
        while generation unsuccessful do
            get sender's rolling/falling object's trajectory
            get OST of receiver
            generate level by matching sender and receiver
            verify the reachability of targets
            if generation successful then
                break
            if maximum generation attempts reached then
                return none
        allocate birds to level
        generate solution tactic
        return generated level
return none
```

the OST of the obstructed entity, the birds are allocated to the level and the solution is generated. The pseudocode for this level template is given in Algorithm 3.

Algorithm 3 Clearing paths

```
Input  : Obstacle Entity, Obstructed Entity
Output  : A level with clearing paths deception
if obstacle entity can be cleared substantially then
    if obstructed entity has an OST then
        while generation unsuccessful do
            generate level by matching obstacle and obstructed entities
            verify the reachability of targets
            if generation successful then
                break
            if maximum generation attempts reached then
                return none
        allocate birds to level
        generate solution tactic
        return generated level
return none
```

Entity Strength Analysis

The strength of an entity is considered with respect to its capability of protecting pigs within it. In this level template, di erent bird types are used, as this a ects the maximum damage that can be done to an entity. An agent needs to analyze the strength of the entities in the level and determine the correct bird type to use on each entity. Algorithm 4 shows the pseudocode of this level template. The number of birds that needs to solve an entity is termed as the bird usage in the pseudocode. The template creates levels by selecting two entities: one entity that is strong against one bird type (i.e., high bird

usage) but weak against another (i.e., low bird usage) and another entity that shows opposite strengths and weaknesses to the same bird types. The solution of the level is generated considering the interactions needed to solve the two entities using the selected bird types.

Algorithm 4 Entity strength analysis

```
Input : Entity 1, Entity 2
Output : A level with entity strength analysis deception
for birdX in all bird types do
    for birdY in all bird types except BirdX do
        if entity1 solvable by birdX and entity2 solvable by birdY then
            if entity1 birdX usage < entity2 birdX usage and entity2 birdY usage < entity1 birdY
            usage then
                generate level with entity1, entity2, birdX, birdY
                break
        else if entity1 solvable by birdY and entity2 solvable by birdX then
            if entity1 birdY usage < entity2 birdY usage and entity2 birdX usage < entity1 birdX
            usage then
                generate level with entity1, entity2, birdX, birdY
                break
if generation successful then
    verify the reachability of targets
    generate solution tactic
    return generated level
return none
```

Non-greedy Actions

In Angry Birds, greedy agents tend to kill the most pigs in a single action. In this level template, an entity that has more pigs and is easy to solve (termed a greedy entity) is combined with an entity that has a few pigs and is hard to solve (termed a non-greedy entity). The easiness/hardness of an entity is based on the type and the number of birds needed to solve it. To ensure the level can only be solved by doing the non-greedy action rst, the two selected entities should have OSTs with di erent bird types. The order of birds in the level is selected such that the birds needed to solve the non-greedy entity are given rst. As the agent cannot change the order of the birds, if it chooses to solve the greedy entity rst, then it cannot solve the non-greedy entity. The solution of the level is generated considering the OSTs of the two entities. Algorithm 5 shows the pseudocode of this template.

Non- xed Tap Times

The e ect of a bird's special power on an entity depends on the time that an agent taps the bird during its ight. A bird can cause more damage to the entity if it is tapped at the correct time. The correct tap time needs to be determined considering the distance to the target and the bird's special power. Entities that can only be solved by using a speci c tap time of a bird are selected as feasible entities to generate levels with this

Algorithm 5 Non-greedy actions

```
Input  : Greedy Entity, Non-greedy Entity
Output  : A level with non-greedy actions deception
if  2 entities' 2 OSTs use di erent birds  then
    if  greedy entity is easier to solve than non-greedy entity  then
        while  generation unsuccessfuldo
            generate level with the greedy entity and non-greedy entity
            allocate birds in the order for non-greedy and greedy entities
            verify the reachability of targets
            if  generation successfulthen
                break
            if  maximum generation attempts reached  then
                return  none
        generate solution tactic
        return  generated level
return  none
```

deception. Two entities with two di erent tap times are used to generate a level, to ensure that agents with xed tap times fail to solve the level. The solution is generated taking into account the interactions needed to solve both entities in the level. Algorithm 6 shows the pseudocode of this level template.

Algorithm 6 Non- xed tap time

```
Input  : Entity 1, Entity 2
Output  : A level with non- xed tap time deception
for  birdX in all bird types with special powers  do
    if  entity1 only solvable by birdX's speci c tap time (T x) then
        for  birdY in all bird types with special powers  do
            if  entity2 only solvable by birdY's speci c tap time (T y) then
                if  Tx 6= Ty then
                    generate level with entity1, entity2, birdX, birdY
                    break
if  generation successfulthen
    verify the reachability of targets
    generate solution tactic
    return  generated level
return  none
```

TNT

TNT deception is generated using two level templates. Using the rst template, if there is an OST for an entity and that tactic involves targeting TNTs, then a level is generated directly with this entity and the birds needed for its OST. This form of the deception needs an agent to understand that TNTs explode when hit to cause greater damage, which can be used to achieve the goal. In the second template, TNTs are used to distract an agent. This template selects an entity with TNTs and without pigs (termed as a distracting entity) along with another entity with pigs (termed as a

distracted entity). The birds allocated to the level are the birds needed for the OST of the distracted entity. Shooting a bird to explode the TNTs will not help solve the level and will not leave enough birds to solve the distracted entity. The pseudocode for the second template is shown in Algorithm 7. The solution for the level is the OST of the distracted entity.

Algorithm 7 TNT

Input : Distracting Entity, Distracted Entity
Output : A level with TNT deception

```
if distracting entity has TNTs and no pigs then
    if distracted entity has an OST then
        generate level with distracting entity and distracted entity
        allocate birds needed for the distracted entity's OST
        verify the reachability of targets
        generate solution tactic
        return generated level
return none
```

Once a deceptive level and its solution have been generated using one of these templates, they are passed to the Level Validation Module.

3.4.4 Level Validation Module

The last module in the generation process validates the physical stability and solvability of the generated levels. In Angry Birds, all the objects should remain stationary at the start of the level. The Level Validation Module first verifies the stability of the level using the Box2D physics engine. The velocities of objects in the level are observed after two seconds of simulating the level to determine the stability of the level. After confirming stability, the Level Validation Module then verifies the solvability of the level, using its generated solution tactic. The final outcome of this process is a stable deceptive level, along with its confirmed solution.

3.5 Results and Evaluations

In this section, we present the results and evaluation of the proposed deceptive level generation procedure. The majority of our implementation was coded using Python 3.7. The exceptions were the Physical Entity Analyzer Module and the Level Validation Module that required simulating the Box2D physics engine in Unity, coded in C#. The simulations were done by speeding up the physics engine by 50 times. The software was run on a Windows 10 desktop PC with an i9-9900KS CPU and 64GB RAM. The average time (for 100 runs) consumed by each module is Physical Entity Extraction Module took 21 milliseconds to extract an entity with 16 blocks, Physical Entity Analyzer Module took 85.71 seconds to analyze an entity using ten tactics, Level Template Matching Module took 2.64 seconds to generate a level, and Level Validation Module took 9.01 seconds to validate a level. Therefore, the online stage of the generation process can generate a

(a) Rolling/Falling Objects (b) Rolling/Falling Objects (c) Clearing Paths

(d) Clearing Paths (e) Entity Strength Analysis (f) Entity Strength Analysis

(g) Non-Greedy Actions (h) Non-Greedy Actions (i) Non- xed Tap Time

(j) Non- xed Tap Time (k) TNT (Solution) (l) TNT (Distraction)

Figure 3.4: Examples of generated levels for the six deception categories. The solution tactic for each level is illustrated using the red dotted arrows. The arrow path shows the trajectory of the bird, and the arrowhead is pointed to the target. Numbers show the order of bird shots. `Tap at x%' means the bird needs to be tapped at the length of x% in its trajectory path, which is only available for birds with special powers. The blue lines in (a) and (b) are the trajectories of falling objects.

Table 3.1: Stability rate (R_i), solvability rate (S_i), the average deceptive score (D_i), and the average difference of the solve rates for human-created and generated levels (C_i) for the six deception categories.

Deception Category	R_i	S_i	D_i	C_i (%)
Rolling/falling Objects	1.00	0.82	0.91	3.41
Clearing Paths	0.98	0.95	0.90	-1.75
Entity Strength Analysis	0.98	0.90	0.89	-6.67
Non-greedy Actions	1.00	0.81	0.95	7.56
Non-fixed Tap Time	0.97	0.87	0.97	0.00
TNT	0.96	0.92	0.95	-9.26

deceptive level in 11.65 seconds on average. Twelve levels generated for the six deception categories are shown in Fig. 3.4.

In the following sections, we evaluate our generated levels using four metrics that measure stability, solvability, deceptiveness, and similarity to human-created levels. Even though both stability and solvability are normally verified by the Level Validation Module, this module was disabled for these experiments in order to assess how often a generated level passed each of these validity checks.

3.5.1 Stability

The stability rate (R_i) of a deception category i is calculated as the percentage of levels that were stationary within the Box2D physics engine at the start of the game levels. A set of 100 levels for each deception category was used (600 levels in total). The results are presented in the second column of Table 3.1. This shows that our generation procedure is capable of generating levels with a stability rate of almost 100%.

3.5.2 Solvability

The physical outcomes of the interactions considered in the generation process might differ from the outcomes when actually playing the level, typically due to differences in the location of entities when generating levels versus their original location during entity analysis, as discussed in Section 3.4.2. Therefore, a generated level might not always be solvable using the intended solution tactic (i.e., the generated solution). Hence, we evaluate the solvability of the generated levels to measure the competence of the generator in creating levels that can be solved with the intended solution tactic.

The solvability rate (S_i) of a deception category i is calculated as the percentage of levels which could be solved using its provided solution tactic. A set of 50 levels for each deception category was used (300 levels in total). The results are presented in the third column of Table 3.1. This shows that our generation procedure is capable of generating levels with a solvability rate higher than 81% for the six deceptions.

3.5.3 Deceptiveness

To measure the degree of deception of the levels in a deception category, we de ne a metric, the average deceptive score. A level is considered less deceptive if it can be solved by multiple tactics without understanding the deception. The solution tactic created when generating a level is the tactic that an agent would follow if the deception is understood. To calculate the deceptive score of a level, the level is tested with di erent known tactics, and the number of tactics that solved the level, excluding the solution tactic, is used. Equation 3.1 shows the average deceptive score D_i() calculated for a deception category i by averaging the deceptive scores of all the levels tested for the deception. N is the total number of levels tested, T_n is the total number of tactics used to test the nth level, and P_{nt} is 1 if the nth level is solved by t^{th} tactic or 0 otherwise. D_i 2 [0; 1] and a higher score means a higher deceptiveness.

$$D_i = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{T_n} \sum_{t=1}^{T_n} (1 \quad P_{nt}) \tag{3.1}$$

For this experiment, we developed and ran a portfolio agent with ten variants of strategies mentioned in Section 3.4.2. A set of 50 levels for each deception category was used (300 levels in total). The fourth column of Table 3.1 shows the average deceptive score calculated for the six deceptions. The results show that our method can generate deceptive levels with an average deceptive score over 0.89 for the six deceptions.

3.5.4 Comparison with Human-Created Levels

From human capabilities, we are very adept at creating deceptive levels. Therefore, we evaluate generated levels against human-created levels by examining whether the generated levels exhibit similar characteristics to the human-created levels for AI agents. We compare the solve rates of agents for human-created and generated levels that belong to the same deception category. The average level solve rate di erence between the human-created and generated levels (C_i) for a deception category i is shown in Equation 3.2. A is the number of agents tested, M is the total number of human-created levels, and N is the total number of generated levels. P_{am} is 1 if the ath agent solved m^{th} human-created level or 0 otherwise, similarly P_{an} is 1 if the ath agent solved n^{th} generated level or 0 otherwise. A positive value for C_i indicates that the generated levels are more di cult to solve than the human-created levels on average for AI agents and vice versa.

$$C_i = \frac{1}{A} \sum_{a=1}^{A} \left(\frac{1}{M} \sum_{m=1}^{M} P_{am} \quad \frac{1}{N} \sum_{n=1}^{N} P_{an} \right) \tag{3.2}$$

For this evaluation, we used 30 handcrafted Angry Birds levels from the previous work [Stephenson and Renz, 2018] representing the six deception categories (ve levels on

Figure 3.5: Level solve rates of three agents for human-created and generated levels. The deception indexes on the x-axis are in the order: rolling/falling objects (1), clearing paths (2), entity strength analysis (3), non-greedy actions (4), non-xed tap time (5), and TNT (6).

average per category). We generated 300 levels from our generator (50 levels per category). Three state-of-the-art Angry Birds agents, Datalab, Eagle's Wing, and Bambirds discussed in Section 2.1.3, were used for the experiment. The solve rates of the three agents for the six deceptions for human-created and generated levels are shown in Fig. 3.5. This gure depicts that the solve rates of the generated levels are correlated to the solve rates of the human-created levels. This portrays that agents show similar behaviours when playing both human-created and generated levels. The ϵ_i values (percentage) calculated for the six deceptions from the results of the three agents are in the fth column of Table 3.1. The levels generated for two deception categories were more di cult to solve than the human-created levels, while three deception categories were easier. The human-created and generated levels of non- xed tap time deception were equally di cult for the agents.

3.6 Conclusions and Future Work

In this chapter, we have presented a methodology to generate deceptive game levels for Angry Birds. The proposed methodology can generate levels for six deception categories that the state-of-the-art Angry Birds playing agents are vulnerable to. Even though the idea of handcrafting deceptive levels for Angry Birds has been previously investigated, this is the rst attempt at generating deceptive levels for a complex physics-based game like Angry Birds. In addition, our approach generates the solution for the levels, which is a feature that is not available in any of the existing Angry Birds level generators and which could be bene cial for learning agents in their training process. The levels generated from the proposed procedure were evaluated using four metrics: stability, solvability, deceptiveness, and a comparison with human-created levels. The results of these metrics demonstrate that the generation process can competently create deceptive levels for the six deceptions considered.

3.6 Conclusions and Future Work

This work aims to enable the development of advanced Angry Birds playing agents that can perform well under deceptions by providing su cient training/testing data. To deal with these deceptions, the AI techniques of the current agents can be improved to expand their reasoning, planning, and generalizations skills. Future level generation research can involve generating more complex levels by combining multiple deception categories. Additional deception categories may also be proposed if more aws within the state-of-the-art agents can be identi ed. The deception categories we have examined, rolling/falling objects, path clearing, entity strength analysis, and non-greedy actions, nd parallels in real-world physical scenarios. Consequently, this study can establish a foundational framework for generating deceptive challenges within realistic physical environments.

Chapter 4

Phy-Q: A Testbed for Physical Reasoning

While humans possess a natural adeptness at comprehending the behaviours of physical objects and selecting appropriate actions to achieve objectives, this ability remains a formidable challenge for AI systems. In response, we propose a new testbed aimed at addressing this challenge by compelling agents to reason about physical scenarios and respond intelligently. Stepping beyond the domain of physical deceptions in Angry Birds discussed in the above chapter, our research advances by drawing inspiration from the foundational physical understanding inherent in human infancy and the essential skills vital for robotic navigation within real-world environments. Within this context, we identify fteen essential physical scenarios. In the testbed, we craft diverse task templates for these scenarios, ensuring that each scenario's task templates share a single speci c strategic physical rule for the solution. This structured approach enables us to evaluate two levels of generalization: local generalization and broad generalization We conduct a comprehensive evaluation involving human players, learning agents employing varied input types and architectures, and heuristic agents employing di erent strategies.

Motivated by the methodology used to calculate human IQ, we introduce the concept of the physical reasoning quotient (Phy-Q score). This score serves as a metric for gauging the physical reasoning intelligence of agents within the context of the considered physical scenarios. Our evaluation yields two signi cant ndings: Firstly, all agents fall signi cantly short of human performance. Secondly, learning agents, despite displaying competence in local generalization, grapple with grasping the underlying physical reasoning principles and struggle to achieve broad generalization We encourage the development of intelligent agents capable of attaining Phy-Q scores comparable to human performance.

To access the testbed, please refer to: https://github.com/phy-q/benchmark .

4.1 Introduction

A fundamental aspect of human cognitive progression lies in the capacity to deduce properties and behaviours of objects within physical environments [Yi* et al., 2020; Davis, 2006]. As discussed in Section 2.1, humans cultivate these physical reasoning skills right from their earliest stages of development. The evaluation of the ability to generalize performance across novel physical puzzles stands as a prevalent metric for gauging physical reasoning capabilities in contexts as varied as children [Diezmann and Watters, 2000; Cheke et al., 2012], animals Perdue et al. [2018]; Emery and Clayton [2009]; Taylor et al. [2009]; Jelbert et al. [2014], and AI agents Bakhtin et al. [2019]; Allen et al. [2020a,b]; Ahmed* et al. [2021].

Chollet's study [Chollet, 2019] on the measure of intelligence proposes a qualitative spectrum of di erent forms of generalization that includes local generalization and broad generalization. Current evidence [Lake et al., 2017; Jo and Bengio, 2017; Marcus, 2018; Justesen et al., 2018] suggests that contemporary deep learning models are local-generalization systems, i.e., systems that adapt to known unknowns within a single task. Broad generalization, on the other hand, can be characterized as `adaptation to unknown unknowns across a broad category of related tasks' and is being increasingly emphasized among the AI research community [Chollet, 2019; Geirhos et al., 2020; Firestone, 2020]. Moreover, when solving physics puzzles, it is common that a player must use a strategy to work out a plan and must use dexterity to accurately execute the strategic plan [Isaksen et al., 2017]. For instance, in a snooker game, a player needs to plan the path of the white cue ball, e.g., where it should go and where to stop, and then execute the strike that precisely produces the planned path. One of the views of cognitive psychology researchers is that humans possess inaccurate forward physics prediction models [McCloskey, 1983; Smith and Vul, 2013; Siegler, 1976; Keil, 2003; Gilden and Pro tt, 1994; Levillain and Bonatti, 2011; Lawson, 2012; Pro tt et al., 1990] and hence require practice to improve dexterity, high dexterity requirements of physics tasks make it unfair to compare AI agents' physical reasoning ability with average humans'. For example, when a human player fails a physics puzzle, it is hard to tell if it is due to incorrect physical reasoning or due to the inability to make precise actions. Despite the recent advancement in physical reasoning benchmarks and testbeds [Bakhtin et al., 2019; Allen et al., 2020a,b; Riochet et al., 2020; Yi* et al., 2020; Wolf, 2020; Ahmed* et al., 2021; Bear et al., 2021; Crosby et al., 2020], there is a lack of a benchmark or a testbed with human comparable strategic physics puzzles and one that explicitly evaluates learning agents' local and broad generalization.

To close these gaps, we propose a new testbed Phy-Q and the associated Phy-Q score that measures physical reasoning intelligence using the physical scenarios we identi ed. Inspired by the physical knowledge acquired in infancy and the abilities required by the robots to operate in the real world, we have created a wide variety of tasks with low dexterity requirements in the video game Angry Birds. Also, we included a module to aid trajectory planning. We believe the contributions from this work pave the way to

develop agents with human-level strategic physical reasoning capabilities.

Our main contributions can be summarized as follows:

- Phy-Q: A testbed for physical reasoning: We designed a variety of task templates in Angry Birds with 15 physical scenarios, where all the task templates of a scenario can be solved by following a common strategic physical rule. Then we generated task instances from the templates using a task variation generator. This design allows us to evaluate both the local and the broad generalization ability of an agent. We also de ne the Phy-Q score, a quantitative measure that re ects physical reasoning intelligence using the physical scenarios we considered.
- An agent-friendly framework: We propose a framework, which allows training multi-agent instances simultaneously with accelerated game-play speed up to 50 times.
- Establishing results for baseline agents: The evaluation consists of nine baseline agents: four of our best-performing learning agents, four heuristic-based agents, and a random agent. For each of the baseline agents, we present the Phy-Q score, the broad generalization performance, and the local generalization performance. We have collected human player data so that agent performance can be directly compared to human performance.
- A guidance for agents in AIBIRDS competition: In 2016, Angry Birds was considered to be the next milestone in AI where AI will surpass humans [Grace et al., 2018]. A time horizon of four years was predicted and so far, such a breakthrough seems very unlikely. In the AIBIRDS competition, heuristic methods generally perform better than their deep learning counterparts, but it remains unclear what has contributed to the gap in the performances. It has also not yet been analysed why current AI agents fall short when compared to humans. By the systematic analysis of agents from the AIBIRDS competition, we show how they need to be improved to achieve human-level performance.

4.2 Background and Related Work

In this section, we conduct a comparison between ten related physical reasoning benchmarks and two physics-based AI game competitions to show how Phy-Q testbed advances upon existing work. The comparison is made with respect to six criteria. 1) Measuring broad generalization in individual physical scenario/s, i.e., testing the ability of an agent to generalize to tasks that require the same physical rule to solve. 2) Categorization of tasks of the test environment into di erent physical scenarios, i.e., agents can be evaluated for individual scenarios to recognize the scenarios that they can perform well. 3) Procedural generation of tasks or variations of the tasks, i.e., the tasks/variants of the tasks in the test environment are created algorithmically, facilitating the users to generate any amount of data. 4) Destructibility of objects in the environment, i.e., if

Table 4.1: Comparison of Phy-Q with related physics benchmarks and competitions

Test Environment	generalization to individual physical scenario/s	categorization of tasks to physical scenarios	procedurally generated tasks variations	destructible objects	observe outcome of a desired physical action	human player data
PHYRE	7	7	X	7	X	7
Virtual Tools	7	X	7	7	X	X
OGRE	7	7	X	7	X	7
IntPhys 2019	X	X	X	7	7	X
CLEVRER	X	7	X	7	7	7
CATER	X	X	X	7	7	7
Physion	X	X	X	7	7	X
CoPhy	X	X	X	7	7	X
CausalWorld	X	X	X	7	X	7
RLBench	7	7	X	7	X	7
Computational Pool	7	7	7	7	X	7
Geometry Friends	7	7	X	7	X	7
AIBIRDS	7	7	X	X	X	X
Phy-Q (ours)	X	X	X	X	X	X

the environment contains objects that can be destroyed upon the application of forces. Having destructible objects makes the environment more realistic than an environment that only has indestructible objects since the agents need to consider the magnitude of the force that is applied to the objects. For example, when a robot moves a cup, it needs to reason that the force to exert should be large enough to grab the cup but not too large to break it. 5) Observing the outcome of a desired physical action, i.e., if an agent can physically interact and observe the outcome of the action the agent takes. 6) Inclusion of human player data, i.e., if the evaluation has results of human players.

For this comparison, we direct our attention towards the physical reasoning environments detailed in Section 2.1.1. These include PHYRE [Bakhtin et al., 2019], Virtual Tools game [Allen et al., 2020b], and OGRE [Allen et al., 2020a], which are game-based benchmarks; IntPhys [Riochet et al., 2020], CLEVRER [Yi* et al., 2020], CATER [Girdhar and Ramanan, 2020], and Physion [Bear et al., 2021], which are video based benchmarks; CoPhy [Wolf, 2020] which is an image-based benchmark; CausalWorld [Ahmed* et al., 2021] and RLBench [James et al., 2020] which are robotic benchmarks; Computational Pool [Archibald et al., 2010] and Geometry Friends [Prada et al., 2015], which are AI game competitions. We also included the AIBIRDS [AIBIRDS, 2023] competition for the comparison to show what properties in Phy-Q facilitate the systematic evaluation of AIBIRDS competition agents. Table 4.1 summarises the comparison.

The most closely related physical reasoning test environment to ours is PHYRE, which also consists of tasks to measure two levels of generalization of agents. PHYRE tests if agents can generalize to solve tasks within a task template (within-template) and

if agents can generalize between di erent task templates (cross-template). The cross-template evaluation in PHYRE does not guarantee that the physical rules required to solve the testing tasks exist in the training tasks. This leads to uncertainties in understanding agents' performance: inferior performance may not be an indicator of inferior physical reasoning but of a di cult training and testing split. Whereas the broad generalization evaluation in ours always ensures that physical rules required in testing tasks are covered in the training tasks, hence we guarantee a more systematic evaluation of the physical reasoning capabilities of AI agents. According to the task design in PHYRE, tasks need to be solved by trial and error. That is, even when the physical rule is known, multiple attempts are still needed to solve the tasks. Therefore, PHYRE promotes developing agents with physical dexterity. In contrast, we focus on strategy-based physical reasoning tasks that can be solved by a single attempt when the physical rule is understood. We promote developing agents that can understand a physical rule rather than taking a precise action in a physical environment (i.e., agents with strategic physical reasoning capabilities). Furthermore, a limited number of object shapes, motion, and material properties on scene dynamics hinders the ability of a comprehensive evaluation, as performing well on these tests might not indicate greater physical reasoning ability in more general and realistic contexts [Bear et al., 2021]. Therefore, compared to PHYRE, in Phy-Q we have 1) 3 more object shapes (rectangles, squares, and triangles) to allow more diverse physical dynamics, 2) destructible objects to make our environment more realistic, and 3) objects with three di erent materials that have di erent densities, bounciness, and friction to allow physical reasoning in a more realistic context.

As a recent visual and physical prediction benchmark, Physion evaluates algorithms' physical prediction capability using videos of eight di erent physical scenarios. Compared to Physion, the Phy-Q testbed has a more comprehensive set of 15 physical scenarios enabling the evaluation of agents in a wider range of physical scenarios. Phy-Q testbed requires agents to interact with the environment and select the desired action to accomplish physical tasks. Therefore, on top of predicting a physical event's outcome, agents need to apply the acquired physical knowledge to solve new situations, which is considered to be a more advanced type of task in Bloom's taxonomy [Krathwohl, 2002]. In addition, a study on forward prediction for physical reasoning [Girdhar et al., 2020] con rms that predicting the outcome in physical events does not necessarily increase performance in domains that require selecting an action. Therefore, the research problem focused in Physion is di erent to the problem in the Phy-Q challenge.

Despite Angry Birds being a simpli ed and controlled physics environment as compared to the much messier real physical world, no AI system has been developed that comes close to human performance. Since the start of this competition, many di erent AI approaches have been proposed, ranging from modern deep reinforcement learning methods to more old-school heuristic methods, e.g. qualitative physical reasoning methods. However, none of these approaches have reached the milestone of achieving human-level performance. One major reason is that an agent's performance in the competition does not enable an agent developer to identify the physical scenarios that the agent falls

short of. This is because the tasks in the competition are generally complex with multiple physical scenarios within the same task. In this work, we show how the Phy-Q testbed can be used to guide the competition agents through a systematic evaluation of agents' performance.

4.3 Phy-Q Testbed

In this section, we introduce our testbed, discuss the physical scenarios we have identi ed, and explain the evaluation settings we have used in the testbed.

4.3.1 Introduction to the Phy-Q Testbed

Based on the 15 identi ed physical scenarios (discussed in detail in Section 4.3.2), we develop a physical reasoning testbed using Angry Birds. As discussed in Section 2.1.2, we use a research clone of the game developed in Unity [Ferreira and Toledo, 2014b] to develop the testbed. The game environment is a deterministic 2D world where objects in motion follow Newtonian physics. The goal of the player is to destroy all the pigs using the provided set of birds. The initial state of a game level is physically stable (i.e., none of the objects is in motion), and the goal is not achieved. The action of an agent is to shoot the bird on the slingshot by providing the release coordinates relative to the slingshot and the tap time of the bird to activate powers (if available). This means the action space is essentially continuous. When playing, an agent takes a sequence of actions, i.e., shoots the birds in a prede ned order. The agent passes a game level when it destroys all pigs with the provided set of birds and fails otherwise. We do not provide the full world state that includes the exact location of objects in the simulator or their physical properties, such as mass and friction, to the agents, as these properties are not directly observable in the real world. Instead, an agent can request screenshots and/or a symbolic representation of the game level at any time while playing. A game screenshot is a 480 x 640 coloured image and the symbolic representation is in JSON format containing all objects in the screenshot represented as a polygon of its vertices (provided in order) and its respective colour map. The colour map provides the list of 8-bit quantized colours that appear in the game object with their respective percentages.

4.3.2 Physical Scenarios in Phy-Q Testbed

In this section, we explain the 15 physical scenarios we consider in our testbed. Firstly, we consider the basic physical scenarios associated with applying forces directly on the target objects, i.e., the e ect of a single force and the e ect of multiple forces [Sanborn et al., 2013]. On top of the application of single force, we also include scenarios associated with more complex motion, including rolling, falling, sliding, and bouncing, which are inspired by the physical reasoning capabilities developed in human infancy [Bliss and Ogborn, 1994; Kaiser et al., 1985]. Furthermore, we de ne the objects' relative weight [Wang et al., 2018], the relative height [Baillargeon and DeVos, 1991; Hespos and Baillargeon, 2001], the relative width [Wang, 2004], the shape di erences [Newcombe et al., 1999;

Baillargeon et al., 2008], and the stability [Wilcox and Chapa, 2004] scenarios, which require physical reasoning abilities infants acquire typically in a later stage. On the other hand, we also incorporate clearing path, adequate timing, and manoeuvring [Kemp et al., 2007], and taking non-greedy actions [Knox et al., 2012], which are required to overcome challenges for robots to work safely and e ciently in physical environments. To sum up, the physical scenarios[1] we consider and the corresponding high-level strategic physical rules that can use to achieve the goal of the associated tasks are mentioned below. Each of these scenarios tests a di erent aspect of the agent's skill, physical understanding, and planning ability.

1. Single force: Target objects have to be destroyed with a single force.
2. Multiple forces: Target objects need multiple forces to destroy.
3. Rolling: Circular objects have to be rolled along a surface to a target.
4. Falling: Objects have to fall onto a target.
5. Sliding: Non-circular objects have to be slid along a surface to a target.
6. Bouncing: Objects have to be bounced o a surface to reach a target.
7. Relative weight: Objects with correct weight have to be moved to reach a target.
8. Relative height: Objects with the correct height have to be moved to reach a target.
9. Relative width: Objects with the correct width or the opening with the correct width have to be selected to reach a target.
10. Shape di erence: Objects with the correct shape have to be moved/destroyed to reach a target.
11. Non-greedy actions: Actions have to be selected in the correct order based on physical consequences. The immediate action may be less e ective in the short term but advantageous in the long term. i.e., reach fewer targets in the short term to reach more targets later.
12. Structural analysis: The correct target has to be chosen to break the stability of a structure.
13. Clearing paths: A path have to be created before the target can be reached.
14. Adequate timing: Correct actions have to be performed within time constraints.
15. Manoeuvring: Objects have to be carefully guided to reach a target.

[1] Example tasks representing the 15 scenarios and the solutions for those tasks are available at https://github.com/phy-q/benchmark

(a) Single force (b) Multiple forces (c) Rolling

(d) Falling (e) Sliding (f) Bouncing

(g) Relative weight (h) Relative height (i) Relative width

(j) Shape di erence (k) Non-greedy actions (l) Structural analysis

(m) Clearing paths (n) Adequate timing (o) Manoeuvring

Figure 4.1: 15 example tasks in Phy-Q representing the 15 physical scenarios. The slingshot with birds is situated on the left of the task. The goal of the agent is to kill all the green-coloured pigs by shooting birds from the slingshot. The dark-brown objects are static platforms. The objects with other colours are dynamic and subject to the physics in the environments.

4.3.3 Task Templates and Task Generation

We design task templates in Angry Birds for each of the 15 physical scenarios mentioned above. A task template can be solved by a speci c strategic physical rule and all the templates belonging to the same scenario can be solved by the high-level strategic physical rules discussed above. The task templates are handcrafted and we ensure that if an

agent understands the strategic physical rule to solve the template, they can solve the template without requiring highly accurate shooting, e.g., the template can be solved by shooting at a speci c object rather than shooting a speci c coordinate. This design criterion is followed to reduce the dexterity requirement when solving the tasks in our testbed. We have developed 2-8 task templates for each scenario, totalling 75 task templates. Figure 4.1 shows example task templates for the 15 scenarios.

We generate 100 game levels from each template and we refer to these game levels as tasks of the task template. All tasks of the same template share the same strategic physical rule to solve. Similar to PHYRE, the tasks are generated by varying the location of the game objects in the task template within a suitable range. Furthermore, various game objects are added at random positions in the task as distractions, ensuring that they do not alter the solution of the task. When generating the tasks each task template has constraints to satisfy such that the physical rule of the template is preserved. For example, the constraints can be which game objects should be directly reachable by a bird shot from the slingshot, which game objects should be unreachable to the bird, which locations in the game level space are feasible to place the game objects, etc. These constraints are speci c to each task template; they were determined by the template developers and hardcoded in the task generator.

Although we provide 100 tasks for each task template, we also provide a task variation generation module to generate more tasks if needed. Figure 4.2 shows task templates of relative height scenario and example tasks generated from a single task template. All the 75 task templates and example task variations can be found in Appendix B.2.

4.3.4 Proposed Evaluation Settings

The spectrum of generalization proposed by Chollet [Chollet, 2019] can be used to measure intelligence as laid out by theories of the structure of intelligence in cognitive psychology. There are three di erent levels in the spectrum: local generalization, broad generalization, and extreme generalization. Having 15 physical scenarios, a variety of task templates for each scenario, and task variations for each task template, our testbed is capable of evaluating all three di erent generalization levels. However, in this work, we focus on measuring the *local generalization* and the *broad generalization* of agents, as local generalization is the form of generalization that has been studied from the 1950s up to this day and there is an increasing research interest in achieving broad generalization [Chollet, 2019].

Formally, consider each scenario $scenario_i$ in the set of all scenarios $SCENARIO$, where $|SCENARIO| = 15$, we de ne template $template_j \in scenario_i$, where $|scenario_i| = NT_i$ and NT_i is the number of templates we included for $scenario_i$. As we have 100 tasks for each template, we de ne $task_k \in template_j$, where $|template_j| = 100$ for all templates. i.e., each scenario is a set of tasks and the tasks in a scenario are partitioned into templates.

To evaluate local generalization within a particular template, we train an agent on some

Figure 4.2: The rst and the second rows show the task templates of the relative height scenario, and the third and fourth rows show the tasks generated using the second task template in the rst row.

(80% in practice) of the tasks in a template and evaluate it on the remaining tasks of the same template. More formally, to evaluate local generalization, we train an agent on a subset of the tasks from a template j, $TrainTasks^j_{local}$ $template_j$ and test on the rest of the tasks within the same template as $TestTasks^j_{local} = template_j \cap TrainTasks^j_{local}$. We evaluate the local generalization for all of the 75 templates.

To evaluate broad generalization within a particular scenario, we train an agent on the tasks of some of the templates of that scenario and evaluate it on the tasks of the other templates of the same scenario (See Appendix B.4 for the division of task templates for training and testing for each scenario). More formally, to evaluate the broad generalization, we train an agent on the training tasks on a subset of templates, $TrainTasks^i_{broad} = [\{j \mid template_j \in scenario_i^{train}\}\, TrainTasks^j_{local}$, and evaluate on the testing tasks of the rest of the templates within the same scenario, $TestTasks^i_{broad} = [\{j \mid template_j \in scenario_i^{test}\}\, TestTasks^j_{local}$, where $scenario_i^{train}$ is the set of training tem-

Figure 4.3: The local generalization and the broad generalization evaluation settings.

Figure 4.4: An illustration of how generalizing a physical rule is evaluated in the broad generalization evaluation using bouncing scenario as an example.

plates and $scenario_i^{test}$ is the testing templates for $scenario_i$. We also make sure that $\forall i; scenario_i^{train} \setminus scenario_i^{test} = \emptyset$. We evaluate the broad generalization performance for all 15 scenarios.

In broad generalization, we assume that if an agent learns the required strategic physical rule to solve a set of task templates, it should be able to apply the same strategic physical rule to solve unseen tasks from other templates within the same scenario. As opposed to this, the performance on local generalization evaluation may not represent an agent's physical rule generalizing capability but memorizing a special-purpose heuristic. Figure 4.3 is a diagrammatic representation of the two evaluation settings, and Figure 4.4 shows an illustration of how generalizing a physical rule is evaluated in the broad generalization evaluation setting.

Our physical reasoning quotient (Phy-Q) is inspired by deviation IQ [Wechsler, 1958] for humans. We calculate Phy-Q of an agent using the results of our broad generalization evaluation since we consider that this evaluation measures the agent's ability in generalizing strategic physical rules. When calculating the Phy-Q we exclude the first two scenarios: single force and multiple forces, as the solution for the two scenarios is directly shooting the bird to the exact location of the pig. Given that we have provided a trajectory planner for both humans and agents, solving the tasks of the two scenarios is straightforward. This is also evident from the exceptionally high results (in Section 4.5.2) of the Pig Shooter agent that directly shoots at the pigs without doing any physical