Learning Physical Intuition for Robotic Manipulation

Oliver M Groth

St Peter's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2021

Learning Physical Intuition for Robotic Manipulation
Candidate: Oliver M Groth
Supervisors: Professor Ingmar Posner, Professor Andrea Vedaldi
Examiners: Professor Victor Prisacariu, Professor Andrew Davison
Date of examination: 23rd November, 2021
Date of revision: 17th December, 2021

University of Oxford
Applied Artificial Intelligence Group (A2I)
Visual Geometry Group (VGG)
Department of Engineering Science

Statement of Authorship

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Oliver M Groth
St Peter's College
December 2021

10100 01111

10100 01000 00101

01111 01110 00101

10111 01000 01111

01000 00101 01100 00100

01101 00101

10100 01111 00111 00101 10100 01000 00101 10010

10111 01000 00101 01110

00101 10110 00101 10010 11001 10100 01000 01001 01110 00111

00101 01100 10011 00101

00110 00101 01100 01100

00001 10000 00001 10010 10100

Acknowledgements

This thesis would not have been possible without the guidance and support of my supervisors, Prof. Ingmar Posner and Prof. Andrea Vedaldi. Their advise, encouragement and constructive feedback has been invaluable in shaping my work over the last four years and has greatly helped me to grow as a researcher myself. I am also very grateful for the support of Prof. Martin Riedmiller who has supervised my internship at DeepMind and has helped me applying my research ideas to new challenges. Lastly, I would like to express my gratitude to my previous supervisors, Prof. Carsten Rother and Prof. Fei-Fei Li, who have introduced me to the exciting world of Artificial Intelligence research and without whom I would not have dared to apply to a PhD program in the first place.

I am also incredibly grateful for the support and companionship of my collaborators and labmates which I have had the privilige to work with at the Oxford Robotics Institute, at the Visual Geometry Group and at DeepMind. From bouncing wild research ideas, through sharing the burden of the submission deadlines all the way to cheering each other's successes - you have had a big impact on my research and made the countless hours along the way infinitely more enjoyable. Especially, dear Fabian, Sébastien, Olivia, Walter, Ștefan, Chia-Man, Sudhanshu, Yizhe, Markus, Giulia, Vibha and Tim – thank you! Also, I would like to thank Gül, João, Sam, Shu, Andrew and Lili for helping me to think about the 'bigger picture' of our research and dabbling in many entertaining philosophical discussions together. Credit for a big part of my D*Phil* goes to you. In addition to my colleagues, I also want to thank my wonderful friends – in Oxford and beyond. You have always given me a space where I could feel at home, recharge my batteries and detach from the rollercoaster ride of research whenever I needed it.

Finally, I owe my deepest gratitude to my parents. Bettina and Thomas: Without your unceasing trust, patience and support, I would not have made it to this point. I am forever grateful for the 'roots and wings' you have provided me with.

Abstract

When we compare object manipulation capabilities in humans and contemporary robots, we observe an intriguing dichotomy: On one hand, robots have access to advanced compute capacity and precise models of physics, yet their object manipulation skills are comparatively narrow and brittle. On the other hand, the human understanding of physics is allegedly acquired from experience and exhibits many predictive shortcomings, yet their manipulation skills far exceed any contemporary robot's. Motivated by this observation, this thesis studies the question how much robotic manipulation can benefit from embracing data-driven, approximate models of physics and poses the hypothesis that a tight integration of intuition and control can unlock sophisticated manipulation behaviour.

In particular, three aspects of physical intuition are investigated: (i) high-level intuitions for visual task assessment and their application in object stacking and tool use, (ii) low-level intuitions for rigid-body motions and their application in rearrangement planning and visuomotor control, (iii) the integration of dynamics approximation into control policy learning and its application in structured exploration of an environment.

In the first part, we demonstrate the effectiveness of a visual stability classifier in planning and constructing stable stacks of objects with varying geometries. We also employ a similar task classification technique in a goal-reaching task and show that the associated variational latent space induces an affordance manifold which can be traversed to synthesise suitable tools for a given task. In the second part, we demonstrate that the introduction of dynamics modelling into an object-centric latent space facilitates object disentanglement from raw visual training data and allows to generate physically plausible scenes and videos from scratch. Visual dynamics approximation is also used in our novel, goal-conditioned, visuomotor control architecture where it enables zero-shot transfer to unseen object rearrangement tasks. Finally, we integrate dynamics forecasting and control policy learning in the third part of this thesis and optimise both components using a curiosity objective. This setup leads to the unsupervised emergence of complex, human interpretable manipulation and locomotion behaviour and highlights the crucial importance of physical intuition in the learning process of sophisticated, embodied behaviour.

Contents

1 Introduction

Physical intuition, imagination and forecasting a scene's evolution over time are remarkable human capabilities that enable our sophisticated object manipulation skills (cf. e. g. Rosenbaum et al. [165]). For example, we can easily tell whether a stack of plates would topple, which block of a Jenga tower can be safely removed or how to place objects on a see-saw such that it remains balanced. Besides inferring its geometric and physical properties, we are also very proficient in estimating an object's *affordances* (Gibson [54]), i. e. what an object enables us to do in our environment like reaching around an obstacle using a hook-like stick. Multiple studies show the effectiveness of physical intuition for example when human subjects were asked to extrapolate trajectories of sliding objects (J. Wu et al. [210]), estimate relative masses of objects (J. B. Hamrick et al. [74]) or imagine how structures can be built from elementary geometric building blocks (Yildirim et al. [226]). It is speculated that humans possess an internal model of physics which they use to base their physical reasoning on (P. W. Battaglia et al. [13]). However, it remains unclear how this mental model is acquired and how much it aligns with real Newtonian physics since it is prone to common misconceptions and can be altered based on certain training stimuli (Kubricht et al. [112]).

Although no comprehensive model of human physical intuition has been formulated to date, there is growing consensus around the fact that such a model is likely acquired from experience, e. g. via playful interaction with the world during infancy (Baillargeon [8]). Humans develop intuitions about physical phenomena like *object permanence* or the existence of *gravity* at an early age (Baillargeon et al. [9] and Kim and Spelke [98]) and utilise this knowledge to manipulate objects in their environment far surpassing any contemporary robot's capabilities. Crucially, this mental model of physics is an approximation which is just good enough to predict the outcome of certain events, e. g. estimate the trajectory of a ball in oder to catch it, but it lacks any exact modelling. Juxtaposing human manipulation capabilities with those of robots reveals an interesting discrepancy: While human intuitions about physics are only data-driven approximations with known shortcomings, their tight integration with motor control, e. g. in hand-eye coordination, affords highly robust and adaptive manipulation capabilities. In contrast, robotic systems are classically built with exact models of physics which are leveraged for *task-and-motion-planning* (TAMP). However, such model-based approaches need to spend considerable effort on matching up noisy sensor input to their model priors, for instance when a raw video stream needs to be mapped to a model of 3D objects which can be ultimately utilised by a planning algorithm. Furthermore, anticipating all circumstances under which model and planner would need to operate is close to impossible. This lack of adaptability makes classical robotic manipulation systems appear brittle – despite their superior models and planners – compared to human manipulation which operates on much less precise physical models but is grounded in observations. Given the dichotomy between the precise, yet brittle modelling of physics in robotic manipulation and the imprecise, yet robust intuition about physics in human manipulation, it appears natural to ask: *How much could robotic manipulation benefit from embracing data-driven, approximate models of physics?* Trying to answer this question is the main subject of this thesis.

With the advent of deep learning (e. g. Goodfellow et al. [58]), approximate, data-driven modelling has revolutionised many robotics-adjacent fields like Computer

Vision (e. g. He et al. [75], Krizhevsky et al. [111], and Simonyan and Zisserman [180]) and Natural Language Processing (e. g. Brown et al. [22], Cho et al. [28], Sutskever et al. [190], and Vaswani et al. [197]). Within robotics, Mnih et al. [139] have pioneered a new avenue for model-free control employing deep reinforcement learning and subsequent seminal works have paved the path towards end-to-end robotic learning systems (Levine et al. [118, 120] and Lillicrap et al. [124]). With so many domains being impacted by deep learning methods, physics modeling makes no exception: P. Battaglia et al. [14], Watters et al. [204], and J. Wu et al. [210] pioneered a new field within Machine Learning aptly named *neural physics*. These new techniques unlocked the study of new, approximative – *intuitive* – models of physical systems and their application in downstream applications. Robotic manipulation provides a rich field for experimentation with intuitive models of physics, which is the reason why it was chosen to serve as the experimental environment of the research conducted in this thesis. Additionally, it enables to transfer the concepts from Cognitive Science such as imagination and curiosity to concrete contributions in core fields of Machine Learning and Robotics such as *representation learning*, *neural physics* and *visuomotor control*.

In this work, we posit that a tight integration of intuitive physics models with robotics control renders possible sophisticated manipulation skills in artificial agents. To that effect, we investigate a broad range of intuitions and their repsective applications in manipulation: from high-level classification of complex phenomena like structural stability and tool affordances to low-level regression of rigid-body motion. We show that such intuitive models about physics faciliate the building of stacks comprising of diverse objects, enable the imagination and selection of suitable tools for reaching tasks, enhance generalisation of visuomotor skills and even lead to the unsupervised emergence of complex embodied behaviour. An overview over the key contributions of this thesis is provided in Fig. 1.1.

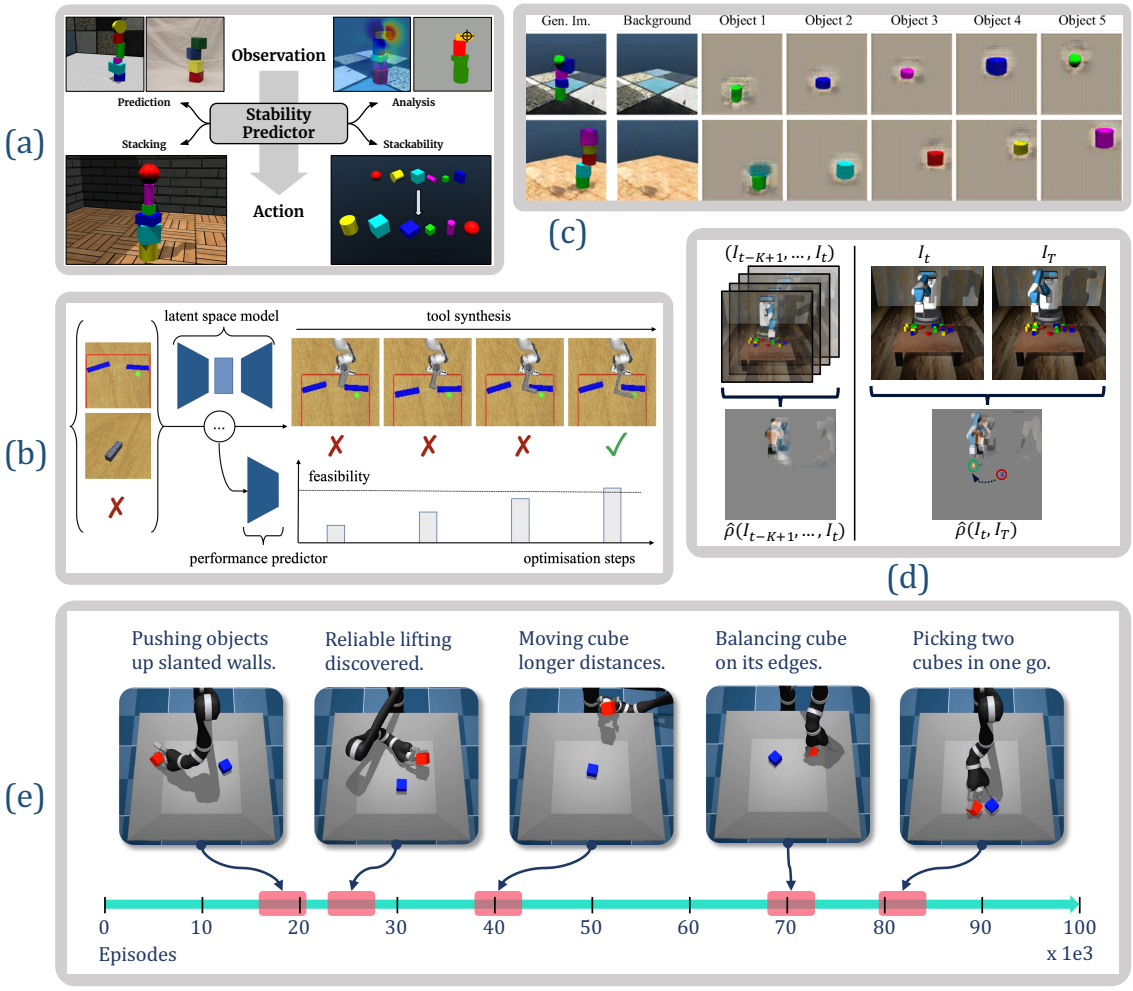


Figure 1.1: An overview over the key contributions of this thesis. **(a)** In ShapeStacks (Chapter 3), we train a visual predictor of *structural stability* and utilise it in a control loop to build stacks from unordered sets of objects. **(b)** In Chapter 4, we employ a similar, high-level intuition classifier about a tool's *applicability* and demonstrate its effectiveness to imagine suitable tools for a given reaching task via latent interpolation. **(c)** RELATE (Chapter 5) employs a graph network to learn from raw video examples how to explain a scene as a set of objects with approximate rigid-body motion. It affords physically plausible composition of novel scenes and faithful extrapolation of trajectories. **(d)** GEECO (Chapter 6) is an end-to-end visuomotor controller which can be conditioned on a single target image. Its internal approximation of dynamics facilitates task inference and affords zero-shot skill transfer to unseen tasks. **(e)** In Chapter 7 we learn a forward-predictive dynamics model with *curiosity-based exploration* and demonstrate that the acquisition of a physical intuition leads to the emergence of complex manipulation behaviour in a completely unsupervised way.

1.1 Thesis Outline

We begin our investigation into the utility of physical intuition for robotic manipulation by reviewing the existing literature and establishing core concepts in Chapter 2. The topic of this thesis is uniquely positioned as it sits in the intersection of multiple disciplines: The motivation of physical intuition is firmly rooted in Cognitive Science; the methodology is drawn from core Machine Learning concepts such as object-centric representation, neural physics approximation and latent space traversal; and its application extends into the domains of object manipulation and visuomotor control in Robotics. We dedicate individual sections to each of these areas for a general overview and provide more focused literature reviews for the topics of individual chapters in Sections 3.2, 4.2, 5.2, 6.2 and 7.2.

In the remainder of this section, we provide an overview over the different chapters in this thesis and how they corroborate individual aspects of our central hypothesis about the benefits of learned physical intuitions in robotic manipulation scenarios. Each of the following subsections presents the central deliberations, contributions and conclusions of Chapters 3 to 7 in that order. For an in-depth description of the methodology and experimental results we refer the reader to the respective, self-contained chapters.

In Chapter 8 we revisit our central hypothesis about the conduciveness of physical intuition to robotic manipulation and re-evaluate the findings of the previous chapters in its context. Specifically, we highlight the contributions of intuitive models about physics to central aspects of robotic manipulation such as visuomotor control, inference of affordances and rearrangement planning. We also reflect critically on their still-existing limitations and present an outlook over potential avenues of future work building upon the contributions in this thesis. Finally, we conclude with a discussion on the central role of physical intuition in the learning process of control policies for embodied systems in general and deliberate on its implications beyond the concrete subject of robotic manipulation.

1.1.1 Object Stacking using Visual Stability Prediction

The popular domain of *stacking games* such as Jenga[1] and Bandu[2] has been chosen as a first environment to investigate the acquisition and application of physical intuition. Stacking involves an understanding of an object's physical properties such as shape, mass and centre-of-mass support and is typically one of the earliest demonstrations of human physical intuition. At a very young age, toddlers often play with bricks to acquire a model of rigid-body-physics (Baillargeon [8]), e. g. that all objects fall down due to gravity, and simultaneously hone their motor skills in the process. In Chapter 3 we design a simulation environment called ShapeStacks (Groth et al. [64]) which is inspired by a real-world stacking game and enables us to conduct experiments investigating the structural stability of stacks consisting of objects of varying geometry. Specifically, we investigate whether an approximation of structural stability can be learned from raw visual input and how such an intuitive stability predictor could be utilised in a control loop.

While Lerer et al. [117] and J. Wu et al. [208] have previously investigated stability prediction with and without physical priors in this scenario, their studies have been restricted to small stacks of only up to six regularly shaped cubes. Our work extends prior art by including different shapes – cubes, cylinders and spheres – with varying sizes and aspect ratios to study the different factors of influence governing structural stability more comprehensively as described in Section 3.3.1. We find in Section 3.4.1 that our model yields state-of-the-art stability prediction performance on simulated and real data. In Sections 3.4.2 and 3.5.1 we also demonstrate that the network learns plausible correlations between visual input and physical attributes governing stability. For instance, it identifies uneven surfaces or unsupported objects as indicators of an imminent collapse and ranks an object's contribution to a structure's stability by its pose and geometry. In Sections 3.5.2 and 3.5.3 we present experiments, where we employ the network's intuition about stability to control a simulated, sampling-based object placement process. This enables the construction of stable stacks with up to

[1] https://jenga.com/

[2] https://www.hasbro.com/common/instruct/Bandu.PDF

twelve objects and balancing instable structures with counter-weights. We conclude from this study that an image classification network can learn a viable approximation of a physical phenomenon such as structural stability and the prediction landscape is smooth enough to govern a downstream manipulation process.

Despite the successful application of vision-based physical intuition, it remains important to understand whether the 'reasoning' about physics in such a model follows sound physical principles, e. g. as investigated by Zhang et al. [231]. Fuchs et al. [51] present a framework to identify and mitigate biases in visual classification which has been applied to the stability predictors used in ShapeStacks. The investigation is supported by a custom set of ShapeStacks scenarios featuring deliberate structural biases, e. g. where a local instability is counter-balanced by another piece higher up in the stack. The study reveals that stability prediction is easily misled by salient visual cues and only exploits correlations between local visual patterns and a global stability label. It can be concluded that visual stability prediction is not a principled application of the underlying physical principles but only a correlation-based approximation grounded in visual pattern recognition. However, as with many applications of deep learning, the data-driven approximation of a process can yield feasible shortcuts to enable downstream applications such as simple stable object stacking.

1.1.2 Tool Synthesis via Affordance Optimisation

Having demonstrated the efficacy of a high-level visual intuition classifier in a simplified manipulation control loop in Chapter 3, we extend this idea to another manipulation domain and refine the method in Chapter 4. In Groth et al. [64] we employ the stability classifier in a discriminative way, i. e. using the stability signal in a closed control loop. However, human physical intuition extends beyond reactive visuomotor control and can also be utilised to imagine interactions with other objects a priori, e. g. how to leverage them to accomplish a certain task. According to *Gibson's definition of affordances* (Gibson [54]), an object can extend an agent's capabilities by providing new means of interaction with the environment. Although not strictly

confined to a physical manifestation, affordances are most commonly studied in physically grounded applications such as the usage of tools. When humans or animals use tools the physical properties of the tool involved, e. g. its geometry or mass, directly *afford* new interactions with the environment. For instance, the heavy head of a hammer affords to hit a nail or the use of a long stick extends one's arm length and affords to reach objects which might be unreachable otherwise.

In tool use, physical intution serves a similar purpose as in the control loop for object stacking: as a prediction of a high-level outcome of a physical process which is approximated by the agent for faster and computationally cheaper evaluation. Cognitive Science provides us with a well-established testbed for the study of physical intution in connection to tools. Specifically, we follow Ambrose [5] and Emery and Clayton [39] to investigate how high-level visual intution can be employed to facilitate tool use in an agent. We construct a simulation environment presenting a reaching task where a robot must reach an obstructed target button while remaining outside a delineated workspace area. The only way to accomplish the task is to use a suitable stick- or hook-like reaching tool to satisfy all constraints. Transfering the visual classification approach developed in Groth et al. [64], we prepare a dataset of task images paired with tool geometries and train a classifier to distinguish feasible from infeasible reaching combinations. However, in order to probe the utility of the classifier beyond simple tool selection, we define a *tool synthesis task* where the agent needs to *imagine* a suitable tool geometry leveraging its intuition about reaching feasibility.

In order to turn a trained classifier of an object's reaching affordance into a generative process imagining a suitable tool from scratch, we introduce three new technical parts to the system in Section 4.3: Firstly, we encode the 3D meshes of the tools using a *neural renderer* Kato et al. [96] and N. Wang et al. [201]. The neural renderer represents the 3D shape as a latent vector of vertex deformations which facilitates the estimation of the factors of variation of tool geometry in the latent representation and enables a smooth interpolation between different tool shapes during latent space

traversal. Secondly, we connect the classification network for reaching feasibility to the structured latent space of a tool to establish a direct connection between the factors of variation in the geometry and the classification label. Lastly, when presented with the picture of a new reaching task, we employ *activation maximisation* (Erhan et al. [41] and Simonyan et al. [179]) using the classifier's gradients to optimise the latent code and synthesise a tool geometry which maximises the perceived task feasibility.

The experiments presented in Section 4.4 show that a high-level intution about an abstract reachability affordance can be successfully turned into a conditional 3D generative process in this framework synthesising suitable tool geometries with high accuracy. This imaginative application of a physical intuition complements the discriminative use in Groth et al. [64] and demonstrates the potential of the approach for control and planning aspects of robotic manipulation. While we acknowledge that the task of tool synthesis is not immediately applicable to a real robotic system, we would like to emphasise that the general framework of coupling a variational latent space attached to a high-level classifier is of broader applicability. This framework holds the tantalising prospect of radically simplifying many complex optimisation processes typically encountered in planning in robotics. For instance, our approach has been adpoted by Mitchell et al. [137] demonstrating that a real quadruped robot is able to walk via pose interpolation in latent space driven only by gradient signals of high-level stability and stance classifiers.

1.1.3 Unsupervised Object-Centric Dynamics Modelling from Raw Vision

The classifier-based, high-level intuition approach presented in Chapters 3 and 4 has two important limitations: Firstly, it can only be applied to assess or imagine a scene *statically* given an input image, e. g. a view of stack or of a reaching task setup. Secondly, the classifiers are trained in a supervised way and require labeled datasets like examples of stable and unstable towers or feasible and infeasible reaching

setups. In Chapter 5 we address both issues by proposing RELATE, a novel object-centric generative model which learns to approximate rigid-body motions from raw videos.

In order to overcome the limitations outlined above, we add two important modelling assumptions. Firstly, we employ a *factorised, object-centric latent space* to represent a visual observation, e. g. a frame in a video, as a set of latent variables. In this setup, each object in the scene is represented as a tuple $(\vec{z}_i, \vec{\theta}_i)$ encoding an object's appearance (shape and texture) and relative position in 2.5D, respectively. Secondly, the *factorisation* of the latent space facilitates the approximation of visual dynamics by reducing the physical modelling involved to *point-mass dynamics* over the set of $\vec{\theta}_i$ while the visual appearance is factored out into the set of $\vec{z}_i$.

Our work builds upon BlockGAN (Nguyen-Phuoc et al. [150]), an object-centric generative model featuring an appropriately factorised latent space of appearance and position variables. However, Nguyen-Phuoc et al. [150] have only considered independent objects in static images. In contrast, we extend their method in Section 5.3 by drawing inspiration from the rich body of work on *neural physics approximation for rigid bodies* (P. Battaglia et al. [14], M. B. Chang et al. [27], Van Steenkiste et al. [196], and Watters et al. [204]) and augment our model with a *relationship module*, Γ, which models the spatial relationships between all objects in a scene and how they evolve over time. Specifically, Γ adjusts positions of objects such that they are *physically plausible* under the training data observed, i. e. they do not intersect with each other or hover above the ground without support. Furthermore, our work differentiates from prior art in this domain (Janner et al. [90] and Y. Ye et al. [223]) in that the training of RELATE does not rely on annotations of object positions such as instance maps or bounding boxes. Instead, a 2D position regressor is part of the discriminator set of the GAN framework which ensures consistency between the latent position variable and the position of the rendered object in the image and regularises against mode collapse in the position encoding. In this way, RELATE can be trained from raw images and

videos only requiring one hyperparamter of the expected upper bound of objects per frame.

RELATE enables multiple applications for physically plausible imagination of visual scenes. Firstly, we demonstrate in Section 5.4.1 that the consistency enforcement of position regression facilitates the *decomposition* of a visual scene into its constituent components as each object needs to be mapped onto a spatially distinct latent variable. Secondly, the structured, object-centric latent space affords *targeted scene editing* such as the insertion or deletion of objects or the change of their position and appearance as described in Section 5.4.2. Thirdly, the recursive application of the relationship module Γ allows to predict the evolution of the object positions over time, effectively sampling an entire video from scratch in a computationally efficient way. In Section 5.4.3 we present video generation results using RELATE and put their visual and physical fidelity into perspective. In summary, RELATE contributes to the imagination aspect of physical intuition, enabling the (targeted) generation of scenes and extrapolating their dynamics into the future. These features can also be leveraged for object manipulation tasks in model-based control or visual planning approaches as examplified in complementary work such as Ebert et al. [35] and Veerapaneni et al. [198].

1.1.4 Visual Dynamics Representation for Transferable Visuomotor Skills

In Chapters 3 to 5 we explore the utility of high-level and low-level physical intution in manipulation-oriented imagination tasks which are mostly conducive to planning. However, the execution of pre-conceived goals, e. g. the rearrangement of objects in a scene, is of equal importance when deploying such approaches to a robotic system. In order to establish the connection between given or imagined goals and robotic control, we develop in Chapter 6 an end-to-end visuomotor control (VMC) model which can be used to rearrange objects in a scene conditioned on a visual instruction.

The task of *object rearrangement* is central to many object manipulation setups and presents a considerable challenge for contemporary robotics (Batra et al. [12]). Surveying the space of related work in goal-conditioned visual object manipulation, we identify a central limitation: *versatility* and the immediate application to novel tasks. Prior art typically accomplishes goal-conditioning by using *model-predictive control* (MPC) or *few-shot imitation learning*. An established line of work in manipulation using MPC is *Visual Foresight* (Ebert et al. [35], S. Nair and Finn [147], and Xie et al. [217]), which employs an action-conditioned video prediction model to sample trajectories, which get the perceived state closer to a desired target state. However, as the imprecision of the video prediction compounds over long time horizons, complex motions, and with unfamiliar objects, these approaches are typically limited to pushing or placing operations moving known ojects to new locations. Few- or one-shot imitation approaches on the other hand typically learn an end-to-end visuomotor control policy during training time which is *finetuned* on a few demonstrations of the novel task during test time. However, this dependency on full task demonstrations and finetuning during test time limits the deployability of such approaches as demonstrations might not always be attainable or on-device controller optimisation feasible. Our proposed visuomotor controller, GEECO, addresses these limitations by featuring a conditioning scheme which is learned end-to-end and can adapt to a new task communicated via a single target image instantly without the need for additional demonstrations or finetuning.

In Section 6.3 we present how GEECO extends the end-to-end VMC architecture by S. James et al. [88] to incorporate goal-conditioning. Specifically, we leverage *dynamic images* (Bilen et al. [18]) as a visual representation of motion dynamics and object displacement. This technique affords an intuitive summarisation of the control trajectory into two parts: the most recent motion within the scene and the remaining displacement difference of objects with respect to a target configuration provided via the goal image. As a consequence, the downstream network can leverage this representation to learn the control policy more efficiently and robustly as it reduces the influence of textures and shifts the controller's focus to dynamic and geometric

aspects of perception. Furthermore, the setup enables GEECO to be trained on raw robotic demonstrations without the need for additional labels such as task information or object annotations.

In Section 6.4 we compare GEECO against two representative models for visual MPC and one-shot imitation and demonstrate significant performance improvements for goal-conditioned pushing and pick-and-place tasks. Additionally, we probe whether the pick-and-place skill, which our model acquired on simplistic demonstrations of up to four objects can be transfered to more challenging scenarios. We do this by applying it to novel scenarios, which feature a heavily cluttered workspace, visual background noise and novel object geometries to handle. We show that our model transfers its learned skill successfully and is able to execute novel manipulation tasks based on a single target image even if the scenario significantly deviates from the training domain. In this way GEECO makes a significant contribution in the domain of end-to-end trainable, goal-conditioned VMC architectures for rigid object manipulation. It is also worth noting that this transfer is being accomplished without any *visual domain randomisation* techniques typically employed to make VMC more robust. This can be attributed to the nature of the dynamic image representation which largely supresses textures and reduces videos to moving object outlines thereby alleviating the need for texture randomisation. Hence, we conjecture that the invariances afforded by GEECO's architecture can also make an important contribution to the simulation-to-reality transfer aspect for visual manipulation controllers.

1.1.5 Emergence of Behaviour during the Acquisition of Physical Intuition

In Chapters 3 to 6 we explore different high- and low-level approximations of physics such as structural stability, tool applicability or rigid-body motions to facilitate the execution of different manipulation tasks. However, the training of the intuition is always disconnected from the application, i. e. the model is trained offline on a dataset and then applied to solve a manipulation task. While being convenient from a machine learning perspective as it allows for controlled evaluation of the gains

afforded by the intuitive physics model, an important counter argument against this setup is that in humans, physical intuition is predominantly learned in an *interactive* way (Goswami [60]). Therefore, in the final Chapter 7 we change the experimental setup such that a forward-predictive model about the environment's dynamics is learned interactively by an agent exploring the environment.

We realise this by employing a *curiosity*-based reinforcement learning setup (Schmidhuber [170]). In this setup, an agent is modelled by two neural networks: the *world model* which predicts the outcome of an action to take in the environment based on the current state and the policy network which predicts the next action to take to maximise a given reward function. In curiosity learning, the reward given for a particular `(state, action, next_state)` transition is scaled by the current error of the world model when predicting the `next_state` from the `(state, action)` input. In that way, the agent is encouraged to explore areas of the state-action-space which are not yet well predictable. In turn, this leads to the collection of relevant data to improve the world model and develop a better understanding of the environment's dynamics.

Our curiosity learning model presented in Section 7.3 implements the world model and policy training in two independent learning loops which are executed *off-policy*. When we apply this learning setup to different robots in simulated manipulation and locomotion domains, we observe that the maximisation of the curiosity objective leads to the emergence of complex, human-interpretable behaviour such as targeted object lifting on a robot arm or single-foot balancing on a humanoid robot (cf. Section 7.4.1). Naturally, as the curiosity objective keeps changing during training of the agent, the corresponding behaviour changes as well. Previously, curiosity-based learning approaches have been applied successfully to 2D and 3D computer games (Burda et al. [23] and Pathak et al. [157]) and have been shown to 'solve' such games, i. e. reach the end of the level, without any additional reward signal just based on the efficient exploration afforded by this technique. Other work has also shown that curiosity-based exploration can serve as an efficient way to pre-train a policy network before

finetuning it on a concrete downstream task (Sekar et al. [174]). However, based on our experimental observations, we believe that neither of those approaches harnesses the full potential of curiosity learning as any 'final' policy only retains a fraction of the diverse and potentially useful behaviours which the agent has uncovered during curious exploration. In contrast, we argue to employ curiosity learning differently by utilising the diverse behaviours which it discovers *during* exploration. We investigate the utility of emerging behaviours quantitatively in Section 7.4.2 by setting up a hierarchical learning experiment where the agent is allowed to reuse policies which are sampled from different points of a curious exploration timeline. We show that by combining emerging behaviour, the agent is able to learn a new downstream manipulation task as quickly as in a case where the learning is supported by a curriculum of reward functions which are hand-designed to learn that specific task. This is a promising signal suggesting that behaviours discovered from curious exploration of the environment's dynamics can be used to bootstrap the learning of complex manipulation tasks.

On top of its immediate contribution to structured exploration and hierarchical reinforcement learning, the approach presented in Chapter 7 also closes the loop between physical intution and robotic manipulation. It presents evidence that the process of learning an approximate model of (rigid-body) dynamics leads to the emergence of complex behaviour as the agent interacts with the environment in a targeted way to collect relevant data. This process ultimatley yields two learned components: a world model, which represents an intuition about the environment's dynamics, and different instances of a control policy which capture diverse behaviour which has emerged during the exploration process. Both components can be leveraged to improve robotic manipulation downstream – either in model predictive control using the learned physical intuition or in model-free control using an instance of the control policy.

1.2 Publications

This thesis is based on material from the following publications in order of their appearance in Chapters 3 to 7:

1. O. Groth, F. B. Fuchs, I. Posner, and A. Vedaldi. "ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking". In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018

2. Y. Wu, S. Kasewa, O. Groth, S. Salter, L. Sun, O. Parker Jones, and I. Posner. "Learning Affordances in Object-Centric Generative Models". In: *Workshop on Object-Oriented Learning at ICML 2020* (July 2020)

3. S. Ehrhardt, O. Groth, A. Monszpart, M. Engelcke, I. Posner, N. J. Mitra, and A. Vedaldi. "RELATE: Physically Plausible Multi-Object Scene Synthesis Using Structured Latent Spaces". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Dec. 2020

4. O. Groth, C.-M. Hung, A. Vedaldi, and I. Posner. "Goal-Conditioned End-to-End Visuomotor Control for Versatile Skill Primitives". In: *IEEE International Conference on Robotics and Automation (ICRA)*. June 2021

5. O. Groth, M. Wulfmeier, G. Vezzani, V. Dasagi, T. Hertweck, R. Hafner, N. Heess, and M. Riedmiller. "Is Curiosity All You Need? On the Utility of Emergent Behaviours from Curious Exploration". In: *arXiv preprint arXiv:2109.08603* (Sept. 2021).

In cases of shared lead authorship, a 'statement of authorship' declaration has been added at the end of the respective chapter listing the individual contributions.

2
Literature Review

In this chapter we present an overview over the literature related to the central investigation of this thesis: the learning of physical intuition for robotic manipulation. As presented in Fig. 2.1, this research sits at the intersection of Cognitive Science, Machine Learning and Robotics. In the following sections, we will provide an overview over the relevant literature from each field pertaining to the nature, acquisition and application of physical intuition.

We start by elaborating on the core concepts of *physical intuition*, *analysis-by-synthesis* and *object affordances* which are motivated by research in human cognition in Section 2.1. Next, we outline in Section 2.2 how these concepts can be implemented using Machine Learning methodology such as *neural physics models*, *object-centric representations* and *structured exploration*. Finally, we conclude this chapter in Section 2.3 by presenting core challenges and applications in Robotics such as *scene rearrangement* and *visuomotor control* which can substantially benefit from learned models of physical intuition.

Since the concept of physical intuition is only vaguely defined in the related literature, we provide a definition in the context of this thesis below. We acknowledge that this definition is constrained to the scope of this thesis, i. e. it only covers aspects

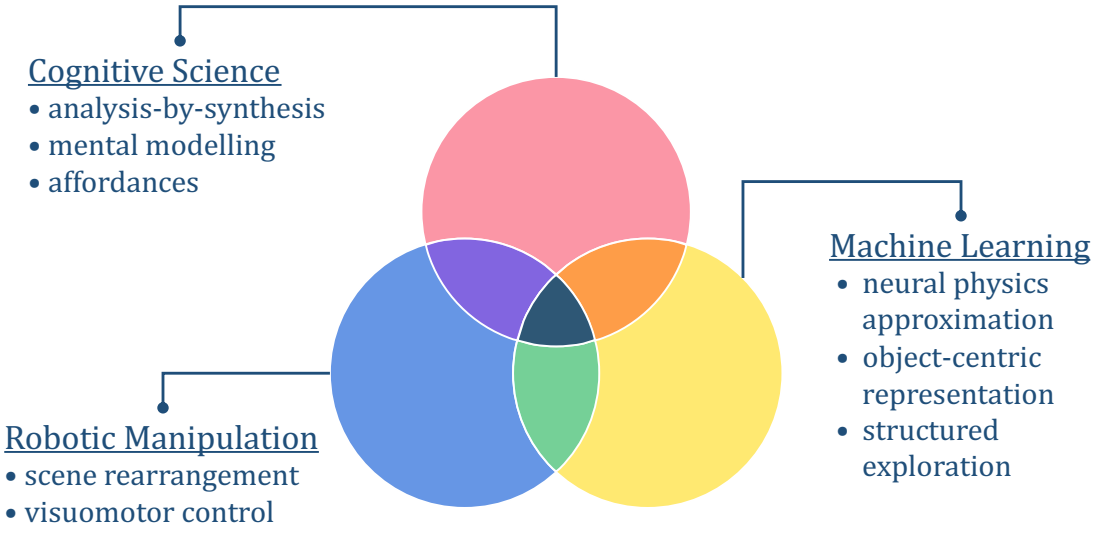


Figure 2.1: An overview over the fields of research related to the investigation in this thesis. The problem of physical intuition in manipulation sits at the intersection of concepts from Cognitive Science, methodology from Machine Learning and applications in Robotics.

of intuition pertaining to the physics of rigid objects and omits phenomena like deformation. However, we believe that the study of physical intuition in the concrete context of rigid-body motion does not invalidate the generality of the approach, which is also extensible into other physical domains.

Physical Intuition

Within the scope of this thesis, we define physical intuition as follows:

- Physical intuition is an approximate prediction about the evolution or outcome of a physical process.
- Physical intuition is learned from data or experience.
- Physical intuition is divided into two categories:
  (i) low-level intuition, which pertains to predictions about rigid-body motion, typically over short time horizons, and
  (ii) high-level intuition, which pertains to qualitative outcomes of physical processes, potentially spanning long time-horizons.

2.1 Physical Intuition in Human Cognition

The subject of human physical intuition, i. e. the approximate inference of physical properties and processes in our environment, is well studied in Cognitive Science.

Humans display their physical intuition in a wide variety of situations, e. g. when assessing whether an object is about to fall over, determining the applicability of a tool for a certain task or extrapolating the trajectories of moving objects. Kubricht et al. [112] present an overview over common intuitions and misconceptions exhibited by humans when assessing physical phenomena, e. g. when predicting the trajectories of colliding balls, swinging pendulums or falling objects. It is interesting that human predictions of object trajectories sometimes carry semblances of Aristotelian physics (Aristotle [7]), e. g. by exhibiting 'straight down beliefs' about object trajectories, rather than being informed by the more recent and accurate laws of motion postulated by Newton [148]. However, despite the apparent inaccuracies, the quality of those predictions is still sufficient to support our superior object manipulation skills.

P. W. Battaglia et al. [13] and J. Hamrick et al. [72] conjecture that humans possess an 'internal physics engine' which allows them to perform 'mental simulations' of physical events, e. g. extrapolating motion trajectories. An investigation into the functional neuroanatomy of intuitive physical inference (Fischer et al. [48]) indicates a connection between the visual stimuli of an observed scene and the multiple demand system of our brain which is responsible for action planning and tool use. This suggests that humans acquire their intuition in a very task-driven way learning approximations of object geometry and physics which are just accurate enough to carry out the tasks they are interested in. They possibly even compromise on exact physical prediction if the accuracy gain does not justify the additional computational burden and response latency – a trade-off between *reacting* and *reasoning* inherent to many human decision making processes, also known as 'thinking fast and slow' (Kahneman [94]). Closely connected to the notion of a mental model of physics which also facilitates the efficient processing of perceptions is the concept of *physical object representations* (PORs) (Yildirim et al. [227]). PORs provide a model which factors a perceived scene into a set of objects with each object being described by its physical properties such as texture, shape, mass, position and velocity. According to this formulation the perception of a scene follows the *analysis-by-synthesis*

principle (Rock [164]): The PORs are the building blocks from which the internal physics engine synthesises a perception. Their parameters (e. g. position and velocity) which represent our belief about the object states are continuously updated such that the synthesised perception stays in accordance with the sensations, e. g. the visual or haptic signals received. If the POR-hypothesis holds, physical intuition, i. e. the data-driven approximation of physics, is foundational to our perception.

Despite exhibiting inaccuracies, the capacity provided by this 'internal physics simulator' enables us to reason even about ill-posed or impossible physical problems such as inferring the relative mass of an object from pure visual perception (J. B. Hamrick et al. [74]) or reversing time to infer how a scene came about (Yildirim et al. [226]). Another remarkable experiment by Yildirim et al. [226] shows that humans can even envision how object stacks have been built by another human. This highlights a strong connection between our physical intuition and our embodied interaction with the environment – likely due to the fact that most 'data' used to train out intuition is acquired by manipulating objects in our environment. The concept of *affordances* (Gibson [54]) is another important link between physical intuition and object manipulation. Physical properties of objects such as the length of a stick or the weight of a hammer's head *afford* certain interactions with the environment which extend our own capabilties, e. g. by using a stick to reach an otherwise unreachable object or by using a hammer to hit a nail. While being an application of an intuitive understanding of physics[1], the use of objects as tools simultaneously lifts the learning of physical intuition into a more abstract space on a higher-level. High-level physical intuitions – as opposed to low-level ones such as approximating motion dynamics – can pertain to qualitative assessments such as: 'Can this object support another one?' or 'Is this stick long enough to reach my goal?' Nevertheless, these more abstract aspects of human physical intuition can also be modelled as marginalisations over the distribution of an underlying 'mental physics engine' and can therefore also be acquired in a data-driven way (P. W. Battaglia et al. [13]).

[1] In cognitive science, it is debated whether the mere use of tools can already be considered a hallmark of intelligence or whether only the process of manufacturing tools constitutes intelligent behaviour, see e. g. Ambrose [5], Beck et al. [15], and Emery and Clayton [39]

2.2 Machine Learning Methods for the Acquisition of Physical Intuition

After outlining the cognitive concepts underpinning our research in Section 2.1, we dedicate this section to give an overview over core methodologies in Machine Learning which can be used to implement the various aspects of physical intuition with a computer[2]. We start with the two variants of physical intuition defined in Chapter 2: low- and high-level intuition. As low-level intuition pertains to the inter- and extrapolation of rigid-body trajectories, it can be naturally cast as a regression task in a position- or velocity space. In contrast, high-level intuition is concerned with the *outcome* of a physical event, e. g. whether a stack of objects falls over or whether a certain task can be solved with a specific tool. Therefore, categorical classification is typically well-suited to model high-level physical intuitions.

Since the estimation of physical parameters and processes from raw data is an important aspect of scene understanding, many recent investigations involving deep learning have been approached from a Computer Vision angle. Mottaghi et al. [142, 143] were among the first to propose a 'Newtonian image understanding' mapping images to a set of pre-defined scenarios of rigid-body motion to anticipate the dynamical evolution of the scene. In a similar vein, Standley et al. [185] leverage web-mined knowledge about shape and mass of common household objects to infer masses from still images. Evidently, the inference of concrete physical parameters from still images is a very ill-posed problem as many important properties such as mass, velocity or friction cannot be determined from appearance alone without a heavy reliance on prior data or possible physical scenarios. Therefore, the focus of physical learning in Computer Vision quickly shifted to video data as this modality allows to observe an object's motion and interaction with the environment (e. g. colliding with another object) which affords more precise inference of quantities like (relative) mass, velocity and friction directly from the data. Finn et al. [45] capture

[2]While we focus in this section on methods specific to the implementation of physical intuition as defined in Chapter 2, J. B. Hamrick [73] provide a comprehensive survey on deep learning analogues of mental simulation and imagination more broadly.

video dynamics in a frame-to-frame transformation function in RGB space which is learned from raw videos. However, this holistic, i. e. full-image-encompassing, dynamics modelling only lends itself to short-horizon video prediction of a few frames into the future but cannot be used to predict precise object trajectories or object interactions. In contrast, Fragkiadaki et al. [50] focus more closely on object-centric modelling of physics by learning to predict the displacement of object glimpses which is better suited for longer prediction horizons. J. Wu et al. [207, 210] go even further by inferring concrete quantities like friction and velocity from videos and map them onto exact physical equations given to the system a priori. While such approaches afford even better predictive capabilties since the extrapolation is based on exact physical knowledge, it would already sit on the edge of our definition of a low-level physical intuition since the physical principles are given and only the initialisation from raw observations is learned.

Concurrently to the development of low-level models for visual dynamics, the discriminative capacity of deep neural networks has also given rise to approaches which attempt to learn higher-level physical semantics in an end-to-end fashion. Given the historical popularity of *blocks worlds*[3] in Computer Vision and the mechanical reasoning tasks typically associated with them (cf. e. g. Gupta et al. [67], Jia et al. [91], and Zheng et al. [232]), those environments have become a popular testbed for high-level physical intuition as well. Lerer et al. [117] and Li et al. [122] pioneered the new task of *visual stability prediction*: Given the input of an image of a stack structure, a neural network predicts whether the stack is about to collapse or not. Reminiscent of the earlier blocks worlds, the structures investigated consisted of randomised cuboids. While these setups are sufficient to learn about the basic mechanics of stacking like the support of the cumulative centre of mass (cf. Section 3.3.2), other aspects like the influence of object geometry are not considered. With ShapeStacks (Groth et al. [64]) we contribute to this line of work by providing scenarios which feature a more diverse object set including cylinders and spheres and control for the stack construction procedure to study previously neglected aspects of visual stability

[3] A simple environment filled with cuboid geometric primitives.

prediction and evaluate its utility in a simulated construction task. Additionally, we analyse the correlations between the end-to-end learned stability prediction and the true mechanics governing the scenarios to identify the boundaries of the generalisation of the end-to-end approach. Our work has also been one of the first benchmarks for principled analysis of visual physical intuition – a line of work which has seen many extensions in 2D and 3D domains afterwards, e. g. by Allen et al. [4], Bakhtin et al. [10], Riochet et al. [163], and Yi et al. [224].

Another popular domain of high-level physical intuition has traditionally been the learning of object affordances. Similar to the visual stability prediction task, deep learning has contributed to recent end-to-end affordance learning approaches (cf. e. g. Do et al. [33] and Myers et al. [144]). Like in visual stability prediction, affordance prediction is typically cast as a classification problem with discrete labels such as `cut`, `scoop` or `pound` and the corresponding classifiers can be interpreted as a high-level intuition marginalising over the underlying physical processes to determine a more abstract outcome. In Chapter 4 we extend this line of work by presenting a twist on the traditional classification setup in affordance learning: Instead of predicting the affordance label directly, we train a classifier to predict an outcome on the task level, i. e. whether a given reaching task is *feasible* given a certain tool. In this approach, an object's affordance is represented as a manifold in the latent space of a classification task which connects it more closely to a concrete manipulation task and enables smooth interpolations between solutions.

While the previously discussed lines of work on low- and high-level physical intuition have been motivated by applications in Computer Vision and Robotics, a concurrent line of work in Machine Learning has proposed *neural physics engines*. P. Battaglia et al. [14] and M. B. Chang et al. [27] pioneered this new field by proposing learnable physics engines based on *Graph Neural Networks* (GNNs) (Z. Wu et al. [213]). These models can approximate the dynamics of many physical problems like the *n-body problem*, *rigid-body collision* or *spring relations* from raw observations. Once trained, these models can serve as simulation engines for accurate physical long-term predictions

aiming at closing the performance gap to hand-written physics engines. However, despite their power of modelling physics accurately, these models typically rely on exact measurement data (e. g. object positions) during training and cannot be readily adopted to operate on raw visual data. Watters et al. [204] propose an extension of interaction networks enabling the training on simplistic visual data like distinctly coloured dots moving on a 2D plane. Our model proposed in Chapter 5 extends prior art in learning physics from visual input. It borrows the GNN backbone as a core concept of a neural physics engine, but connects it with a more sophisticated vision architecture (Nguyen-Phuoc et al. [150]) enabling the training on realistic video footage.

Complementary to the research on neural physics engines which are based on graphs is the line of work dedicated to *object-centric representations*, predominantly inferred from visual input. Object-centric representations in Machine Learning can be seen as an implementation of the cognitive concept of PORs (cf. Section 2.1). Hence, they typically feature a structured latent space which factorises the representation of each object such that its appearance and position are encoded separately. Burgess et al. [25] propose a VAE-based architecture which represents each object in an image as two vectors encoding its texture and 2D image mask respectively. Nguyen-Phuoc et al. [150] add even more prior knowledge to their GAN architecture by treating the position vector of their latent space as an object's 3D centroid position. Such factorised latent spaces are also known as *disentangled representations* (Bengio et al. [17]) and afford the benefit of capturing and controlling different factors of variation in the data in an interpretable way, e. g. along one latent space dimension. With RELATE (Ehrhardt et al. [38]) we build upon the explicit object-centric priors introduced by Nguyen-Phuoc et al. [150] and extend their work by modelling inter-object relationships as well. In line with Van Steenkiste et al. [196], we find that the joint learning of object appearances and their relationships facilitates the discovery of independent objects and yields a disentangled, human-interpretable latent space.

As we have discussed in this section, neural physics engines implement many cognitive aspects of physical intuition such as PORs and a data-driven, low-level approximation of dyamics. As these models become increasingly reliable simulations which an agent can draw upon to imagine the evolution of a scene and plan actions, another connection to cognition becomes apparent: *mental modelling* (Forrester [49]). Ha and Schmidhuber [69] propose *world models* which can be seen as general-purpose predictors of the next perceived state of an agent's environment. Ha and Schmidhuber [68] and Sekar et al. [174] have demonstrated the utility of using world models for robust policy learning, leveraging their generative capabilities for data augmentation and planning. However, the true power of world modelling is conjectured to lie in the acquisition process of the model itself. Oudeyer and Kaplan [155] and Schmidhuber [171] propose predictive models about the environment dynamics as the backbone of a structured exploration of any environment. In this way, an interactive agent can be *intrinsically motivated* to collect data to improve the internal model about the environment and its transition dynamics – for instance by being rewarded for reducing the prediction error of its world model. Therefore, acquiring a physical intuition about environment dynamics in an interactive way can be seen as an instance of *intrinsically motivated learning*.

In the literature, multiple approaches have been proposed to implement intrinsic motivation. Most notably in that regard are the lines of work on *curiosity learning* (Schmidhuber [170]), *empowerment* (Klyubin et al. [105]), *inductive task proposal* (Schmidhuber [172]) and *asymmetric self-play* (Sukhbaatar et al. [189]). In particular, the method of curiosity learning has received considerable attention given its relative ease of implementation compared to other approaches (Burda et al. [23] and Pathak et al. [157]). At its core, curiosity learning extends the classic setup of reinforcement learning (RL) (Sutton and Barto [191]) with a non-stationary reward function which is derived from the world model. In this formulation the *curiosity reward* given to an agent at each state transition is scaled by the *prediction error* (Shelhamer et al. [178]) or *surprise* (Achiam and Sastry [2]) of its current world model. Once the reward has been given, the world model is updated such that the error will be lower the next time

this state transition is encountered. In that way, an agent is rewarded for exploring parts of its environment for which the transition dynamics are not well predictable, yet.[4] It has been shown that this technique leads to comprehensive exploration in many 2D and 3D domains (Burda et al. [23]) and yields robust dynamics models for action planning (Sekar et al. [174]). However, one aspect which has only received scant attention in the literature so far is the behaviour which emerges *during* curious exploration. In Chapter 7 we contribute to the estalished research of curiosity learning by studying the emergence of embodied behaviour and its utility using an off-policy curiosity method. We look at curiosity-based exploration through the lens of *continual learning* (Ring [162]): We show that complex, human-interpretable behaviour emerges and vanishes in a curiosity setup and the self-discovered skills can be harnessed for effective downstream learning via modular composition.

2.3 Robotics as Testbed and Application of Physical Intuition

Due to the mutual interdependence between physical intuition and interaction with the environment which has been outlined in the previous Sections 2.1 and 2.2, robotic manipulation provides an ideal experimentation and application domain for the research of this thesis. Object manipulation has a long history in the field of robotics. First applications were developed around the manipulation of dangerous objects, e. g. handling nuclear materials using teleoperated manipulators (Goertz [56] and Goertz and Thompson [57]). Teleoperated, human-assisted manipulation remains an important area of robotics research, especially in safety-critical, high-stakes domains such as bomb disposal (e. g. Lisle [125]) or remote manipulation on space vehicles (e. g. Yoon et al. [228]). However, driven by new demands such as warehouse automation and indoor service robots, *autonomous* robotic manipulation

[4]For simplicity, we omit cases in which certain parts on an environment are inherently stochastic and not controllable by an agent's actions. This so called 'white noise problem' (Schmidhuber [171]) breaks any curiosity-driven exploration, if not accounted for. However, several works have proposed models to mitigate this problem, e. g. by employing inverse dynamics models whose latent space learns to disregard uncontrollable factors of environment variations, cf. e. g. Agrawal et al. [3] and Pathak et al. [157].

has emerged as an important research area on its own. Despite lower stakes in terms of safety considerations, these indoor environments pose their own set of challenges when it comes to manipulating objects within them. Most notably they feature unknown, unstructured scenes with a variety of diverse objects with unknown physical properties and the nature of tasks – typically a version of automated object rearrangement – precludes extensive human operator intervention for economical reasons. Hence, the acquisition of knowledge about physical properties of the environment is critical for the success of any manipulation task in such environments. Therefore, Batra et al. [12] have recently formalised this task of *scene rearrangement* in indoor environments as a benchmark for embodied intelligence as it poses a significant challenge to any contemporary robotic system.

The conventional approach to autonomous robotic task execution follows the paradigm of *'perceive – plan – act'* which has been foundational since the days of the first autonomous robots such as Shakey from Stanford Research Institutes (Nilson [151]). Based on this paradigm, the traditional system pipeline for robotic manipulation typically consists of modules for visual observation and some form of 3D state representation, the recognition of objects and their poses, the planning of the task execution and its corresponding motion and the execution via a dedicated control algorithm. Each component in this traditional pipeline is in itself a mature field of research and we can only provide examples of representative work here. Implementations of the first component, visual perception and 3D state representation, are typically based on methods running a form of 'synchronous localisation and mapping' (SLAM) (Smith et al. [182]). The next step of object and pose recognition is a long-studied problem in Computer Vision and features a rich body of literature spanning template-matching (e. g. Hinterstoisser et al. [80]), sparse keypoint-based (e. g. Collet et al. [29]) and dense pixel-based (e. g. Brachmann et al. [20]) methods. Once objects of interest have been recognised, the stage of planning a grasp to pick them up can be an extremely challenging engineering problem in itself depending on the kind of robotic manipulator . Bohg et al. [19] present an overview over recent, data-driven grasp pose planners for a wide variety of commonly used

grippers. However, due to the planning complexity of sophisticated grippers like the antropomorphic Shadow Hand (ShadowRobot [175]), most robotic manipulation scenarios to date typically involve much simpler grippers featuring parallel jaws or even suction tubes (Hernandez et al. [76]). Following the pick-up-phase, a traditional pipeline typically invokes some form of motion planning to find a collision-free path from the pick-up to the target location of an object. Lozano-Pérez and Wesley [132] have introduced the *configuration space formulation* for this problem which serves as the backbone of most sampling-based motion planners (Kavraki et al. [97]) to date. Finally, the motion plans are executed using classical control methods which achieve the desired configurations in the robot's joint angle or velocity space by applying suitable torques to the joint motors.

With the advent of deep learning many modules of this traditional pipeline have been implemented as deep neural networks – e. g. visual object detectors, grasp pose regressors or control policies – which are trained on data instead of being handcrafted. However, simply replacing hand-designed modules with their learned counterparts does not leverage the full potential afforded by deep learning. S. L. James [89] argues that new approaches to robotic manipulation should strive to combine the best of both worlds by borrowing the idea of modular design from the traditional pipeline and combining it with the ability of end-to-end optimisation. Such *tightly-coupled manipulation pipelines* (TMPs) exhibit an interpretable and maintainable system design with individual modules which can be trained and debugged independently but also afford coupling all learnable components in a differentiable manner allowing task information to propagate through the entire system for joint optimisation. Additionally, TMPs can result in a new division of labour between the individual modules which does not necessarily resemble the one in the traditional pipeline. For instance, parts of the 3D state representation and object detection modules can be fused into a novel attention module selecting relevant portions of information directly from the visual input stream and passing it on to the grasp pose prediction which is in turn fed into a control policy conditioned on a target pose (S. James and A. Davison [87]). In the context of this emerging paradigm of TMPs, physical intution

can be seen as a new module which learns to approximate physical properties of the environment, e. g. the dynamics of rigid objects, from the interaction data gathered by the robot.

Within the scope of this thesis we specifically focus on manipulation tasks involving the rearrangement of rigid objects in tabletop scenarios. Our experimental setups typically feature a (multi-view) RGB camera stream and a single-arm robot manipulator with a two- or three-point gripper. Consequently, the most immediate application of our results lies within the tabletop rearrangement domain of robotic manipulation. However, in the following paragraphs we will also draw connections between the acquisition and application of physical intuition and broader problem areas in robotics such as scene representation, planning and visuomotor control.

An important aspect of generative physical intuition is its imaginative capability which can benefit robotic manipulation in multiple ways. Firstly, as we have discussed in Section 2.2, recent generative models of (visual) scenes are typically *object-centric*. Naturally, such representations lend themselves easily to task planning methods in robotics as the current and the future state of a scene can be mapped into a representation with explicitly factored object positions. This enables planning rearrangement tasks in an abstract space where for instance manipulation primitives (e. g. a push or pick-and-place action) can be employed to transform object positions into the desired configuration, e. g. as done in King et al. [100]. Secondly, the ability to imagine a future scene state is complementary to a long-standing problem in robotics research: the learning of universal, goal-conditioned policies (Kaelbling [93] and Schaul et al. [169]). Control policies which can be conditioned on a certain goal state, e. g. defining a target configuration with altered object locations, are evidently more general and robust than any given policy which has only been optimised for a single reward function like 'pick up the red cube and put it into the blue bin'. Furthermore, goal-conditioning also alleviates the need to define task-specific reward functions in advance of training the system since all tasks are specified by their respective goal states. While defining a set of suitable goals is also a challenging

requirement, generative models for possible future scene states greatly faciliate this process as they allow for a procedural generation of goal states within a certain data distribution (cf. e. g. A. V. Nair et al. [146]). Another tantalising prospect of scene state imagination for policy training is the aspect of noise control. Ha and Schmidhuber [69] investigate visuomotor control policies learned completely in the imagined space of a world model. Their experiments demonstrate that by controlling for the amount of uncertainty in the imagined scene states during policy training, the resulting policy exhibits an increased robustness compared to when trained on only 'clean' observations.

While the potential of imagination for robotic learning has only recently been embraced due to the availability of powerful deep generative models, methods pertaining to high-level intuitions about geometric affordances and structural stability have been foundational to many robotic applications for a long time – for a survey see e. g. Yamanobe et al. [220]. The inference of affordances is particularly prevalent in robotic tool use. For instance, Z. Liu et al. [128] and Myers et al. [144] segment tools into different parts as different geometries afford different capabilities, e. g. thin edges afford cutting while solid bodies afford pounding. Other applications of affordance prediction are the efficient transfer of manipulation policies to novel objects based on common object affordances (Kjellström et al. [104]) or the anticipation of human object interactions in assistive robotic contexts (Koppula and Saxena [108]). Our approach presented in Chapter 4 presents a novel perspective in that regard representing an affordance as a latent space manifold instead of a discrete label enabling applications in inventive tool use or even robotic tool manufacturing.

Similar to affordances, the prediction of structural stability has been integrated into many robotic applications related to manipulation and construction. The classic block stacking game of Jenga has served as a testbed for manipulation methods which use visual (J. Wang et al. [200]) or force feedback (Kimura et al. [99]) to estimate the stability of the tower and plan safe extraction moves with a robotic gripper. In a similar vein, Ornan and Degani [153] estimate contact normals of piled objects

to plan the piece-wise removal of objects while only minimally perturbing the pile. Conversely, stability prediction also facilitates the stable placement of objects. For instance, Furrer et al. [53] use scanned point clouds of stones to compute stable stack configurations and Pashevich et al. [156] reverse knowledge obtained from a simulated disassembly to learn a policy for building objects into a target shape. While many of these examples have been shown to accomplish challenging real world manipulation tasks, they typically rely on exact domain knowledge, e. g. the structure and forces within a Jenga tower, to compute feasible plans. Our work on intuitive stability prediction in Chapter 3 significantly simplifies the model complexity in such stacking scenarios while also being trainable on only weakly labeled data.

While the aspects of physical intuition investigated in Chapters 3 to 5 cover the spectrum from high-level stability and affordance prediction to low-level dynamics modelling, their common target application is in *visuomotor control* (VMC) in object manipulation. Naturally, the robotics literature boasts a mature body of work on VMC in many different application areas but staying close to the thread of the aforementioned scene rearrangement task, we restrict the scope of this literature overview specifically to VMC methods pertaining to the manipulation of rigid objects. With the widespread adoption of deep learning models for robotic control, it has become attractive to learn entire VMC systems *end-to-end*, which previously consisted of pipelines of dedicated perception, planning and control components. End-to-end training can be conducted using deep reinforcement learning, e. g. deep Q-learning (Mnih et al. [139]), or using provided task demonstrations to either guide the policy search (Levine and Koltun [119]) or cast the training as a supervised imitation learning problem (Levine et al. [118]). While these new approaches to VMC reduce the perceived system complexity compared to a traditional manipulation pipeline, their advantage comes at the cost of requiring task-specific demonstrations or reward functions. Therefore, many end-to-end VMC systems are only capable of executing a single task or a narrow range of tasks as each additional task requires a new reward function or demonstration data which can be difficult to obtain.

In order to overcome this limitation, goal-conditioned VMC policies are needed which can be applied to a new task via the provision of a desired goal state. Goal-conditioned control policies play a crucial role in the manipulation pipeline since they provide the necessary interface to execute a desired goal from a previous planning stage. Typical approaches for goal-conditioned control are *model-predictive control* (MPC), *few-shot imitation* or *meta learning* (Finn et al. [44]). However, each of those paradigms exhibit shortcomings which prevent them from true zero-shot generalisation to a novel task. Few-shot imitation or meta-learning approaches typically require at least one demonstration of the new target task to adapt their manipulation policy either via network modulation (S. James et al. [86]) or finetuning (Finn et al. [47]). This can be difficult as in some occasions, demonstrations are hard to obtain or model finetuning on the deployment hardware can incur a substantial computational burden. In contrast, visual manipulation using MPC can be conditioned on a single goal image. Yet, the limiting factor of model-predictive VMC lies within the model itself which is often implemented as an action-conditioned video predictor. The uncertainty of predicting the next visual observation in addition to the compounding error when unrolling over longer time horizons often limits MPC-based approaches to simple reaching or pushing applications like in Ebert et al. [35] and S. Nair and Finn [147]. Furthermore, it remains an open challenge how the distance between an observation and a given target image can be sensibly measured in pixel space. GEECO, our VMC architecture presented in Chapter 6, addresses the aforementioned shortcomings of goal-conditioned VMC by leveraging *dynamic images* (Bilen et al. [18]) as an approximate representation of visual dynamics. Specifically, the dynamic image representation facilitates task inference and zero-shot transfer based on a single goal image without the need of any finetuning one a new task first.

Reviewing the different aspects of manipulation which are affected by data-driven physical intuition, i. e. scene representation, planning and visuomotor control, it becomes apparent that the work in this thesis cannot be placed at any specific place in the traditional manipulation pipeline. However, in the context of the novel tightly-coupled manipulation pipelines (S. L. James [89]), a physical intuition module can

interact with other learnable modules in many beneficial ways. For instance, the forward rollouts of a dynamics model can be leveraged directly for model-predictive control or distilled into the Q-function of a policy informing about the outcome of a trajectory. Furthermore, the physical predictions could even be used to correct the object perception earlier in the pipeline by confirming or rejecting detection hypotheses based on the spatio-temporal consistency of their trajectories under the physics module. Lastly, the predictive capabilities of a physics module can feed forward in the pipeline supplying a conditionable controller with a desired target state of the scene. In conclusion, a module for physical intuition can be seen as an augmentation of a differentiable manipulation pipeline which supports and corrects the predictions of other connected modules based on the physical rules inferred from the environment.

3

Shapestacks: Learning Vision-based Physical Intuition for Generalised Object Stacking

In this chapter we establish ShapeStacks, a simulation environment and dataset to train and evaluate vision-based stability predictors on the example of towers comprising of different wooden shapes. We demonstrate that a visual classifier trained on this dataset performs commensurately even with much more complex models for stability prediction. Furthermore, we show that the classifier learns plausible correlations between visual inputs and physical phenomena like the source of a structural instability without additional annotation. Lastly, we demonstrate that such a classifier for visual stability prediction can successfully control manipulation processes such as object stacking and placing of counter-weights. This work is published as:

O. Groth, F. B. Fuchs, I. Posner, and A. Vedaldi. "ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking". In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018

Abstract

Physical intuition is pivotal for intelligent agents to perform complex tasks. In this paper we investigate the passive acquisition of an intuitive understanding of physical principles as well as the active utilisation of this intuition in the context of generalised object stacking. To this end, we provide ShapeStacks[1]: a simulation-based dataset featuring 20,000 stack configurations composed of a variety of elementary geometric primitives richly annotated regarding semantics and structural stability. We train visual classifiers for binary stability prediction on the ShapeStacks data and scrutinise their learned physical intuition. Due to the richness of the training data our approach also generalises favourably to real-world scenarios achieving state-of-the-art stability prediction on a publicly available benchmark of block towers. We then leverage the physical intuition learned by our model to actively construct stable stacks and observe the emergence of an intuitive notion of *stackability* - an inherent object affordance - induced by the active stacking task. Our approach performs well exceeding the stack height observed during training and even manages to counterbalance initially unstable structures.

3.1 Introduction

Research in cognitive science (J. B. Hamrick et al. [74] and Kubricht et al. [112]) highlights how the ability of humans to manipulate the environment depends strongly on our ability to intuitively understand its physics from visual observations. Intuitive physics may be just as important for autonomous agents to effectively and efficiently perform complex tasks such as object stacking or (dis-)assembly - and even the creation and use of tools. Central to these deliberations is an understanding of the physical properties of objects in the context of how they are meant to be used. Such object *affordances* are typically pre-defined given knowledge of the task at hand (Kjellström et al. [104] and Koppula and Saxena [108]). In contrast, we posit

[1]Source code & data are available at `https://ogroth.github.io/shapestacks/`

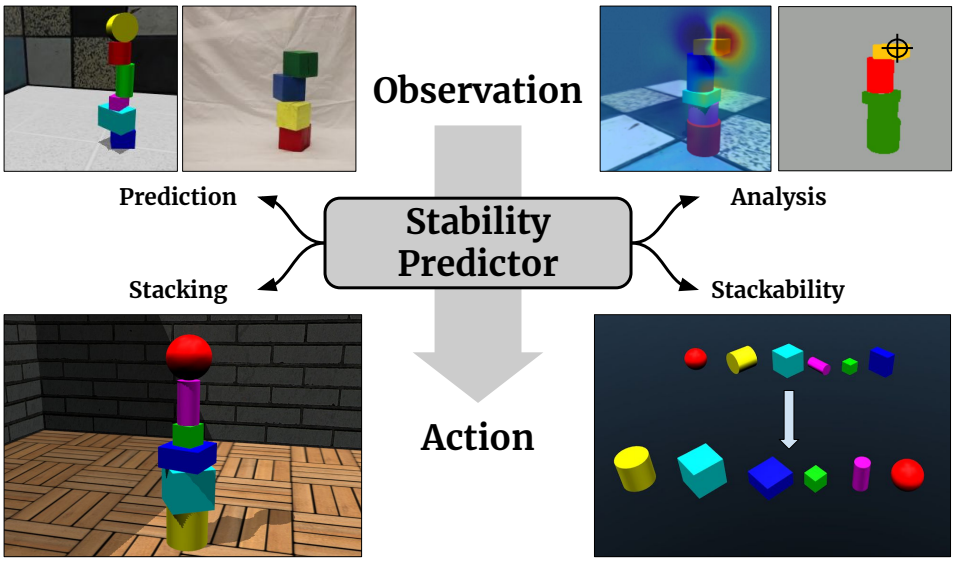


Figure 3.1: We present a visual classifier which is trained on stacks of diverse shapes to distinguish between stable and unstable structures. We demonstrate that the implicit knowledge captured by the predictor can be utilised to detect structural instabilities, infer the *stackability* (utility with regard to stacking) of objects and guide a simulated stacking process solely from visual cues.

that relevant affordances do not need to be specified a priori but can be learned in a task-driven manner.

Inspired by recent work in computer vision (Lerer et al. [117] and J. Wu et al. [208]) and robotics (Furrer et al. [53], Li et al. [122], and Zhu et al. [233]) we consider the task of *object stacking* and the problem of learning – from passive visual observations – its intuitive physical principles. By leveraging the model's acquired intuitions, we are able to utilise the passive observation in an active manipulation task as outlined in Figure 3.1, which sets us apart from prior art in both scope and reach.

Firstly, we argue that in order for agents to perform complex tasks they need to be able to interact with a variety of different object types. We therefore investigate the stacking problem using a broader set of geometric primitives than found in related works. To this end we introduce ShapeStacks, a simulation-based dataset specifically created to enable exploration of stackability of a variety of objects. Furthermore, ShapeStacks is, to the best of our knowledge, the first such dataset with annotations of the mechanical points of failure of stacks, which are inferred by formally analysing

the underlying physics. This makes ShapeStacks the most rigorous and complete publicly available dataset in this space.

Secondly, based on the ShapeStacks dataset, we extend the investigation of stability prediction presented in Lerer et al. [117] and J. Wu et al. [208] to include stacks containing multiple object geometries. This allows for a more rigorous qualitative and quantitative evaluation of system performance. For example, our work, for the first time, quantifies if a model trained for stability prediction correctly localizes the underlying stability violations. We demonstrate that our model based on ShapeStacks outperforms the baseline by Lerer et al. [117] and performs commensurately with the current state-of-the-art (J. Wu et al. [208]) on real-world image data without requiring a physics engine during test time.

Lastly, in order to investigate our main hypothesis – namely that meaningful affordances emerge from representations learned by performing concrete tasks – our work goes beyond the passive assessment of stacked towers as stable or unstable and actively performs stacking. In particular, we argue that, through the passive task of stability prediction, our system implicitly learns to assess the *stackability* of the individual object geometries involved. We demonstrate this by extracting a stackability score for different block geometries and by using it to prioritise piece selection in the construction of tall stacks. By inserting noise in the actual stacking process in lieu of disturbances present in real agents (e. g. motor and perception noise as well as contact physics) we demonstrate that a more intuitive notion of object *stackability* emerges.

As a result, our approach discovers an object's suitability towards stacking, ranks pieces accordingly and successfully builds stable towers. In addition, we show that our model is able to stabilise previously unstable structures by the addition of counterweights, arguably by developing an intuitive understanding of *counterbalancing*.

3.2 Related Work

The idea of vision-based physical intuition is firmly rooted in cognitive science where it is a long standing subject of investigation (Kubricht et al. [112]). Humans are very apt at predicting structural stability (P. W. Battaglia et al. [13]), inferring relative masses (J. B. Hamrick et al. [74]) and extrapolating trajectories of moving objects (Kubricht et al. [112]). Although the exact workings of human physical intuition remain elusive, it has recently gained increasing traction in the machine learning, computer vision and robotics communities. The combination of powerful deep learning models and physics simulators yielded encouraging results in predicting the movement of objects on inclined surfaces (J. Wu et al. [210]) and the dynamics of ball collision (P. Battaglia et al. [14], M. B. Chang et al. [27], and Fragkiadaki et al. [50]).

While some prior work on intuitive physics assumed direct access to physical parameters, such as position and velocity, several authors have considered learning physics from visual observations instead. Examples include reasoning about support relations (Gupta et al. [67] and Jia et al. [91]) and their geometric affordances and inferring forces in *Newtonian* image understanding (Mottaghi et al. [142]). Our aim is similar in that we learn the affordance of *stackability* – an object's utility towards stacking – from visual observation. Importantly, however, in our work affordances are not specified *a priori*, but emerge by passively predicting the stability of object stacks.

The latter is related to several recent works in stability prediction. Lerer et al. [117] pioneered the area by demonstrating feed-forward stability prediction of stacks from simulated and real images, releasing a collection of the latter as a public benchmark. J. Wu et al. [208] proposed more sophisticated predictors based on re-rendering an observed scene and using a physics engine to compute stability, outperforming Lerer et al. [117] on their real-world data. In contrast, our approach achieves performance commensurate to J. Wu et al. [208] while using only efficient feed forward prediction as in Lerer et al. [117].

The problem of structural stability is also well studied in the robotics community, especially in the context of manipulation tasks. Early work implements rule-based approaches with rudimentary visual perception for the game of Jenga (J. Wang et al. [200]) or the safe deconstruction of object piles (Ornan and Degani [153]). More recently, advances in 3D perception and physical simulation have been exploited to stack irregular objects like stones (Furrer et al. [53]).

The experimental setup of Li et al. [121, 122] is related to ours in that a stability predictor is trained for Kappla blocks in simulation which is then applied to guide stacking with a robotic arm. Our work is set apart from them in that we are considering a variety of object geometries as well as more challenging stack configurations. Furthermore, Li et al. [121, 122] do not consider object affordances.

More recently, Zhu et al. [233] show that an end-to-end approach with an end-effector in the loop can be used to learn visuo-motor skills sufficient to stack two blocks on top of one another – both in simulation and in the real world. Their work can be seen as complementary to ours, focusing on the end-effector actuation during stacking while we concentrate on the visual feedback loop and the emerging object affordances.

3.3 The ShapeStacks Dataset

In this section we describe the ShapeStacks dataset, starting from an overview of its contents (Section 3.3.1) followed by an analysis of the physics of stacking (Section 3.3.2). The latter is required to explain the design of ShapeStacks as well as to precisely define some of its physical data annotations. The full dataset including simulation descriptions and data generation scripts is publicly available.

3.3.1 Dataset Content

ShapeStacks is a large collection of 20,000 simulated block-stacking scenarios. The selection of the scenarios emphasises diversity by featuring multiple geometries,

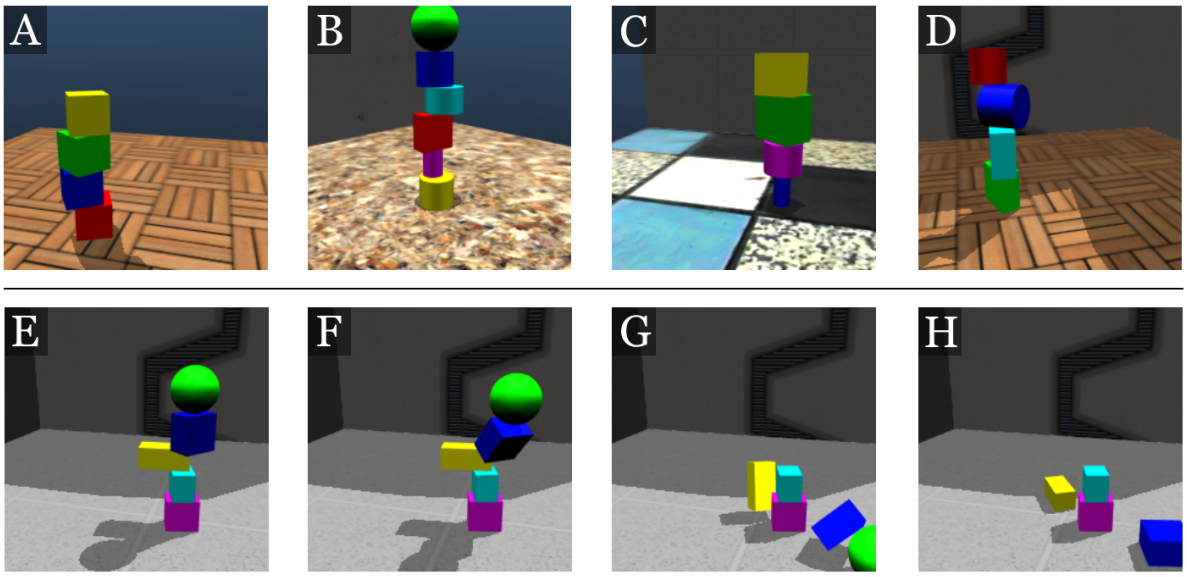
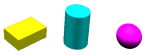


Figure 3.2: Different scenarios from the ShapeStacks data set. (A) - (D) depict initial stack setups: (A) stable, rectified tower of cubes, (B) stable tower where multiple objects counterbalance each other; some recorded images are cropped purposefully to include the difficulty of partial observability, (C) stable, but visually challenging scenario due to colours and textures, (D) violation of planar-surface-principle (VPSF). (E) - (H) show the simulation of an unstable, collapsing tower due to a centre of mass violation (VCOM).

degrees of structural complexity and types of structural stability violations, as shown in Figure 3.2.

A detailed summary of the dataset content is provided in Table 3.1. Each scenario is a single-stranded stack of cubes, cuboids, cylinders and spheres, all with varying dimensions, proportions and colours. The 20,000 scenarios are split roughly evenly among scenarios that contain only cubes (for comparing to related work on stability prediction (Lerer et al. [117] and J. Wu et al. [208]), and scenarios containing cuboids, cylinders and spheres (abbrev. CCS). Stacks have variable heights, from two to six objects, with the majority built up to a height of three. Each scenario can either be stable or unstable. This is determined by running a physics simulation with the given scenario as starting condition[2]. For every stack height, we provide an equal amount of stable and unstable scenarios. Furthermore, unstable scenarios are evenly divided into the two different instability types (cf. Section 3.3.2).

[2]We only report and release scenarios where the simulation outcome aligns with the physical derivation. Scenarios which behave differently due to imprecisions of the simulator are discarded.

	 **CCS** (# Scenarios)			 **Cubes** (# Scenarios)		
Stack height	Train	Val	Test	Train	Val	Test
$h = 2$	1,340	286	286	1,680	360	360
$h = 3$	2,464	528	528	1,680	360	360
$h = 4$	1,716	368	368	1,558	332	332
$h = 5$	678	144	144	1,274	272	272
$h = 6$	194	40	40	1,030	220	220
# Scenarios	6,392	1,366	1,366	7,222	1,544	1,544
# Images	102,272	21,856	21,856	115,552	24,704	24,704

Rendering & Annotation
Rendering
✓ 224×224 RGB
Randomised Scenes
✓ 25 Background Textures
✓ 6 Object Colours
✓ 5 Lighting Conditions
Annotation
✓ 0/1 Stability
✓ VCOM & VPSF
✓ Scene Semantics

Table 3.1: ShapeStacks contents. On the left, we present the number of scenarios and recorded images in both subsets of the dataset. *CCS* consists of cuboids, cylinders and spheres of varying size while *Cubes* only features regular blocks. On the right, we report the rendering and annotation details. See Section 3.3.2 for the derivation of the stability violation types *VCOM* and *VPSF*.

Scenarios are split into train (~ 70%), validation (~ 15%), and test (~ 15%) sets. Each scenario is rendered with a randomised set of background textures, object colours and lighting conditions. We record every scenario from 16 different camera angles and save RGB images of a resolution of 224 x 224 pixels.

Every recorded image carries a binary stability label. Also, every image is aligned with a segmentation map relating the different parts of the image to their semantics with regard to stability. The segmentation map annotates the object which violates the stability of the tower, the first object to fall during the collapse and the base and top of the tower.

3.3.2 The Mechanics of Stacking

While our goal is to study intuitive physics and the emergence of object affordances, we argue that a precise understanding of the physical properties of the scenarios is essential to control data generation as well as to evaluate models. In this paper, we restrict our attention to *single-stranded stacks*: each object S rests on top of another object S' or the ground plane and no two objects are at the same level. That is, we exclude structures such as arches, multiple columns, forks, etc. We also assume that

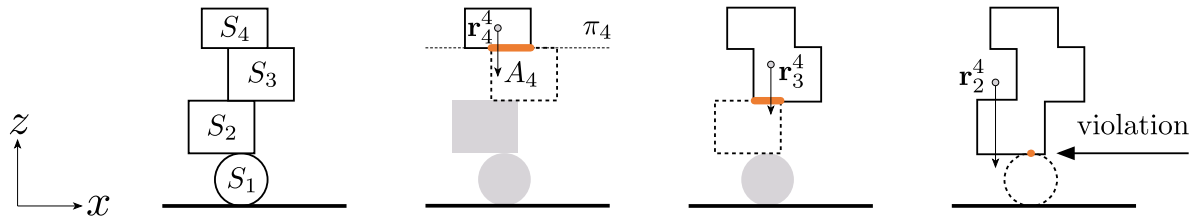


Figure 3.3: Centre of Mass criterion. The stability of a stack can be tested by considering sub-stacks sequentially, from top to bottom. For stability, the projection of the CoM of each sub-stack must lie within the contact surface with the block supporting it. As shown on the right, a cylindrical or spherical object offers an infinitesimally small contact surface which does not afford stability.

all objects are *convex*, so that a straight line between any two points of the object is fully contained within it.

In order to determine the stability of a stack, we must use the notion of *Centre of Mass* (CoM). Let $\mathbf{p} = (x, y, z) \in S_i \subset \mathbb{R}^3$ be a point contained within the rigid-body S_i. If m is the mass of the object and if the material is homogeneous with density ρ, then its CoM is given by $\mathbf{r}_i = \rho \int_{S_i} \mathbf{p}\, dx\, dy\, dz/m$.

We now study the stability of an object on top of another and then generalise the result to a full stack. For that, it is useful to refer to the topmost two blocks in Figure 3.3. Assume that the rigid-body S_4 is immersed in a uniform gravity field acting in the negative direction of the z axis. Furthermore, assume that S_4 is resting on a horizontal surface (in this case S_3) such that all of its contact points are contained in a horizontal plane π and $A \subset \pi$ denotes the convex hull of such points. Then S_4 is stable if, and only if, the projection of its CoM $\mathbf{r}_4$ on π is contained in A (Wieber [205]), which we write as $\text{Proj}_\pi(\mathbf{r}_4) \in A$. If S_4 rests in a stable position on S_3, the combination of (S_3, S_4) can be seen as a rigid-body with CoM $\mathbf{r}_3^4$. We can then check the stability of the entity (S_4, S_3) with respect to S_2. Proceeding iteratively for every object from top to bottom of the stack results in the following lemma illustrated in Figure 3.3:

Lemma 1. *Let $S_1, \ldots, S_n$ be a collection of convex rigid bodies forming a single-stranded tower resting on a flat ground plane S_0. Let $m_1, \ldots, m_n$ be the masses of the objects and $\mathbf{r}_1, \ldots, \mathbf{r}_n$ their centres of mass. Furthermore, let A_i be the contact surface between object S_{i-1} and S_i and let $\pi_i \subset A_i$ be the plane containing it. Assume that π is parallel to the xy plane,*

which in turn is orthogonal to gravity. Then, if the objects are initially at rest, the tower is stable if, and only if,

$$\forall i = 1, \ldots, n-1: \quad \text{Proj}_{\pi_i}(\mathbf{r}^n_{i+1}) \in A_i, \quad \mathbf{r}^n_{i+1} = \frac{\sum_{j=i+1}^{n} m_j \mathbf{r}_j}{\sum_{j=i+1}^{n} m_j} \tag{3.1}$$

where $\mathbf{r}^n_{i+1}$ *is the overall CoM of the topmost* $n-i$ *blocks.*

This lemma can be used to assess the stability of a stack by checking the CoM condition from top to bottom for every interface A_i. Note that what is important is not the centre of mass of the individual blocks, but that of the part of the tower above each surface A_i. Thus it is possible to construct a stable stack that has apparent CoM violations for individual blocks, but that is overall stable due to the counter-balancing effect of the other blocks on top. Importantly, this allows for complex stacks that cannot be constructed in a bottom-up manner by placing only one object at a time.

We specifically distinguish between two types of instabilities. The first is *violation of the planar surface criterion* (VPSF). This is caused by an object stacked on top of a curved surface which violates Equation (3.1) due to the infinitesimally small contact area. It is worth noting that this depends on the shape of the objects and not on the relative object positioning. The second type of instability is called *violation of the centre of mass criterion* (VCOM), and comprises violations of Equation (3.1) that depend instead on the positioning of the objects in the stack. For each unstable scenario we introduce either a VPSF or a VCOM violation for exactly one contact area A_i.

For dataset construction, Lemma 1 thus allows us to tightly control which stability violation occurs in each simulated scenario and to mark in each image which object it is attributable to (cf. Figure 3.4).

3.4 Stability Prediction

In this section, we construct models that can predict the stability of a stack from RGB images alone. We learn these models from passive observations of stable and unstable stacks. Specifically, our vision-based stability classifier is trained to distinguish

between stable and unstable towers (Section 3.4.1) and validated by demonstrating state-of-the-art performance on both simulated and real data. We also quantify how reliably the models can localise the mechanical stability violations present in the unstable stacks (Section 3.4.2).

3.4.1 Training the Stability Predictor

We train a visual classifier for the task of predicting whether a shape stack is stable or not using images[3] from the ShapeStacks dataset, annotated with binary stability labels.

To this end we investigate the use of two neural network architectures commonly used for image-based classification: AlexNet (Krizhevsky et al. [111]) and Inception v4 (Szegedy et al. [192]). In both cases we optimise the network parameters θ given our dataset $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ of images $x^{(i)}$ and stability labels $y^{(i)}$ by minimising the following logistic regression loss:

$$L(\theta; D) = -\sum_{i=1}^{m} y^{(i)} \log\left(\frac{1}{1+e^{-f(x^{(i)};\theta)}}\right) + (1-y^{(i)}) \log\left(1-\frac{1}{1+e^{-f(x^{(i)};\theta)}}\right) \quad (3.2)$$

The unscaled logit output of the CNNs is denoted by $f(x;\theta)$ and the label values are $y = 0$ for stable and $y = 1$ for unstable images. Inception v4 and AlexNet are both trained using the RMSProp optimiser (Hinton et al. [81]) with solver hyper-parameters as reported in (Szegedy et al. [192]) for 80 epochs.

We use the two different subsets of ShapeStacks during training (cf. Table 3.1), each one containing an equal amount of stable and unstable images. Both types of violations (VCOM and VPSF, cf. Section 3.3.2) are evenly represented among unstable images. We also reserve a set of 46,560 images featuring stacks of all shapes as final test set. During training, we augment the training images by randomising colours, varying aspect-ratios, and applying random cropping, vertical flipping and

[3]We only use still images of initial stack configurations and no images depicting collapses from later time points in the simulations.

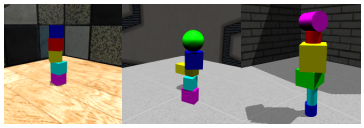
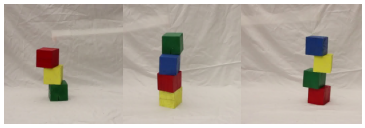
	AlexNet		INCPv4-IMGN		INCPv4		Physnet	VDA
	Cubes	CCS	Cubes	CCS	Cubes	CCS		
Simulated	60.5%	58.8%	76.2%	**84.9%**	77.7%	**84.9**%	N/A	N/A
Real	65.5%	52.5%	73.2%	64.9%	74.7%	66.3%	66.7%	**75**%
Simulated Examples				Real Examples				

Table 3.3: Stability prediction accuracy given as the percentage of correctly classified images into stable or unstable. AlexNet and Inception v4 (INCPv4) are trained from scratch on simulated data consisting of stacks featuring either cubes or CCS. INCPv4-IMGN is pre-trained on ImageNet (Deng et al. [30]). All algorithms are tested on both *real* images from Lerer et al. [117] and *simulated* images from our ShapeStacks test split featuring all shapes.

minimal in-plane rotation. We ensure that all data augmentations still yield physically plausible, upright towers.

Table 3.3 presents the performance of the classifiers on our simulated test data and on the real-world block tower data provided by Lerer et al. [117]. Our experiments suggest that AlexNet provides a useful baseline for CNN performance on this task. However, it is consistently outperformed by the Inception network. We choose the Inception v4 architecture trained on ShapeStacks data as the reference model in all further experiments.

As expected, both models perform best on the real-world data when only trained on cubes as the real-world images also only show stacks of cubes. Best performance is reached for both models on the combined ShapeStacks test data (featuring all shapes) when training is also performed on multiple object types. However, it is surprising how well the Inception network generalises from cubes to other structures suggesting that it learned an intuition about the CoM principle (Section 3.3.2) which is also applicable to more complex shapes. On real images, Inception v4, trained from scratch on our dataset, outperforms the baseline from Lerer et al. [117] and is on par with the more complex visual de-animation approach by J. Wu et al. [208], which translates the observed images into a physical state and checks stability with a

physics engine. We attribute this to the richness of the ShapeStacks dataset as well as to our data augmentation scheme, which results in a visually and structurally diverse set of stacks and hence affords good generalisation.

3.4.2 Instability Localisation

In order to probe whether the network grounds its stability prediction on sound mechanical principles we examine its ability to localise mechanical points of failure. Our approach is similar to that of Lerer et al. [117] though owing to the annotations included in the ShapeStacks dataset we are able to conduct a quantitative analysis on 1,500 randomly sampled images from the test set by comparing the network's attention maps with the corresponding ground truth stability segmentation maps (cf. Figure 3.4).

Specifically, we compute the attention maps by conducting an occlusion study whereby images are blurred using a Gaussian filter with a standard deviation of 30 pixels applied in a sliding window manner with stride 8 and a patch size of 14 x 14 pixels. To avoid creating object-like occlusion artefacts, the blurred patch does not have rigid boundaries but gradually fades into the image (cf. Figure 3.4 A and D). The patched images are given as an input to the stability classifier and the predicted stability scores are aggregated in a map (cf. Figure 3.4 B and E).

We then check whether the maximiser of the attention map is contained within the object responsible for stability violation (cf. Figure 3.4 C and F) and report results in Section 3.4.2. In 79.9% of all unstable cases, the network focuses on the violation region, which we define as the smallest rectangle enclosing the violating object and the first object to fall. For VPSF instabilities, the network attends to the violating, curved object with a likelihood of 52.1%. For VCOM instabilities, the network's main focus still remains on the violating object but is also spread out to the unsupported upper part of the tower (*First Object to Fall + Tower Top*) in 38.1% of the cases, which is in line with the physics governing VCOM instabilities (cf. Eq. (3.1)).

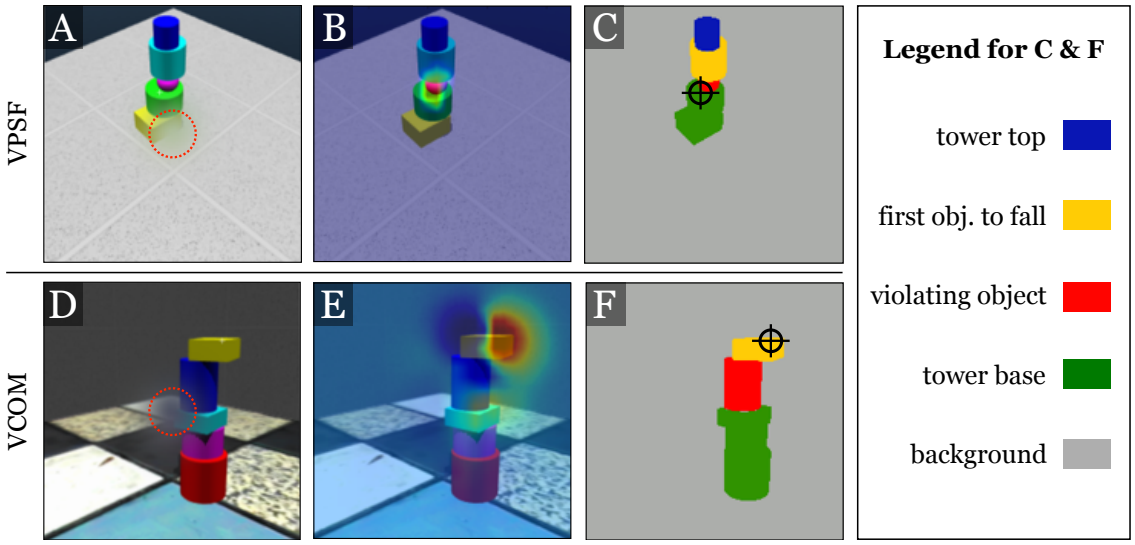


Figure 3.4: Attention visualisation obtained via an occlusion study. A Gaussian blur is applied in a sliding window manner to the image (A, D) , the increase (red) / decrease (blue) in the predicted stability is shown as a heatmap in (B, E), and the latter is compared to ground-truth segmentation maps in (C, F). The centres of attention are compared to the respective segmentation maps (C) and (F) and indeed correlate with the respective violation sites as indicated by the cross hairs.

	Violating Object	First Obj. to Fall	Violation Area	Tower Base	Tower Top	Back-ground
VCOM &VPSF	38.9%	29.3%	79.9%	5.9%	5.5%	20.4%
VCOM	32.7%	30.8%	76.5%	6.5%	7.3%	22.7%
VPSF	52.1%	26.3%	87.1%	4.6%	1.7%	15.4%
rnd. in image	1.6%	1.9%	4.9%	1.7%	1.8%	93.0%
rnd. in tower	19.3%	22.9%	59.0%	20.5%	21.7%	14.5%

Table 3.4: The fraction of times the network attends image areas with specific physical meaning (cf. Figure 3.4). 1,500 images were analysed with an Inception v4 network trained on the CCS data (cf. Section 3.4.1). The first row is aggregated over all instability types and the second and third rows offer a breakdown for the CoM (VCOM) and planar surface violations (VPSF), respectively. The fourth row lists the fractions of the areas occupied with the respective label across the segmentation maps of all unstable scenarios and serves as a reference point of how likely it is to focus on a specific area just by random chance. Likewise, the fifth row reports random chance attention within the tower.

3.5 Stacking and Stackability

So far, we have focussed on predicting the stability of stacks. However, it is not clear whether the models we learned understand the geometric affordances needed for

Figure 3.5: *Top row*: An unordered set of objects with random orientations. *Bottom row*: objects sorted from most stackable (left) to least stackable (right). Every object is oriented in the way which affords best *stackability* according to our network. The scores allow for division between different stability categories as visualised with white vertical lines.

actively building new stacks. Here, we answer this question by considering three active stacking tasks. The first one is to estimate the *stackability* of different objects and prioritise them while stacking (Section 3.5.1). The second is to accurately estimate the optimal placement of blocks on a stack through visual feedback (Section 3.5.2). The third is to counter-balance an unstable structure by placing an additional object on top (Section 3.5.3). All tasks show encouraging performance indicating that models do indeed acquire actionable physical knowledge from passive stability prediction.

3.5.1 Stackability

Different object shapes intrinsically have different stacking potential: While a cuboid can serve as a solid base in every orientation, a cylinder can only support objects when placed upright and a sphere is never a good choice as a supporting object. If an agent is given a set of blocks to stack, it can use an understanding of such affordances to prioritise objects, placing the most stable ones at the bottom of the stack. We define *stackability* of an object (i. e. its utility with regard to stack construction) by answering the question: "How well can this object support the others in my set?" Next, we show how to answer this question quantitatively using our learned stability predictor.

Given a set of objects, we compute their relative stackability scores as follows: Each object is placed on the ground as if it were the base of the stack using one of its

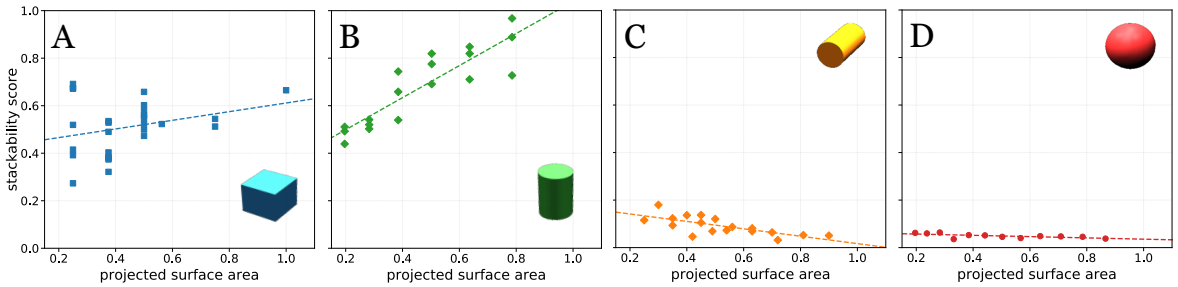


Figure 3.6: Correlation of the *stackability* score with the projected surface area for different object classes. The projected surface area is calculated by projecting the object onto the x-y-plane. Spheres and lying cylinders are given very low *stackability* scores. Upright cylinders and cuboids are generally more stackable as the projected surface area grows.

discrete orientations[4]. Then, all other objects are systematically placed on top of the base object, one at a time, in all of their respective orientations. An image of the resulting combination is generated and assessed for stability using our predictor. Positions for the top objects are sampled within a defined radius around the base object via simulated annealing and the maximum stability score is recorded. The stackability score of the base object is then estimated as the average maximum stability achieved by all the other objects as they are placed on top of it. We also add random perturbations to the base position, with the idea of reflecting stackability robustness in the estimated score.

Stackability can then be used to rank objects' shapes and orientations based on how well they can be expected to support other objects, as illustrated in Figure 3.5. We also examine the model's understanding of stackability quantitatively in Figure 3.6 computing scores over all object classes with varying volumes and aspect ratios. We generally find that the model ranks shapes in a sensible manner, preferring to stack on the largest face of cuboids, then on upright cylinders, and reject spheres as generally unsuitable for stacking. The results suggest that the suitability of different geometries to stacking is implicitly learned by stability prediction.

[4]Cuboids afford three discrete orientations, one for each of its three distinct faces (considering symmetry). Cylinders afford two orientations (upright and sideways) and spheres afford only one orientation due to their radial symmetry.

3.5.2 Stacking Shapes in Simulation

Next, we investigate the ability of the stability predictor to not only order objects in an active stacking scenario, but also to accurately position them in stable configurations. To do so, we design three stacking scenarios involving different shape types: cubes, cuboids and CCS. In each scenario, the method is given a pool of 12 different object shapes and sizes to stack with the goal of building as tall a tower as possible. Every scenario is observed from six cameras (cf. Figure 3.8D) which move upwards as the stack grows to guarantee full coverage of the process at any time. At the beginning of every stacking episode, background textures, object colours and scene lights are randomised. Then the stack order and best orientation for each objects are computed according to the stackability score (cf. Section 3.5.1).

The stacking process commences with the first object being placed at the scene centre. The object at place r in the stacking queue is always spawned at a fixed height h_r above the current tower trunk and candidate positions are sampled in the x-y-plane at $z = h_r$ according to the simulated annealing process described in Section 3.5.1. If no stable position is identified for a particular object (i. e. logistic regression score < 0.5), it is put aside and disregarded for the rest of the process. The process is iterated until the placement of an object results in the collapse of the stack or no more objects are available.

In Figure 3.7, we report achieved stack heights for two differently trained models in the three scenarios with cubes, cuboids and CCS, respectively. For each stacking episode, the algorithm is given a pool of 12 randomised objects. However, CCS scenarios always include exactly two spheres, so the maximum achievable height in this case is 11. We compare two stability predictors: One trained on cubes only (blue bars) and one trained on CSS objects (orange bars). The CCS stability predictor clearly outperforms the one trained on cubes only in all three scenarios. In fact, the cubes predictor only manages perform decently on cube stacking and largely fails when confronted with varied shapes highlighting the importance of training on a diverse shape set.

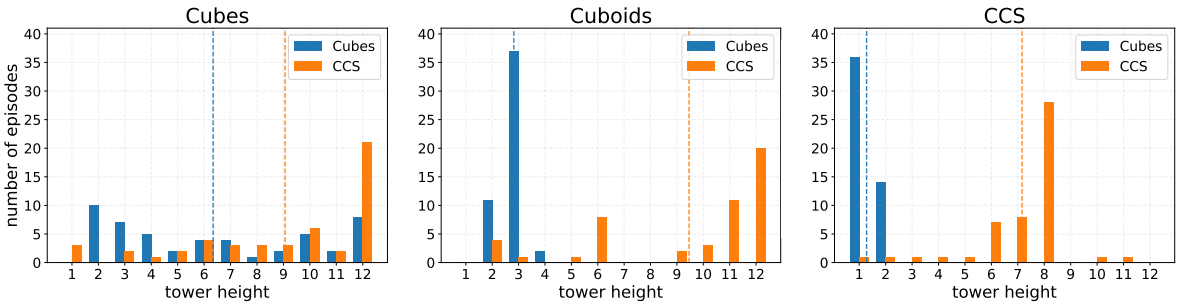


Figure 3.7: Stacking performance. The height of the bars indicate how often the algorithm built a tower with the respective number of objects before it fell over. The mean tower height is indicated with a vertical dashed line.

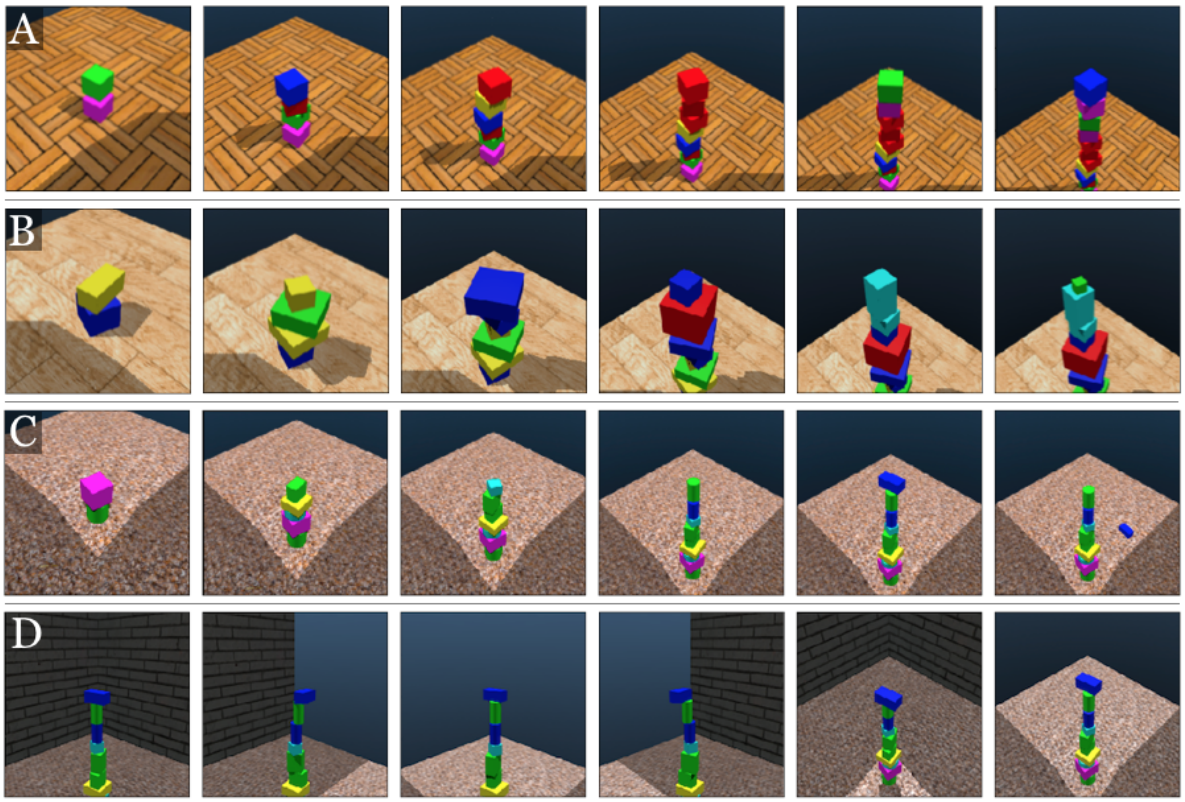
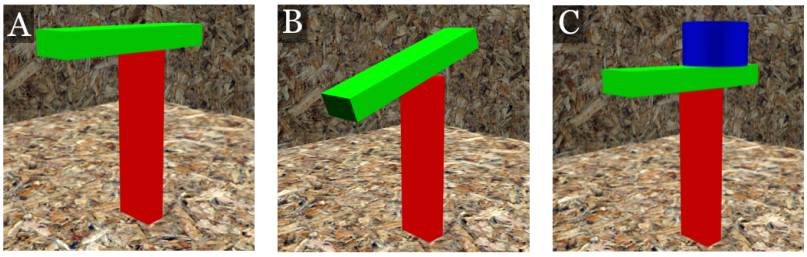


Figure 3.8: Three examples of stacking attempts. In (A) and (B), the algorithm successfully stacked up cubes and cuboids to the maximum height of 12. In C, the algorithm placed the 10th object in a way that violates Eq. (3.1). In (D), the images obtained from the different camera angles are shown for the failed stacking attempt in (C).

3.5.3 Balancing Unstable Structures

In the final task, we present our model with an unstable stack, freeze it such that it does not collapse, and then ask the algorithm to place an additional object on top to counter-balance the instability. This is a subtle task that requires the model to



Object	Success Rate
Cube	76%
Cuboid	94%
Cylinder	72%
Sphere	98%

Figure 3.9: Counterbalancing unstable structures. A: frozen, unstable stack; B: collapsing tower; C: successful placement of a counterweight that prevents collapse. Right: success rates for different counterweight types aggregated over 50 episodes.

understand the concept of counterbalancing and cannot be solved by simply centering a block on top of the one below. Figure 3.9 shows that our algorithm successfully solves this task with high probability in an "unstable T scenario" for different types of counterweight objects.

3.6 Conclusion

We investigate the acquisition of physical intuition and geometric affordances in the context of vision-based, generalised object stacking. To that end, we construct the ShapeStacks dataset featuring diverse stacks of shapes with detailed annotations of mechanical stability violations and release it publicly. We train a visual stability predictor on ShapeStacks which performs commensurately with state-of-the-art on simulated and real world images. Our model also correctly localises structural instabilities, yields an intuitive notion about the stackability of objects and successfully guides a simulated stacking process solely based on visual cues. Our results suggest that an intuitive understanding about physical principles and geometric affordances can be acquired from visual observation and effectively utilised in manipulation tasks.

Acknowledgements

This research was funded by the European Research Council under grant ERC 638009-IDIU and the EPSRC AIMS Centre for Doctoral Training at Oxford University. We

would like to thank Markus Wulfmeier for helpful comments on the draft of this paper. We also thank Adriàn Peñate-Sànchez, Rob Weston and Georgi Tinchev for proofreading.

4

Learning Affordances in Object-Centric Generative Models

In this chapter we investigate the connection between a high-level, physical intuition – the *reachability affordance* of a stick-like tool – and its connection to a structured latent space capturing factors of variation pertaining to 3D geometry. We demonstrate that the signal derived from the high-level intuition classifier enables a traversal of the underlying latent space which corresponds to smooth interpolations of tool shapes. This is shown on the example of imagining suitable 3D tool geometries given an image of the reaching task at hand by simply following the gradient towards increased reachability. On top of arriving at a feasible solution, our model also provides smooth deformations of an initial tool shape during the optimisation process indicating that an appropriate manifold of physically plausible solutions has been learned. This work was presented orally and received an outstanding paper award as:

Y. Wu, S. Kasewa, O. Groth, S. Salter, L. Sun, O. Parker Jones, and I. Posner. "Learning Affordances in Object-Centric Generative Models". In: *Workshop on Object-Oriented Learning at ICML 2020* (July 2020)

Abstract

Given visual observations of a reaching task together with a stick-like tool, we propose a novel approach that learns to exploit task-relevant object affordances by combining generative modelling with a task-based performance predictor. The embedding learned by the generative model captures the factors of variation in object geometry, e. g. length, width, and configuration. The performance predictor identifies sub-manifolds correlated with task success in a weakly supervised manner. Using a 3D simulation environment, we demonstrate that traversing the latent space in this task-driven way results in appropriate tool geometries for the task at hand. Our results suggest that affordances are encoded along smooth trajectories in the learned latent space. Given only *high-level* performance criteria (such as task success), accessing these emergent affordances via gradient descent enables the agent to manipulate learned object geometries in a targeted and deliberate way.

4.1 Introduction

The advent of deep generative models e. g. Burgess et al. [25], Engelcke et al. [40], and Greff et al. [62] with their aptitude for unsupervised representation learning casts a new light on learning *affordances* (Gibson [54]). This kind of representation learning raises a tantalising question: Given that generative models naturally capture factors of variation, could they also be used to expose these factors such that they can be modified in a task-driven way? We posit that a task-driven traversal of a structured latent space leads to *affordances* emerging naturally along trajectories in this space. This is in stark contrast to more common approaches to affordance learning where it is achieved via direct supervision or implicitly via imitation e. g. Do et al. [33], Grabner et al. [61], Y. Liu et al. [127], Myers et al. [144], and Tikhanoff et al. [193]. The setting we choose for our investigation is that of tool synthesis for reaching tasks as commonly investigated in the cognitive sciences (Ambrose [5] and Emery and Clayton [39]).

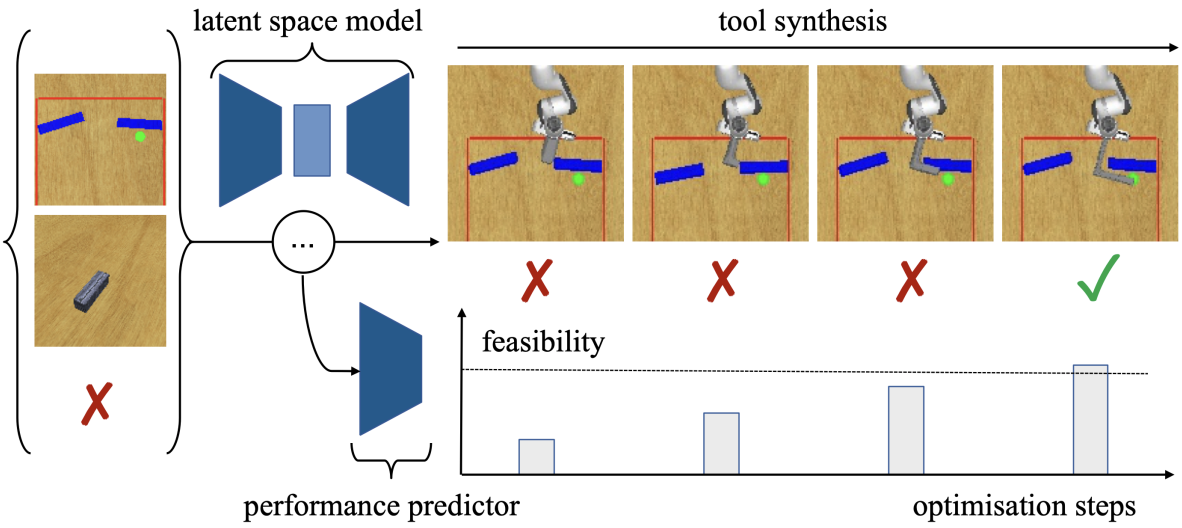


Figure 4.1: Tool innovation for a reaching task. Our model is trained on data-triplets {task observation, tool observation, success indicator}. The goal is to determine if the given tool can reach the green goal while avoiding blue barriers and remaining behind the red boundary. When the tool cannot satisfy these constraints, our approach (via the performance predictor) *imagines* how one may augment it in order to solve the task. We are interested in what these augmentations, during *tool synthesis*, imply about the learned object representations.

In order to demonstrate that a task-aware latent space encodes useful affordance information we require a mechanism to train such a model as well as to purposefully explore the space. To this end we propose an architecture in which a task-based performance predictor (a classifier) operates on the latent space of a generative model (see Fig. 4.1). During training the classifier is used to provide an auxiliary objective, aiding in shaping the latent space. Importantly, however, during test time the performance predictor is used to guide exploration of the latent space via activation maximisation (Erhan et al. [41], Simonyan et al. [179], and Zeiler and Fergus [230]), thus explicitly exploiting the structure of the space. While our desire to affect factors of influence is similar in spirit to the notion of disentanglement, it contrasts significantly with models such as β-VAE (Higgins et al. [79]), where the factors of influence are effectively encouraged to be axis-aligned. Our approach instead relies on a high-level auxiliary loss to discover the direction in latent space to explore.

Our experiments demonstrate that artificial agents are able to *imagine* an appropriate tool for a variety of reaching tasks by manipulating the tool's task-relevant affordances.

To the best of our knowledge, this makes us the first to demonstrate an artificial agent's ability to imagine, or synthesise, images of tools appropriate for a given task via optimisation in a structured latent embedding. Similarly, while activation maximisation has been used to visualise modified input images before e. g. Mordvintsev et al. [141], we believe this work to be the first to effect deliberate manipulation of factors of influence by chaining the outcome of a task predictor to the latent space, and then decoding the latent representation back into a 3D mesh. Beyond the application of tool synthesis, we believe our work to provide novel perspectives on affordance learning and disentanglement in demonstrating that object affordances can be viewed as *trajectories* in a structured latent space as well as by providing a novel architecture adept at deliberately manipulating interpretable factors of influence.

4.2 Related Work

An *affordance* describes a potential action to be performed on an object (e. g. a doorknob *affords* being turned) (Gibson [54]). In computer vision and robotics, affordances are commonly learned in a supervised fashion where models discriminate between discrete affordance classes or predict masks for image regions which afford certain types of human interaction e. g. Do et al. [33], Kjellström et al. [104], Mar et al. [134], Myers et al. [144], Stoytchev [188], and Tikhanoff et al. [193]. Most works in this domain learn from object shapes which have been given an affordance label a priori. However, the affordance of a shape is only properly defined in the context of a task.

Recent advances in 3D shape generation employ variational models (Girdhar et al. [55] and J. Wu et al. [211]) to capture complex manifolds of 3D objects. Besides their expressive capabilities, the latent spaces of such models also enable smooth interpolation between shapes. Remarkable results have been demonstrated including 'shape algebra' (J. Wu et al. [211]) and the preservation of object part semantics (Kohli et al. [107]) and fine-grained shape styles (Yifan et al. [225]) during interpolation. This shows the potential of disentangling meaningful factors of variation in the

latent representation of 3D shapes. Inspired by this, we investigate whether these factors can be exposed in a task-driven way. We propose an architecture in which a generative model for 3D object reconstruction (S. Liu et al. [126]) is paired with activation maximisation e. g. Erhan et al. [41], Simonyan et al. [179], and Zeiler and Fergus [230] of a task-driven performance predictor.

4.3 Method

Our overarching goal is to perform task-specific tool synthesis for 3D reaching tasks. We frame the challenge of tool imagination as an optimisation problem in a structured latent space obtained using a generative model. The optimisation is driven by a high-level, task-specific performance predictor, which assesses whether a target specified by a goal image I_G is reachable given a particular tool and in the presence of obstacles (cf. Figure 4.1). To map from tool images into manipulable 3D tools, we first train an off-the-shelf 3D single-view reconstruction model taking as input tool images I_T^i, I_T^j and corresponding tool silhouettes I_S^i, I_S^j as rendered from two different vantage points i and j. This 3D reconstruction model, which is depicted in Figure 4.2, is an implementation of the encoder-decoder architecture proposed by Kato et al. [96] and S. Liu et al. [126].

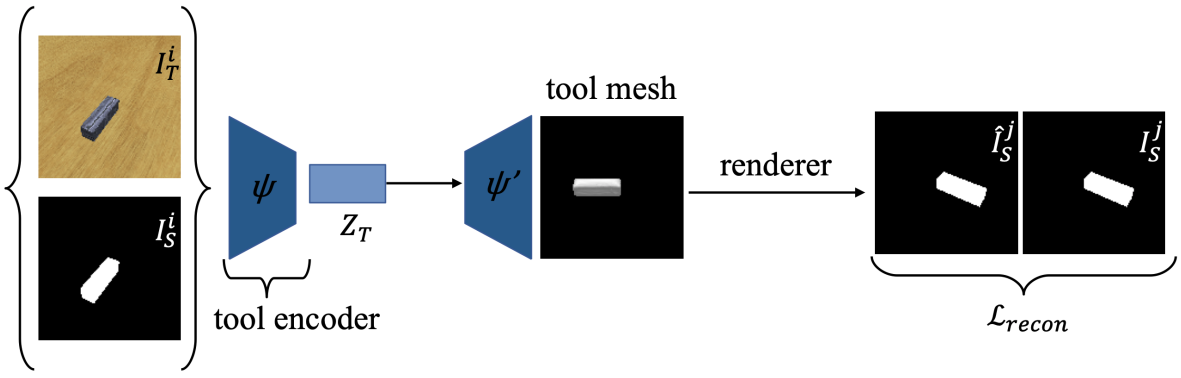


Figure 4.2: System architecture for 3D reconstruction model based on silhouette views.

After training, the encoder can infer the tool representation that contains the 3D structure information given a single-view RGB image and its silhouette as input. This

representation is implicitly used to optimise the tool configuration to make it suitable for the task at hand (cf. Figure 4.3).

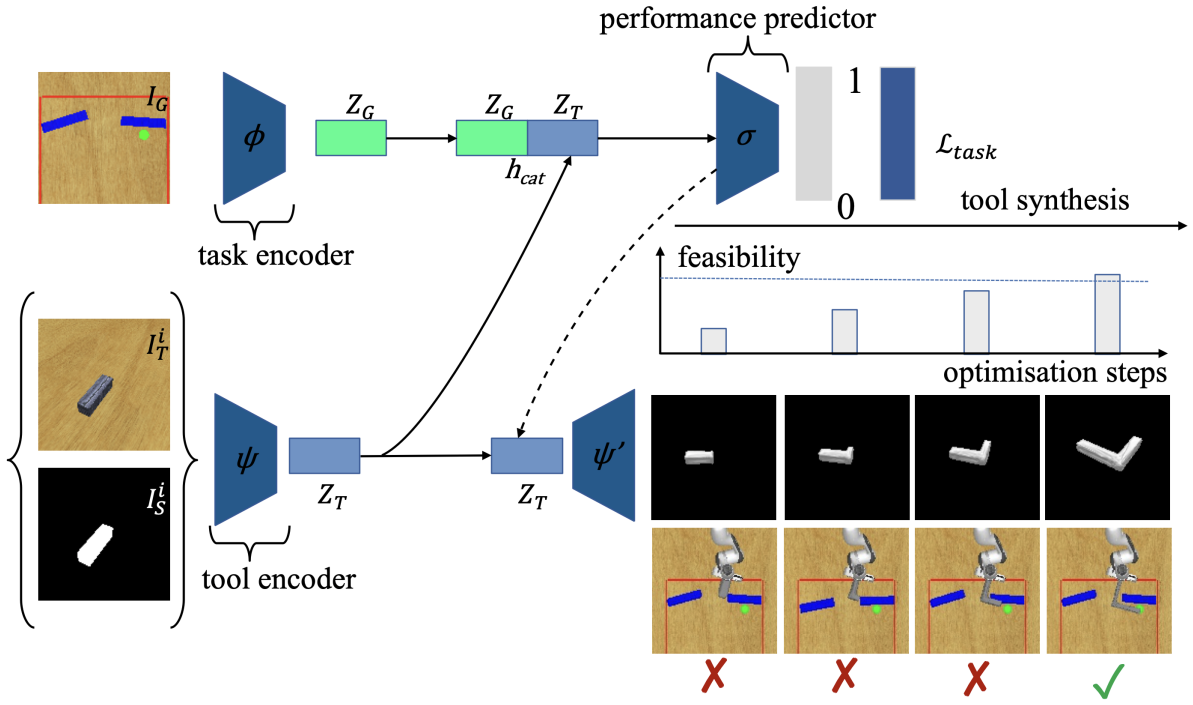


Figure 4.3: The model architecture. A convolutional encoder ϕ represents the task image I_G as a latent vector $\mathbf{z}_G$. In parallel, given I_T^i,I_S^i, the 3D tool encoder ψ produces a latent representation $\mathbf{z}_T$. The variational model (ψ, ψ') is trained as described in Kato et al. [96] and N. Wang et al. [201]. The concatenated tool-task representation $\mathbf{h}_{cat}$ is used by a classifier σ to estimate the success of the tool for solving the task. Given the gradient signal during optimisation for task success via the classifier, the latent tool representation $\mathbf{z}_T$ gets updated to render an increasingly suitable tool for the task.

More formally, we consider N data instances: $\{(I_G^n, I_T^{n,i}, I_T^{n,j}, I_S^{n,i}, I_S^{n,j}, \rho^n)\}_{n=1}^N$, where each example features a task image I_G, tool images I_T in two randomly selected views i and j, and their corresponding silhouettes I_S, as well as a binary label ρ indicating the feasibility of reaching the target with the given tool. Examples of task images and model inputs for five scenarios types, A–E, are shown in Supplementary Figure 4.A.1. In all our experiments, we restrict the training input to such sparse high-level instances. For additional details on the dataset, we refer the reader to the supplementary material.

4.3.1 Representing Tasks and Tools

Given that our tools are presented in tool images I_T, it is necessary for the first processing step to perform a 3D reconstruction of I_T, from pixels into meshes. To achieve this single view 3D reconstruction of images into their meshes, we employ an architecture proposed in prior work (Kato et al. [96] and N. Wang et al. [201]). The 3D reconstruction model consists of two parts: an encoder network and a mesh decoder. Given the tool image and its silhouette in view i, I_T^i and I_S^i, we denote the latent variable encoding the tool computed by the encoder, $\boldsymbol{\psi}$, as

$$\boldsymbol{\psi}(I_T^i, I_S^i) = \mathbf{z}_T. \tag{4.1}$$

The mesh decoder takes $\mathbf{z}_T$ as input and synthesises the mesh by deforming a template. A differentiable renderer (S. Liu et al. [126]) predicts the tool's silhoutte $\hat{I}_S^j$ in another view j, which is compared to the ground-truth silhouette I_S^j to compute the silhouette loss $\mathcal{L}_s$. This silhouette loss $\mathcal{L}_s$ together with an auxiliary geometry loss $\mathcal{L}_g$ formulates the total 3D reconstruction loss

$$\mathcal{L}_{recon} = \mathcal{L}_s + \mu\mathcal{L}_g, \tag{4.2}$$

where μ is the weight of the geometry loss. We refer the reader to S. Liu et al. [126] for the exact hyperparameter and training setup of the 3D reconstruction model.

Task images I_G are similarly represented in an abstract latent space. For this we employ a task encoder, ϕ, which consists of a stack of convolutional layers. ϕ takes the task image I_G as input and maps it into the task embedding $\mathbf{z}_G$.

4.3.2 Task-driven Learning

The learned representation $\mathbf{z}_T$ appears to encode task-relevant information such as tool length, width, and shape. In order to perform tool imagination, the sub-manifold of the latent space that corresponds to the task-relevant features needs to be accessed and traversed. This is achieved by adding a three-layer MLP as a classifier **ff**. The classifier **ff** takes as input a concatenation $\mathbf{h}_{cat}$ of the task embedding $\mathbf{z}_G$ and the tool

representation $\mathbf{z}_T$, and predicts the softmax over the binary task success. The classifier learns to identify the task-relevant sub-manifold of the latent space by using the sparse success signal ρ and optimising the binary-cross entropy loss, such that

$$\begin{aligned}\mathcal{L}_{task}\left(\mathbf{ff}\left(\mathbf{h}_{cat}\right), \rho\right) = &- \left(\rho \log\left(\mathbf{ff}(\mathbf{h}_{cat})\right)\right. \\ &\left. + (1-\rho)\log\left(1-\mathbf{ff}(\mathbf{h}_{cat})\right)\right),\end{aligned} \tag{4.3}$$

where $\rho \in \{0, 1\}$ is a binary signal indicating whether or not it is feasible to solve the task with the given tool. The whole system is trained end-to-end with the loss given by

$$\mathcal{L}\left(I_G, I_T^i, I_S^i, I_T^j, I_S^j, \rho\right) = \mathcal{L}_{recon} + \mathcal{L}_{task}. \tag{4.4}$$

Note that the gradient from the task classifier σ propagates through both the task encoder ϕ and the toolkit encoder ψ, thereby helping to shape the latent representations of the toolkit with respect to the requirements for task success.

4.3.3 Tool Imagination

Once trained, our model can synthesise new tools by traversing the latent manifold of individual tools following the trajectories that maximise classification success given a tool image and its silhouette (Supplementary Figure 4.3). To do this, we first pick a tool candidate and concatenate its representation $\mathbf{z}_T$ with the task embedding $\mathbf{z}_G$, warm-starting the imagination process. The concatenated embedding $\mathbf{h}_{cat}$ is then fed into the performance predictor $\mathbf{ff}$ to compute the gradient with respect to the tool embedding $\mathbf{z}_T$. We use activation maximisation (Erhan et al. [41], Simonyan et al. [179], and Zeiler and Fergus [230]) to optimise for $\mathbf{z}_T$ given the loss $\mathcal{L}_{task}$ of the success estimation $\mathbf{ff}\left(\mathbf{h}_{cat}\right)$ and a feasibility target $\rho_s = 1$ such that

$$\mathbf{z}_T = \mathbf{z}_T + \eta \frac{\partial \mathcal{L}_{task}\left(\mathbf{ff}\left(\mathbf{z}_T\right), \rho_s\right)}{\partial \mathbf{z}_T}, \tag{4.5}$$

where η denotes the learning rate for the update. We apply this gradient update for S steps or until the success estimation $\mathbf{ff}\left(\mathbf{z}_T\right)$ reaches a threshold γ, and use $\psi'(\mathbf{z}_T)$ to generate the imagined 3D tool mesh represented by $\mathbf{z}_T$.

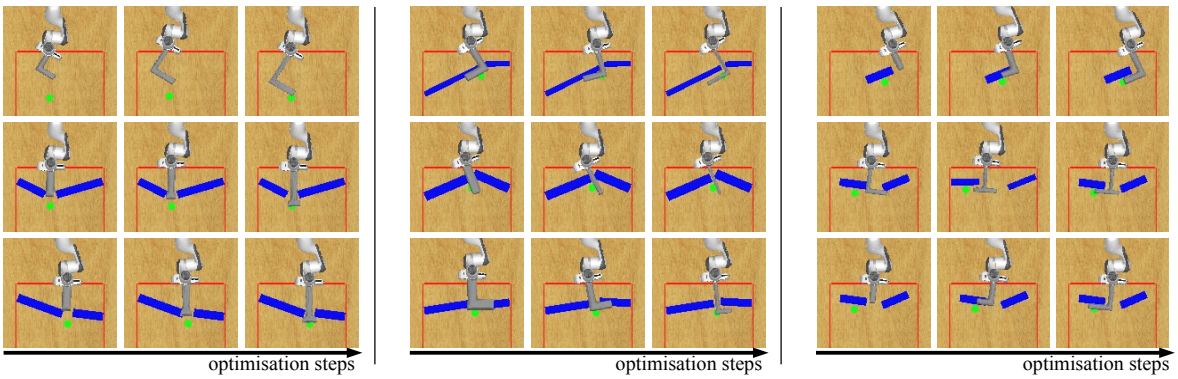


Figure 4.4: Qualitative results of tool evolution during the imagination process. Each row illustrates how the imagination procedure can succeed at constructing tools that solve the task by: increasing **tool length** (***left***), decreasing **tool width** (***middle***), and altering **tool shape** (***right***) creating an appropriately oriented hook. Each row in each grid represents a different imagination experiment.

4.4 Experiments

4.4.1 Model Training

In order to gauge the influence of the task feasibility signal on the latent space of the tools, we train the model in two different setups. A *task-driven* model is trained with a curriculum: First, the 3D reconstruction module is trained on tool images alone. Then, the performance predictor is trained jointly with this backbone, i. e. the gradient from the predictor is allowed to back-propagate into the encoder of the 3D reconstruction network. In a *task-unaware* ablation we keep the pre-trained 3D reconstruction weights fixed during the predictor training removing any influence of the task performance on the latent space. A *random walk* baseline reveals that a simple stochastic exploration of the latent space is not sufficient to find suitable tool geometries.

4.4.2 Qualitative Results

Qualitative examples of the tool imagination process are provided in Figures 4.4 and 4.5. In the right-middle example of Figure 4.4, a novel T-shape tool is created, suggesting that the model encodes the vertical stick-part and horizontal hook-part as distinct elementary parts. The model also learns to interpolate the direction of the

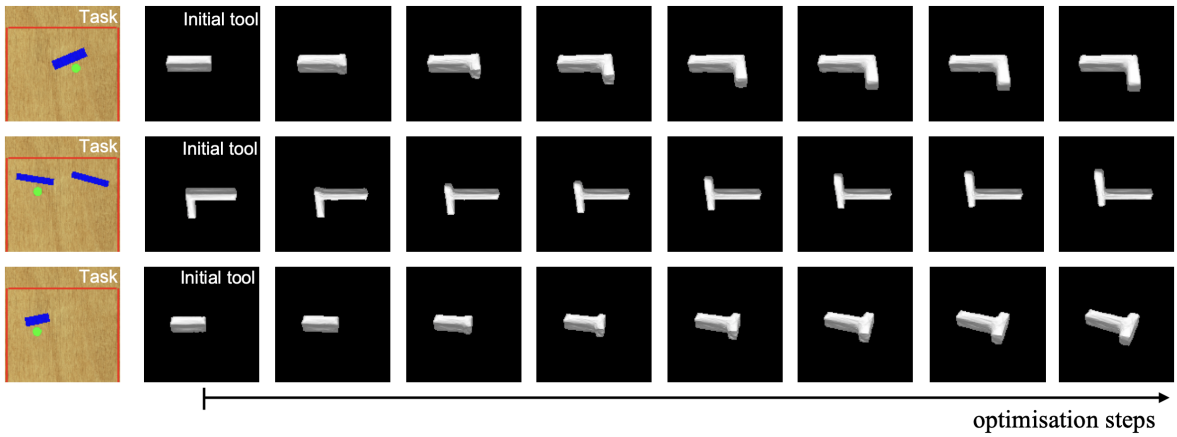


Figure 4.5: Examples of tool synthesis progression during the imagination process. In the top row, a stick tool morphs into a hook. The middle row shows a left-facing hook transforming into a right-facing hook. In the bottom row, the tool changes into a novel T-shape. Constraints on these optimisations are specified via task embeddings corresponding to the task images on the far left.

hook part between pointing left and right, which leads to a novel tool. As shown in Figure 4.5, tools are modified in a smooth manner, leading us to hypothesise that tools are embedded in a continuous manifold of changing length, width, and configuration. Optimising the latent embedding for the highest performance predictor score often drives the tools to evolve along these properties. This suggests that these geometric variables are encoded as *trajectories* in the structured latent space learnt by our model and deliberately traversed via a high-level task objective in the form of the performance predictor.

4.4.3 Quantitative Results

We evaluate whether our model can generate tools to succeed in the reaching tasks. For each instance the target signal for feasibility is set to $\rho_s = 1$, i. e. *success*. Then, the latent vector of the tool is modified via backpropagation using a learning rate of 0.01 for 10,000 steps or until $\mathbf{ff}(\mathbf{h}_{cat})$ reaches the threshold of $\gamma = 0.997$. The imagined tool mesh is generated via the mesh decoder ψ'. This is then rendered into a top-down view and tested using a feasibility test which checks whether all geometric constraints are satisfied.We report our results in Table 4.1 as mean success performances within a 95% confidence interval around the estimated mean. The

	Tool Imagination Success [%]		
Scenario	Task-Driven	Task-Unaware	Random Walk
A	**96.4 ± 2.3**	55.6 ± 6.2	3.6 ± 2.3
B	**78.8 ± 5.1**	42.0 ± 6.1	5.6 ± 2.9
C	**76.4 ± 5.3**	56.8 ± 6.1	23.6 ± 5.3
D	**81.2 ± 4.8**	**75.2 ± 5.4**	2.4 ± 1.9
E	**86.4 ± 4.3**	**88.4 ± 4.0**	13.6 ± 4.3
Total	**83.8 ± 2.0**	63.6 ± 2.7	9.8 ± 1.7

Table 4.1: Comparison of imagination processes when artificially warmstarting from the same unsuitable tools in each instance. Best results are highlighted in bold. The task scenarios, A–E, are described in Supplementary Figure 4.A.1.

task-unaware ablation provides a much stronger baseline compared to the random walk baseline transforming tools successfully in 63.6% of the cases. However, the task-driven model significantly outperforms it, boasting a global success rate of 83.8% on the test cases. This finding implies that the joint training of 3D latent representation and task performance prediction shapes the latent space in a 'task-aware' way encoding properties which are conducive to task success (e. g. length, width, and configuration of a tool).

4.5 Conclusion

In this paper we investigate the ability of an agent to synthesise tools via task-driven imagination in a set of simulated reaching tasks. Our approach uses a novel architecture in which a high-level performance predictor drives an optimisation process in a structured latent space. The model successfully learns to modify interpretable properties of tools such as length, width, and configuration. The experimental results suggest that these object affordances are encoded as *trajectories* in a learnt latent space, which are sought out during activation maximisation of the task predictor. In addition, more task-appropriate trajectories were found by jointly training the performance predictor with the encoder than by training them independently. This work may help in our understanding of object affordances, while offering up a novel way to disentangle interpretable factors of variation.

Acknowledgements

This research was supported by an EPSRC Programme Grant (EP/M019918/1) as well as the EPSRC AIMS Centre for Doctoral Training at Oxford University and the China Scholarship Council. This work was supported by a DPhil scholarship for SK from the Future of Humanity Institute, University of Oxford. We are grateful to Martin Engelcke for sharing his implementation of MONet. The authors would also like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility (http://dx.doi.org/10.5281/zenodo.22558) and the STFC Hartree Centre (https://www.jade.ac.uk/) in carrying out this work.

Appendix

4.A Dataset of Controlled Reaching Scenarios

To investigate tool imagination, we designed a set of simulated reaching tasks with clear and controllable factors of influence. Each task image is comprised of a green target button, three red lines delineating the workspace area, and, optionally, a set of blue obstacles. We also vary the goal location and the sizes and positions of the obstacles. For each task image, we provide a second image depicting a tool, i. e. a straight stick or an L-shaped hook, with varying dimensions, shapes, and textures. Given a pair of images (i. e. the task image and the tool image), the goal is to predict whether the tool for a given task scene can reach the target (the green dot) while simultaneously avoiding obstacles (blue areas) and remaining on the exterior of the workspace (i. e. behind the red line). Depending on the task image, the applicability of a tool is determined by different subsets of its attributes. For example, if the target button is unobstructed, then any tool of sufficient length will satisfy the constraints (regardless of its width or shape). However, when the target is hidden behind a corner, or only accessible through a narrow gap, an appropriate tool also needs to feature a long-enough hook, or a thin-enough handle, respectively. As depicted in Figure 4.A.1, we have designed five scenario types to study these factors of influence in isolation and in combination. The task setting and tool are first rendered in a top-down view for the geometric applicability check. Then the tool is rendered in 12 different views for the 3D reconstruction. The geometric applicability check verifies whether or not any of the tools can reach the target while satisfying the task constraints.

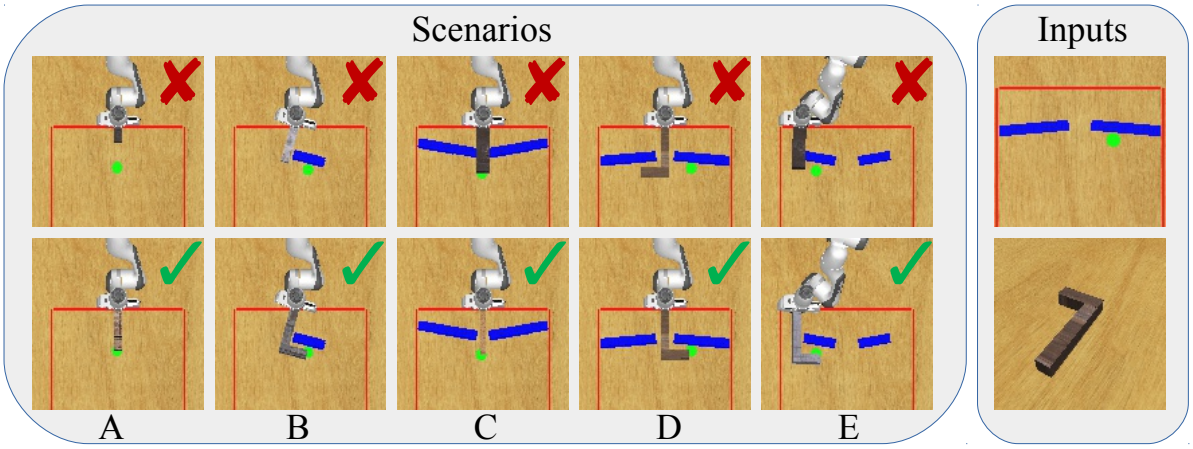


Figure 4.A.1: (Left) Task examples from our dataset. Top and bottom rows correspond to unsuccessful and successful tool examples respectively. Columns `A - E` represent five different task scenario types each imposing different tool constraints including width, length, orientation and shape. Note that the robot is fixed at its base on the table and constrained to remain outside the red boundary. Hence, it can only reach the green target with a tool while avoiding collisions with the blue obstacles. (Right) Model inputs {task observation, tool observation} during training and test time.

Dataset Construction Minutiae Our synthetic dataset consists of pairs of task image (in top-down view) and tool image as well as the corresponding silhouettes in 12 different views. We also record the tool image from a top-down view for the tool-applicability check. All images are 128×128 pixels. The applicability of a tool to a task is determined by sampling 100 interior points of the tool polygon, overlaying the sampled point with the target and rotating the tool polygon between $[-60, +60]$ degrees from the y-axis which is oriented perpendicular to the tabletop. If any such pose of the tool satisfies all constraints, i. e. touching the space behind the red line while not colliding with any obstacle, we consider the tool applicable for the given task. All train- and test-cases are constrained to have only sticks and hooks as tools.

The dataset contains a total of 18,500 scenarios. Table 4.A.1 shows a breakdown of each by scenario type and split. Each split has an equal number of feasible and infeasible instances for each scenario type.

Type	Training	Validation
A	4,000	500
B	4,000	500
C	4,000	500
D	4,000	500
E	-	500
Total	16,000	2,500

Table 4.A.1: Number of instances by scenario type.

4.B Architecture and Training Details

The task-based classifier consists of two sub-parts: a task encoder and a classifier. The task encoder stacks five convolutional blocks followed by a single convolutional layer. First, the five consecutive convolutional blocks make use of 16, 32, 64, 64, 128 output channels respectively. Each convolutional block contains two consecutive sub-convolutional blocks, each with a kernel size of 5 and padding of 2 (and with stride of 1 and stride of 2, respectively). Second, the additional convolutional layer has kernel size 4, stride 1, padding 0, and 256 output channels. All convolutional layers use an ELU nonlinearity. The classifier is a three layer MLP with input dimension 768, and successive hidden layers of 1024 and 512 neurons respectively. Each of these layers use eLU activation functions. Given that the classifier outputs logits for a binary classifier, the output dimension is 2. All experiments are performed in PyTorch, using the ADAM optimiser with a learning rate of 0.0001 and a batch size of 16.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	
Publication Status	□ Published □ Accepted for Publication □ Submitted for Publication **x** Unpublished and unsubmitted work written in a manuscript style
Publication Details	Yizhe Wu*, Sudhanshu Kasewa*, Oliver Groth*, Sasha Salter, Li Sun, Oiwi Parker Jones, Ingmar Posner "Learning Affordances in Object-Centric Generative Models". In: Workshop on Object-Oriented Learning at ICML 2020 (July 2020)

Student Confirmation

Student Name:	Oliver Groth		
Contribution to the Paper	- contributed to the idea of using activation maximization of the feasibility classifier for the adjustment of tool geometry - designed experimental setup of reaching task and implemented prototypes for procedural generation of dataset and tool feasibility checking - designed system and architecture figures - contributed to the paper writing, specifically: Introduction, Related Work, Experiments		
Signature	Oliver Groth	Date	28/09/2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Prof. H. Ingmar Posner		
Supervisor comments	This is an accurate account of the candidate's contribution.		
Signature		Date	28 September 2021

This completed form should be included in the thesis, at the end of the relevant chapter.

5

RELATE: Physically Plausible Multi-Object Scene Synthesis Using Structured Latent Spaces

In this chapter we combine recent advances in object-centric, generative modelling and neural physics approximation to devise a model which can learn a probability distribution over plausible scene configurations in 2.5D. We demonstrate that the inductive biases about spatial relationship modelling facilitate proper disentanglement between scene background and salient foreground objects in the model's latent space while still being trainable on raw, unsupervised data. We demonstrate our model's efficacy presenting state-of-the-art performance for scene sampling in simulated and real data domains, physically-plausible scene editing and even the generation of short video snippets preserving the consistency of spatial interactions over time. This work is published as:

S. Ehrhardt, O. Groth, A. Monszpart, M. Engelcke, I. Posner, N. J. Mitra, and A. Vedaldi. "RELATE: Physically Plausible Multi-Object Scene Synthesis Using Structured Latent Spaces". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Dec. 2020

Abstract

We present RELATE, a model that learns to generate physically plausible scenes and videos of multiple interacting objects. Similar to other generative approaches, RELATE is trained end-to-end on raw, unlabeled data. RELATE combines an object-centric GAN formulation with a model that explicitly accounts for correlations between individual objects. This allows the model to generate realistic scenes and videos from a physically-interpretable parameterization. Furthermore, we show that modeling the object correlation is *necessary* to learn to disentangle object positions and identity. We find that RELATE is also amenable to physically realistic scene editing and that it significantly outperforms prior art in object-centric *scene* generation in both synthetic (CLEVR, ShapeStacks) and real-world data (cars). In addition, in contrast to *state-of-the-art* methods in object-centric generative modeling, RELATE also extends naturally to dynamic scenes and generates *videos* of high visual fidelity. Source code, datasets and more results are available at `http://geometry.cs.ucl.ac.uk/projects/2020/relate/`.

5.1 Introduction

We consider the problem of learning to generate plausible images of scenes starting from parameters that are physically interpretable. Furthermore, we wish to learn such a capability from raw images alone, without any manual or external supervision. Image generation is often approached via Generative Adversarial Networks (GAN) (Goodfellow et al. [59]). These models learn to map noise vectors, used as a source of randomness, to image samples. While the resulting images are realistic, the random vectors that parameterize them are not interpretable. To address this issue, authors have recently proposed to *structure* the latent space of deep generative models, giving it a partial physical interpretability (Nguyen-Phuoc et al. [149, 150] and Steenkiste et al. [186]). For example, HoloGAN (Nguyen-Phuoc et al. [149]) samples volumes and cameras to generate 2D images of 3D objects, and BlockGAN (Nguyen-Phuoc et al. [150]) creates scenes by composing multiple objects. The resulting GANs

have been shown to learn concepts such as viewpoint and object disentangling from raw images.

BlockGAN is of particular interest because, via its relatively strong architectural biases, it provides *interpretable* parameters for the scene, incorporating concepts such as position and orientation. However, BlockGAN comes with a significant limitation in that it assumes that objects are mutually *independent*. This approximation is acceptable only when objects interact weakly, but it is badly violated for medium to densely packed scenes, or for scenes such as stacking wooden blocks or cars following a path, where the (object) correlation is strong.

Recent work in object-centric generative modeling has attempted to specifically address this by capturing correlations in latent space (e. g. Engelcke et al. [40] and Steenkiste et al. [186]). However as object state information remains significantly entangled in these models they have, to date, been unable to operate on real-world data.

In this paper, we introduce RELATE, a model which explicitly leverages the strong architectural biases of BlockGAN to effectively model correlations between latent object state variables. This leads to a powerful model class, which is able to capture complex physical interactions, while still being able to learn from raw visual inputs alone. Empirically, we show that only when we model such interactions our GAN model correctly disentangles different objects when they exhibit even a moderate amount of correlation (Figs. 5.4.1 and 5.4.2). Without this component, the model may still generate high fidelity images, but it generally fails to establish a physically-plausible association between the parameters and the generated images. Our results also demonstrate that GANs are surprisingly sensitive to the correlation of objects in natural scenes, and can thus be used to directly learn these *without* resorting to techniques such as variational auto-encoding (Kingma and Welling [101]).

We demonstrate the efficacy of RELATE in several scenarios, including balls rolling in bowls of variable shape (Ehrhardt et al. [37]), cluttered tabletops (CLEVR (Johnson et al. [92])), block stacking (ShapeStacks (Groth et al. [64])), and videos of traffic at

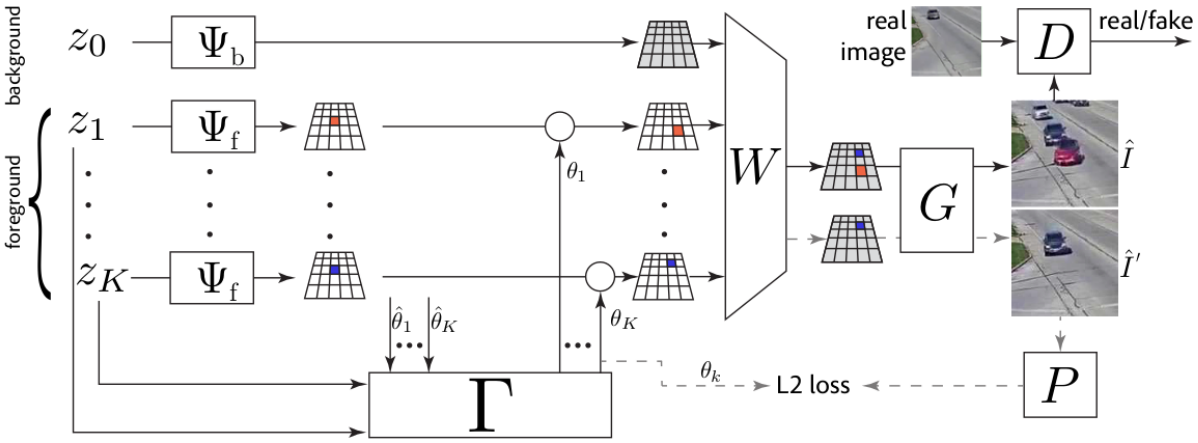


Figure 5.1.1: Image generation using RELATE. Individual scene components, such as background and foreground objects are represented by appearance z_0 and pairs of appearance and pose vectors $(z_i, \theta_i), i \in \{1, \ldots, K\}$, respectively. The key spatial relationship module Γ adjusts the initial independent pose samples $\hat{\theta}_i$ to be physically plausible (e. g. non-intersecting) to produce θ_i. The structured scene tensor W is finally transformed by the the generator network G to produce an image $\hat{I}$. RELATE is trained end-to-end in a GAN setup (D denotes the discriminator) on real unlabelled images.

busy intersection. By ablating the interaction module, we show that modeling the spatial correlation between the objects is key. Furthermore, we compare RELATE to several recent GAN- and VAE-based baselines, including BlockGAN (Nguyen-Phuoc et al. [150]), GENESIS (Engelcke et al. [40]) and OCF (Anciukevicius et al. [6]), in terms of *Fréchet Inception Distance (FID)* (Heusel et al. [78]), and outperform even the best state-of-the-art model by up to 29 points.

Qualitatively, we show that modeling spatial relationships strongly affects scene decomposition and the enforcement of spatial constraints in the generated images. We also show that the physically interpretable latent space learned by RELATE can be used to edit scenes as well as to generate scenes outside the distribution of the training data (e. g. containing more or fewer objects). Finally, we show that the parameterization can be used to generate long plausible video sequences (as measured according to FVD score (Unterthiner et al. [195])) by simulating their dynamics while preserving their spatial consistency.

5.2 Related Work

Interpretable Object-Centric Visual Models. Inspired by the *analysis-by-synthesis* approach for visual perception discussed in cognitive science (Yildirim et al. [227]), recent work (Burgess et al. [25], Engelcke et al. [40], Greff et al. [62], and Steenkiste et al. [186]) propose structured latent space models to explain and synthesize images as sets of constituent components which are individually represented using VAEs (Kingma and Welling [101]) or GANs (Goodfellow et al. [59]). Other approaches favor explicit symbolic representations over distributed ones when parsing an image (Santoro et al. [168] and J. Wu et al. [209]) or propose probabilistic programming languages to formalize image generation (T. D. Kulkarni et al. [113]). In both cases, object-centric modeling allows decomposition of images into components and also enables targeted image modification via interpolation in symbol or latent space, e. g. altering position or colour of an object parsed by the model. Interpretable and controllable factors of image generation are desirable properties for neural rendering models and have been investigated in recent image generation models, e. g. Anciukevicius et al. [6], Liao et al. [123], and Nguyen-Phuoc et al. [149, 150]. However, despite the modeling effort put into the object representations, inter-object interactions are typically only modeled in a less explicit way, e. g. via image layers (Steenkiste et al. [186] and J. Wu et al. [209]), depth ordering variables (Anciukevicius et al. [6]) or an autoregressive *scene prior* (Engelcke et al. [40]). Our work adds to this line of work by proposing a spatial correlation network which facilitates disentanglement of learned object representations and can be trained from raw observations.

Neural Physics Approximation. Harnessing the power of deep learning to approximate physical processes is an emerging trend in the machine learning community. Especially the approximation of rigid-body motions with neural networks already boasts a large body of literature, e. g. P. Battaglia et al. [14], M. B. Chang et al. [27], Fragkiadaki et al. [50], Kossen et al. [110], Van Steenkiste et al. [196], and Watters et al. [204]. Such learned approximations of object interactions have been successfully employed in object manipulation (Janner et al. [90]) and tracking (Fuchs et al. [52] and

Kosiorek et al. [109]). However, most entries in this line of work are only applied to visual toy domains or rely on segmentation masks or bounding boxes to initialize their object representations before the networks approximate the object dynamics. While we leverage ideas from neural dynamics modeling, we go beyond the established scope of visual toy domains such as coloured point masses or moving MNIST digits (LeCun [115]) and learn directly from rich visual data such as simulated object stacks and real traffic videos without further annotation.

Extraction and Generation of Video Dynamics. Videos are a natural choice of data source to learn about the physics of rigid bodies. Physical information extracted from videos can either be explicit such as estimates of velocity or friction (T. Ye et al. [221]) or implicitly represented in the latent space, e. g. sub-spaces corresponding to pose variation (Denton and Birodkar [31]). More recently, object-centric approaches have also been leveraged to acquire better video representations for future frame prediction (Y. Ye et al. [223]) or model-based reinforcement learning (Veerapaneni et al. [198]). Several studies have also attempted to learn entire video distributions as spatio-temporal tensors in GAN frameworks (Kalchbrenner et al. [95], Saito et al. [167], Vondrick et al. [199], and Xue et al. [219]) yielding impressive first results for full video generation in artificial and real domains. In contrast to prior art, our model departs from a monolithic spatio-temporal tensor representation over an entire video. Instead we cast the video learning and generation process as temporal extension of the object-centric representation of a single frame, lowering the computational burden while still faithfully representing long-range dynamics.

5.3 Method

RELATE (Fig. 5.1.1) consists of two main components: An interaction module, which computes physically plausible inter-object and object-background relationships, and a scene composition and rendering module, which features an interpretable parameter space factored into appearance and position vectors. The details are given next.

5.3.1 Physically-interpretable scene composition and rendering

RELATE considers scenes containing up to K distinct objects. The model starts by sampling *appearance parameters* $z_1, \dots, z_K \sim \mathcal{U}([-1,1]^{N_f})$ for each individual foreground object as well as a parameter $z_0 \sim \mathcal{U}([-1,1]^{N_b})$ for the background. These parameters are small noise vectors, similar to the ones typically used in generative networks. Different from the object poses below, they are sampled independently, thus assuming that the appearance of different objects is independent.

For rendering an image, the appearance parameter z_k is first mapped to a tensor $\Psi_k \in \mathbb{R}^{H \times H \times C}$. This is done via two separate learned decoder networks, one for the background $\Psi_0 = \Psi_b(z_0)$ and one for the foreground objects $\Psi_k = \Psi_f(z_k)$. Here H is the horizontal and vertical spatial resolution of the representation (see Table 5.D.1 in supplementary) and C is the number of feature channels (see Tables 5.D.2 and 5.D.3). Since we assume that individual objects are much smaller than the overall scene, we restrict Ψ_k, $k \geqslant 1$ to be non-zero only in a fixed smaller $H' < H$ window in the center of the tensor.

Each foreground object also has a corresponding *pose parameter* θ_k, which is geometrically interpretable. For simplicity, we assume $\theta_k \in \mathbb{R}^2$ to be a 2D translation, acting on the tensor Ψ_k via bilinear resampling:

$$\hat{\Psi}_k = \theta_k \cdot \Psi_k \quad \text{such that} \quad [\hat{\Psi}_k]_u = [\Psi_k]_{u+\theta_k}$$

where $u \in \mathbb{R}^2$ is a spatial index and $[\cdot]_u$ means accessing the column of the tensor in bracket at spatial location u (using padding and bilinear interpolation if u does not have integer coordinates). However, θ_k can easily be extended to represent full 3D transformations as previously shown in BlockGAN (Nguyen-Phuoc et al. [150]).

Foreground and background objects are composed into an overall scene tensor $W \in \mathbb{R}^{H \times H \times C}$ via element-wise max- (or sum-) pooling as $W_u = \max_{k=0,\dots,K} [\hat{\Psi}_k]_u$. In this manner, the scene tensor is a function $W(\Theta, Z)$ of the pose parameters $\Theta := (\theta_1, \dots, \theta_K)$ and the appearance parameters $Z := (z_0, z_1, \dots, z_K)$. Finally, a decoder network $\hat{I} = G(W)$ renders the composed scene as an image (see Table 5.D.4).

Discussion. This model is 'physically interpretable' in the sense that it captures (1) the identities of K distinct objects and (2) their pose parameters as translation vectors. This should be contrasted to traditional GAN models, where the code space is given as an uninterpretable, monolithic noise vector z. Despite the structure given to the code space, there is no guarantee that the model will actually learn to map it to the corresponding structure in the example images. However, we found empirically that this is the case as long as the correlations between the different objects are also captured.

5.3.2 Modeling Correlations in Scene Composition

RELATE departs significantly from prior art such as BlockGAN as it does not assume the parameters θ_i of the different objects to be independent. In order to model correlation, we propose a two-step procedure, based on a residual sampler. First, we sample a vector of K i.i.d. poses $\hat{\Theta} \sim \mathcal{U}([-H''/2, H''/2]^{2K})$ where $H'' < H$ is smaller than the spatial size H of the tensor encoding. Then, we pass this vector to a 'correction' network Γ that remaps the initial configuration to one that accounts for the correlation between object locations and appearances, as well as between objects and the background (coded by the appearance component z_0 in z): $\Theta := \Gamma(\hat{\Theta}, Z)$. In practice, we expect object interactions, as any physical law, to be *symmetric* with respect to the order of the objects. We obtain this effect by implementing Γ as running K copies of the *same* corrective function in parallel:

$$\theta_k = \hat{\theta}_k + \zeta(\hat{\theta}_k, z_k, |z_0, \{z_i, \hat{\theta}_i\}_{i \geqslant 1, i \neq k}). \tag{5.1}$$

The function ζ is implemented in a manner similar to the Neural Physics Engine (NPE) (M. B. Chang et al. [27]):

$$\zeta(\hat{\theta}_k, z_k, |z_0, \{z_i, \hat{\theta}_i\}_{i \geqslant 1, i \neq k}) = f(\hat{\theta}_k, z_k, z_0, h^s_k), \quad h^s_k = \sum_{q \neq k} g(\hat{\theta}_k, z_k, \hat{\theta}_q, z_q), \tag{5.2}$$

where f and g are Multi Layer Perceptrons (MLPs) (tables 5.D.5, 5.D.6) operating on stacked vector inputs and h^s is an embedding capturing the interactions between the K objects. Besides symmetry, an advantage of this scheme is that it can take

an arbitrary number of objects K due to the sum-pooling operator used to capture the interactions. In this manner, the sampler Γ is automatically defined for any value of K. For each scene, K is sampled uniformly from a fixed interval $[K_{min}, K_{max}]$. Furthermore, sampling independent quantities followed by a correction has the benefit of injecting some variance on the objects positions at the early stage of training, which helps to avoid converging to trivial/bad solutions.

Ordered scenes. An advantage of RELATE is that it can be easily modified to take advantage of additional structure in the scene. For scenes where objects have natural order, such as stacks of blocks, we experiment with conditioning pose θ_i on the preceding pose θ_{i-1}, using a Markovian process. This is done by first sampling $\hat{\theta}_1 \sim \mathcal{U}([-H''/2, H''/2])$, and then applying a correction to account for the background z_0 as before, finally sampling the other objects in sequence:

$$\theta_1 = \hat{\theta}_1 + f_0(\hat{\theta}_1, z_1, z_0), \qquad \forall k > 1: \quad \theta_k = \theta_{k-1} + f_1(\theta_{k-1}, z_{k-1}, z_0), \tag{5.3}$$

where f_0, f_1 are implemented as MLPs as before (tables 5.D.8, 5.D.9). Note that this can be interpreted as a special case of the model above in the sense that we can write $\Theta := \Gamma(\hat{\Theta}, Z)$, provided that $\hat{\theta}_k = 0$ for $k \geqslant 2$.

Modeling dynamics. RELATE can also be immediately extended to make dynamic predictions. For this, we sample the initial positions $\theta_k(0)$ as before and then update them incrementally as $\theta_k(t+1) = \theta_k(t) + v_k(t+1)$, where $v_k(t)$ is the object velocity. In order to obtain the latter, we let $V_k(t) = [v_k(t-i)]_{i=2,1,0}$ denote the last three velocities of the k-th object. The initial value $V_k(0) = e_v(z_k, z_0, \theta_k(0))$ is initialized as a function of the appearance parameters and initial positions (Table 5.D.7); and we use the NPE style update equations (M. B. Chang et al. [27]), where e_v, f_v and g_v are MLPs,

$$v_k(t+1) = f_v(\theta_k(t), z_k, V_k(t), z_0, h_k^d(t)), \quad h_k^d(t) = \sum_{q \neq k} g_v(\theta_k(t), z_k, V_k(t), \theta_q(t), z_q, V_q(t)). \tag{5.4}$$

5.3.3 Learning Objective

Training our model makes use of a training set I_i, $i = 1, \ldots, N$ of N images of scenes containing different object configurations. No other supervision is required. Our learning objective is a sum of *two high fidelity losses* and *a structural loss* which we describe below.

For high fidelity, images $\hat{I}$ generated by the model above are contrasted to real images I from the training set using the standard GAN discriminator $\mathcal{L}_{\text{GAN}}(\hat{I}, I)$ and style $\mathcal{L}_{\text{style}}(\hat{I}, I)$ losses from Nguyen-Phuoc et al. [150] (see Section 5.C).

In addition, we introduce a regularizer to encourage the model to learn a non-trivial relationship between object positions and generated images. For this, we train a position regressor network P that, given a generated image $\hat{I}$, predicts the location of the objects in it. In practice, we simplify this task and generate an image $\hat{I}'$ by retaining only object k of the K objects at random and minimizing $\|\check{\theta}_k - P(G(W(z_0, z_k, \theta_k)))\|_2^2$. Here the symbol $\check{}$ means that gradients are not back-propagated through θ_k: this is to avoid mode collapse of the position at zero. P shares most of its weights with the discriminator network (see Table 5.D.10).

In the case of dynamic prediction, the discriminator takes as input the sequence of images concatenated along the RGB dimension and is tasked to discriminate between fake and real sequences. Similar to a static model we also have a position regressor which is tasked to predict the position of an object rendered at random with zero velocity.

5.4 Experiments

Implementation details. We learn mappings Ψ_b and Ψ_f using the same Adaptive Instance Normalization (AdaIN) (X. Huang and Belongie [85]) architecture. The spatial size of their output tensors is set to $H = 16$ and the final output image to 128×128 (which is reduced when needed for fair comparison to other methods). We use the Adam (Kingma and Ba [102]) optimizer for learning and train for a

fixed number of epochs and always select the last model snapshot. We consider two types of baselines: standard generative models such as DCGAN (Radford et al. [159]) and DRAGAN (Kodali et al. [106]), and object-centric generative baselines such as GENESIS (Engelcke et al. [40]) and OCF (Anciukevicius et al. [6]), quoting results from the original papers whenever possible. In addition we also add BlockGAN2D as an ablation of our method.

Datasets. We conduct experiments on four different datasets. First, we consider a relatively simple dataset, BallsInBowl (Ehrhardt et al. [37]), for assessing the model features and ablations. It consists of videos of two distinctly coloured balls rolling in an elliptical bowl of variable orientation and eccentricity. Interactions amount to object collisions and the fact that they must roll within the bowl. To this, we add two popular synthetic datasets CLEVR (Johnson et al. [92]) (cluttered tabletops) and ShapeStacks (Groth et al. [64]) (block stacking). Finally, we have collected a new dataset RealTraffic containing five hours of footage of a busy street intersection, divided into fragments containing from one to six cars. Especially the last dataset contains many interactions between the individual cars as they adapt their speed to the surrounding traffic which happens frequently when the light changes and cars either slow down because of a queue on red or accelerate when the lights change to green again. Further details about training, evaluation and datasets can be found in the appendix, sections 5.D, 5.E and 5.F.

5.4.1 Generating Static Scenes

Ablation study. We start experimenting with the comparatively simple BallsInBowl dataset to conduct basic ablations. The first ablation removes the spatial correlation module Γ and the position regression loss, therefore reducing RELATE to a 2D version of BlockGAN. We also consider *'w/o residual'*, where the addition $\hat{\theta}_k$ in Eq. (5.1) is removed, and *'w/o pos. loss'*, where the position regression loss regularizer is removed. Table 5.4.1 shows that each component of RELATE yields an improvement in terms of FID scores on this dataset supporting our spatial modeling decisions. Furthermore,

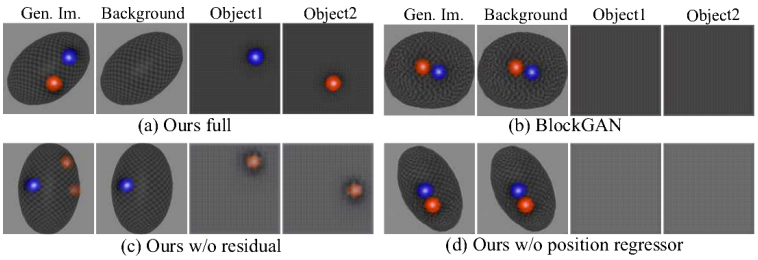


Figure 5.4.1: Ablation Study. For every case we render every component of our method independently. We show that only our full model is able to correctly disentangle individual components of the scene.

Table 5.4.1: Ablation study. FID score (lower is better) on BallsInBowl. Ours (full) reaches the highest fidelity by a large margin.

BlockGAN 2D	152.3
Ours w/o residual	133.9
Ours w/o pos. reg.	154.8
Ours (full)	**81.9**

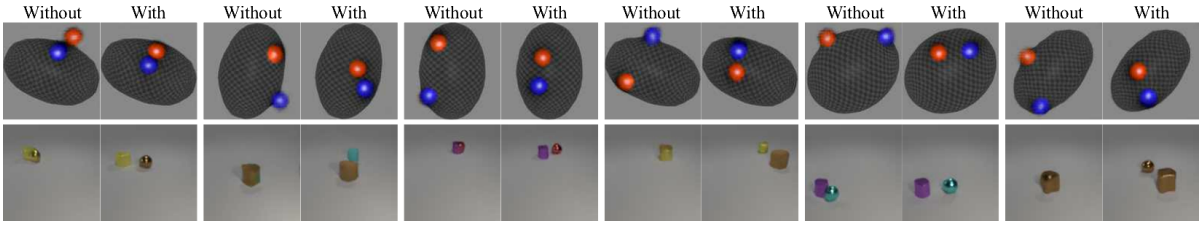


Figure 5.4.2: Effect of the interaction module Γ. We show pairs of images without and with the correction function Γ is applied. For BallsInBowl the correction moves the balls within the bowl, and for CLEVR it pushes apart intersecting objects.

in Fig. 5.4.1 we show qualitatively that only RELATE (a) is able to correctly disentangle the underlying scene factors. We do this by generating the same image while retaining a single factor, which correctly isolates the background, and, in turn, both individual objects. BlockGAN 2D (b) and *'ours w/o pos. reg.'* (d) fail to disentangle the factors entirely, mapping everything to the background component[1]. *'Ours w/o residual'* (c) shows that the model partially fails to disentangle, with the background encoding some but not all the objects. Finally, Fig. 5.4.2 visualises the effect of the interaction module Γ. Recall that this is implemented as a 'correction' function that accounts for correlation starting from independently-sampled parameters. For BallsInBowl, the correction module moves the balls within the bowl, and for the CLEVR it pushes objects apart if they intersect.

Quantitative evaluation. In Table 5.4.2, we compare RELATE to existing scene generators on ShapeStacks, CLEVR and RealTraffic. We report performance in

[1]Additional quantitative information about the difference in object disentanglement between our model and BlockGAN 2D is provided in Section 5.A.1.

RELATE variant	CLEVR-5 General	CLEVR-5vbg General	CLEVR General	ShapeStacks Ordered	RealTraffic General
DCGAN†	264.8	361.8	247.8	197.6	47.6
DRAGAN†	80.8	84.4	108.0	**57.2**	**38.8**
OCF	N/A	83.1	N/A	N/A	N/A
GENESIS	211.7	169.4	151.3	233.0	167.1
BlockGAN2D ‡	63.0	53.3	78.1	99.3	57.9
Ours	**58.4**	**36.4**	**62.9**	**95.8**	**42.0**
Ours + scale	**37.4**	**35.9**	**44.9**	**79.1**	46.8

Table 5.4.2: Comparison to state-of-the-art methods. FID score (lower is better) for various datasets. We consistently outperfom prior art in object centric scene generation. 'Ordered' refers to the variant discussed in sec. 3.2. †are standard GANs which FID score are evaluated on 64×64 images for 'General' variant (see Section 5.D).‡is the 2D variant of BlockGAN (Nguyen-Phuoc et al. [150]) which sometimes fails to be object-centric (see Section 5.A).

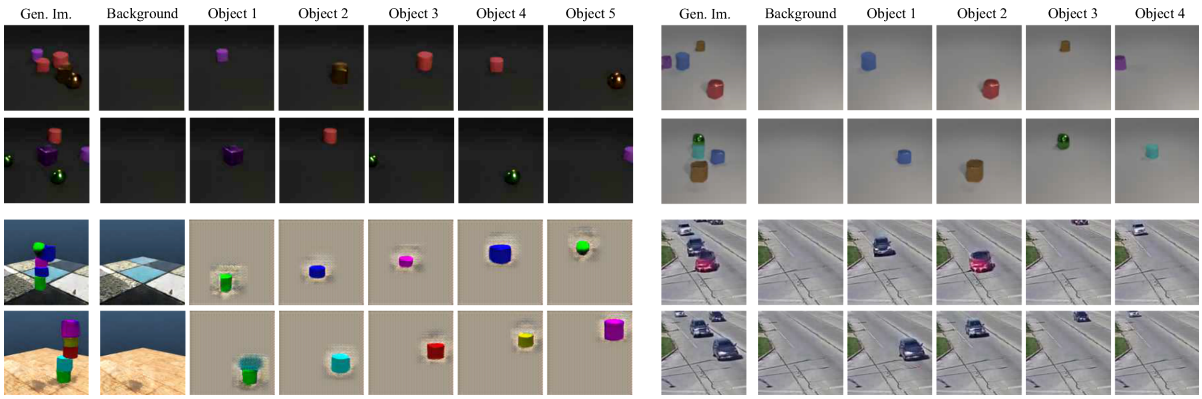


Figure 5.4.3: Component-wise scene generation. From a generated image (left) RELATE can render each component individually for each dataset. For CLEVR and RealTraffic objects are rendered after being composed with the background (cf. Section 5.4.2). Top left picture has increased contrast for easier visualization.

terms of FID score (Heusel et al. [78]) computed between 10,000 images sampled from our model and the respective test sets. For CLEVR, we train RELATE and BlockGAN on a restricted version of the data containing from three to six objects in an image[2], and at test time, we require all models to sample images with three to ten objects. We consistently outperform all prior object-centric methods in all scenes and scenarios according to FID scores. In particular, on CLEVR, RELATE can generate a larger

[2]Note that GENESIS was trained on the full training set featuring three to ten objects.

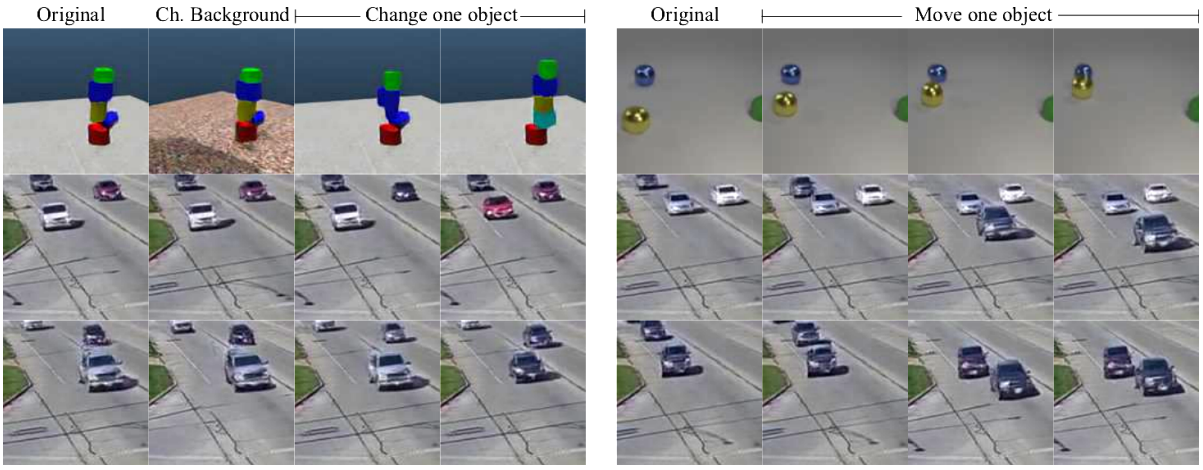


Figure 5.4.4: Image editing. Left: We demonstrate the capacity of RELATE to change the background and the appearance of individual objects. Right: RELATE is also able to modify the position of a single object.

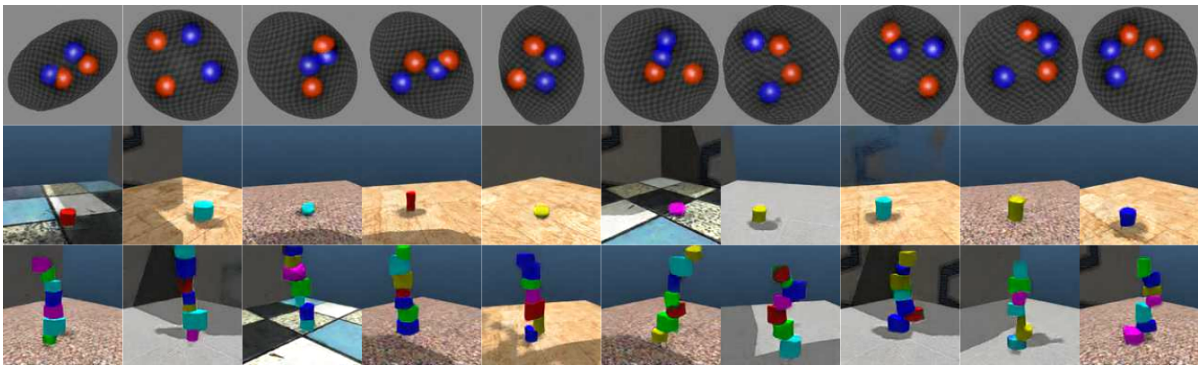


Figure 5.4.5: Out-of distribution generation. RELATE can generate images outside the training distribution. The first row shows generating a bowl with four balls, whereas the training set only features exactly two. The last two rows depict towers of one and seven objects, whereas the training images only had stacks of height two to five.

number of objects than seen during training suggesting its improved generalisation capabilities which are demonstrated further in Fig. 5.4.5. Our method also consistently out-performs standard GANs on all CLEVR datasets and is on par with DRAGAN on RealTraffic. More qualitative results can be found in Section 5.G.

5.4.2 Interpretability of the Latent Space and Scene Editing

As shown in the ablation studies in Fig. 5.4.1, RELATE successfully disentangles a scene into independent components - in contrast to BlockGAN2D which struggles to separate individual objects from the background. Figure 5.4.3 shows that RELATE

can *disentangle* also far more complex scenes in RealTraffic, ShapeStacks and CLEVR. Note that for RealTraffic and CLEVR we render objects composed with the background. In fact, in these datasets the size and appearance of each object is correlated to their position in the background because of the camera perspective. In addition to qualitative results, we also compute a disentanglement score in Table 5.A.1 which measures how well our model is disentangling individual components of the scene. We found that our model manages to consistently separate each individual objects of the scene and outperform BlockGAN2D on the most challenging datasets which is in line with the qualitative evidence we observe. Next, in Fig. 5.4.4 we use RELATE to *edit* a generated scene. For example we can change the position or appearance of individual objects. Finally, we show that RELATE can generate *out-of-distribution* scenes. This is achieved in particular by sampling a different number of objects. In Fig. 5.4.5, for instance, RELATE is trained on ShapeStacks seeing towers of height two to five. However, it can render taller towers of up to seven objects, or even just a single object. Likewise, in BallsInBowl it can generate bowls with four balls having seen only two during training. Furthermore, in ShapeStacks each tower is composed of blocks of *different* colours, but RELATE can relax this constraint rendering objects with repeated colours.

5.4.3 Simulating Dynamics

We train this model on BallsInBowl and RealTraffic to predict 15 and 10 consecutive frames respectively. During generation, we sample videos with a sequence length of 30 frames and measure the faithfulness with respect to the distribution of the test data via the *Fréchet Video Distance (FVD)* (Unterthiner et al. [195]). We achieve FVD scores of 556 and 2253 respectively. This is perceptibly better than 920 and 3370 for a baseline consisting of time-shuffled sequences from the respective training sets, which feature perfect resolution but poor dynamics. Qualitatively in Fig. 5.4.6 we see that the model does understand the motion and captures interaction with the background. For instance, in BallsInBowl the balls do have a curved motion because of the shape of the bowl and decrease in speed when reaching the edges of the bowl which are in

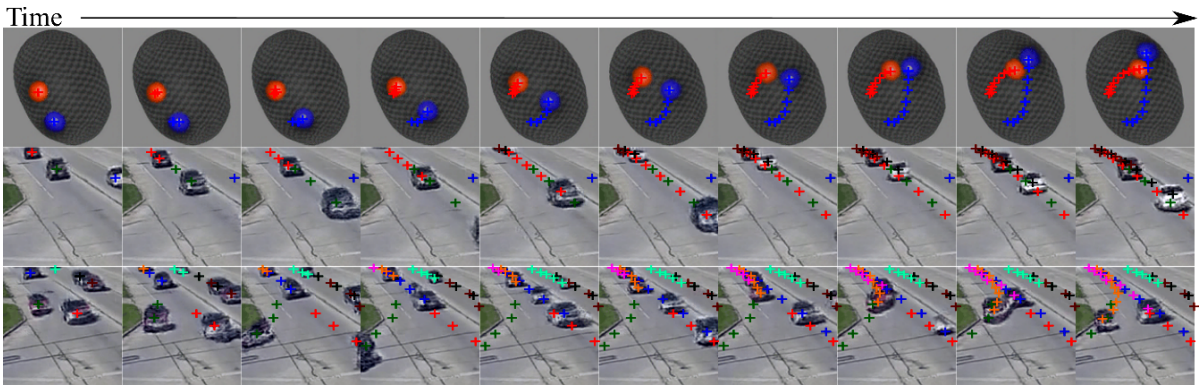


Figure 5.4.6: Video generation. We show consecutive video frames generated by RELATE overlayed with crosses representing projections of the model's estimated pose parameters for each object. In BallsInBowl the interaction with the environment is well captured as the balls stay within the bowl. In RealTraffic the cars stay in their lane, or can decide to make a right turn (last row).

Figure 5.4.7: Failure case. Our model struggles to understand big changes of aspect ratio. For instance, in the static case, when we drag two cars towards the bottom of the image, we can see that the left grey car disappears at some point (highlighted in red) and reappears on the edge of the image. This explains why the video generation on RealTraffic fails to capture the true data distribution more faithfully resulting in lower FVD score. This effect can however be mitigated with a scale augmented model (see Section 5.4.3).

higher position (see first row of Fig. 5.4.6). In RealTraffic the cars do stay in their respective lane. Interestingly our model is able to handle different types of motions correctly (see third row Fig. 5.4.6) and uses the sample vector to decide whether the cars should go straight or turn. Finally, we see that we can also generate videos with much more cars than the upper bound (5) with which the system was trained (see last row Fig. 5.4.6).

Limitations While the dynamics in video generation look realistic in most cases, RealTraffic also exposed the limitations of our approach. In this dataset the perspective range of the camera is important. As a result, cars at the bottom of the image, which appear bigger in the training data, often do not get generated properly in the static case (see highlighted frames in Fig. 5.4.7). We hypothesize that this is also

the main reason why the cars' appearance (and hence FVD score) deteriorates in the dynamic scenarios: since the style parameters z_i are fixed to preserve identity, it is not possible for the model to account accurately for the appearance change introduced by large changes of perspective over the course of a sequence. To verify our hypothesis in Section 5.A.2 we propose a simple modification of our pipeline that accounts for scale changes in an image. This modification simply allows the network to predict H' for each individual object instead of it being a constant. We found that in general this allows our network to train on datasets with more important range of scales such as CLEVR3 (Anciukevicius et al. [6]) and renders objects at more different scales for CLEVR and RealTraffic (see Fig. 5.A.1). Besides in Table 5.4.2 we show that this modification almost always results in higher FID scores from our main model as well as a significant boost of 397 points in FVD score for RealTraffic (1855 vs 2253).

5.5 Conclusion

We have introduced RELATE, a GAN-based model for object-centric scene synthesis featuring a physically interpretable parameter space and explicit modeling of spatial correlations. Our experimental results suggest that spatial correlation modeling plays a pivotal role in disentangling the constituent components of a visual scene. Once trained, RELATE's interpretable latent space can be leveraged for targeted scene editing such as altering object positions and appearances, replacing the background or even inserting novel objects. We demonstrate our model's effectiveness by presenting *state-of-the-art* scene generation results across a variety of simulated and real datasets. Lastly, we show how our model naturally extends to the generation of dynamic scenes being able to generate entire videos from scratch. A main limitation of our current model is its restriction to planar motions which prevents it from representing arbitrary 3D motions featuring angular rotation more faithfully, most notably highlighted by the experiments for video generation. This effect can however be partially mitigated by a scale-augmented model which we introduce in the appendix. We believe that our work can contribute to future research in object-centric scene representation

by providing a scalable, spatio-temporal modeling approach which is conveniently trainable on unlabeled data.

Broader Impact

Our method advances the ability of computers to learn to understand environments in images in an object-centric way. It also enhances the capabilities of generative models to generate realistic images of "invented" environment configurations.

Overall, we believe our research to be at low to no risk of direct misuse. At present, our generation results are insufficient to fool a human observer. However, it has to be noted that the sampling process is, as in many other deep generative models, capable of revealing patterns observed in the training data, e. g. specific textures or object geometries. Such data privacy concerns are not applicable in the street traffic data used in our research, since the resolution of the videos is far too low to identify individual drivers or recognize cars' license plates. However, 'training data leakage' should be taken into consideration when the model is trained on more sensitive datasets.

In a positive prospect, we believe that our model contributes to further the development of less opaque machine learning models. The explicit object-centric modelling of image components and their geometric relationships is in many of its aspects intelligble to a human user. This facilitates debugging and interpreting the model's behaviour and can help to establish trust towards the model when employed in larger application pipelines.

However, the key value of our paper is in the methodological advances. It is conceivable that, like any advance in machine learning, our contributions could ultimately lead to methods that in turn can and are misused. However, there is nothing to indicate that our contributions facilitate misuse in any direct way; in particular, they seem extremely unlikely to be misused directly.

Acknowledgments

This work is supported by the European Research Council under grants ERC 638009-IDIU, ERC 677195-IDIU, and ERC 335373. The authors acknowledge the use of Hartree Centre resources in this work. The STFC Hartree Centre is a research collaboratory in association with IBM providing High Performance Computing platforms funded by the UK's investment in e-Infrastructure. The authors also acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (`http://dx.doi.org/10.5281/zenodo.22558`). Special thanks goes to Olivia Wiles for providing feedback on the paper draft, Thu Nguyen-Phuoc for providing the implementation of BlockGAN and Titas Anciukevičius for providing the generation code for CLEVR variants. We finally would like to thank our reviewers for their diligent and valuable feedback on the initial submission of this manuscript.

Appendix

In this supplementary material we provide further details about RELATE. The appendix is organised as follows: First, we provide an additional object decomposition score and a description of the scale experiment in Section 5.A. Further discussions about the method are presented in Section 5.B. Details on the exact loss functions used can be found in Section 5.C. In Section 5.D we elaborate on the model's implementation details. Section 5.E is dedicated to explaining the details of the baselines and their respective training protocols. Section 5.F contains a thorough explanation of every dataset and the data collection procedure where applicable. Finally, we provide more qualitative results in Section 5.G.

5.A Additional Experiments

5.A.1 Disentanglement Study

	CLEVR-5 General	CLEVR-5vbg General	CLEVR General	ShapeStacks Ordered	RealTraffic General	BallsInBowl General
BG2D	19.0	18.0	18.0	272.0	22.0	98.0
Ours	17.0	17.0	19.0	17.0	23.0	26.0

Table 5.A.1: Disentanglement score. For each dataset of Table 5.4.2 we report respectively the distance and correlation score described in Section 5.A. Our model outperforms BlockGAN2D (BG2D) in the most complex scenario: ShapeStacks and BallsInBowl. Both model reach similar scores for the other scenes.

We have conducted additional experiments to provide more quantitative insights of the disentanglement capabilities of our model. While measures such as MIG (Locatello et al. [129]) are typically used to quantify disentanglement, computing this score is not applicable in our case since our model does not feature an inference component

to compute the posterior $q(z|x)$. Hence, we have devised a proxy procedure: We toggle each object of an image individually (out of 5 objects generated) and measure how the generated image changes. We report the distance between the pixel location corresponding to the maximum image change and the location (scaled θ_i) of the object that was toggled. In Table 5.A.1 we report the median distance between θ_i and the pixel location corresponding to the maximum image change. We note that our model generally outperforms BlockGAN2D, most notably for ShapeStacks where BlockGAN is not able to disentangle different objects at all. In addition, we note that for the model trained on ShapeStacks, the discriminator can predict the position of a stack's base object with 11.3 mean pixel error on the test set.

5.A.2 Scale Experiment

In order to tackle the limitation discussed in Section 5.4, we propose to augment our model with scale prediction. Practically, this translates to predicting H' for each individual object instead of keeping it fixed. Therefore, we now assign H'_k instead of H' to each foreground component of the scene. H'_k is computed by a module sc following the equation: $H'_k = H' \times (1 + sc(z_0, \theta_k, z_k))$. More details on sc can be found in Table 5.A.2. We summarize all hyperparameters used for the training of the model in Table 5.A.3. For evaluation, we sample z_0 from $\mathcal{U}([-1, 1]^{N_b})$, except for the RealTraffic video model where we use $\mathcal{U}([-0.5, 0.5]^{N_b})$, a range better suited for optimal background fidelity on this dataset.

Layer name	Layer Type	Input size	Output size	Activation
FCsc_1	Linear	$N_f + 2 + N_b$	32	LeakyReLU
FCsc_2	Linear	32	32	LeakyReLU
FCsc_3	Linear	32	1	Tanh

Table 5.A.2: Network architecture for module sc.

5.B Further Discussions

As noted in the conclusions, RELATE is limited to 2D representations. However, our relaionship module is generic enough to be exported to 3D and could be

Dataset	Learning rate	Epoch nums	M	$K_{min} - K_{max}$	H′	N_b	N_f	H″/H sampling range
CLEVR3	0.0001	40	2	2-3	6	1	20	$[-0.6, 0.6]^2$
CLEVR5	0.0001	40	2	2-5	4	1	20	$[-0.6, 0.6]^2$
CLEVR5-vbg	0.0001	40	2	2-5	4	1	20	$[-0.6, 0.6]^2$
CLEVR	0.0001	40	2	3-6	6	1	20	$[-0.6, 0.6]^2$
ShapeStacks	0.001	30	2	2-5	4	5	20	$[-0.6, 0.6] \times [0, 0.6]$
RealTraffic	0.0001	20	2	1-5	6	1	20	$[-0.6, 0.6]^2$

Table 5.A.3: Hyperparameters for each datasets for the scale augmented model. Epoch nums are the number of epochs we trained for. Model is described in Section 5.A.2

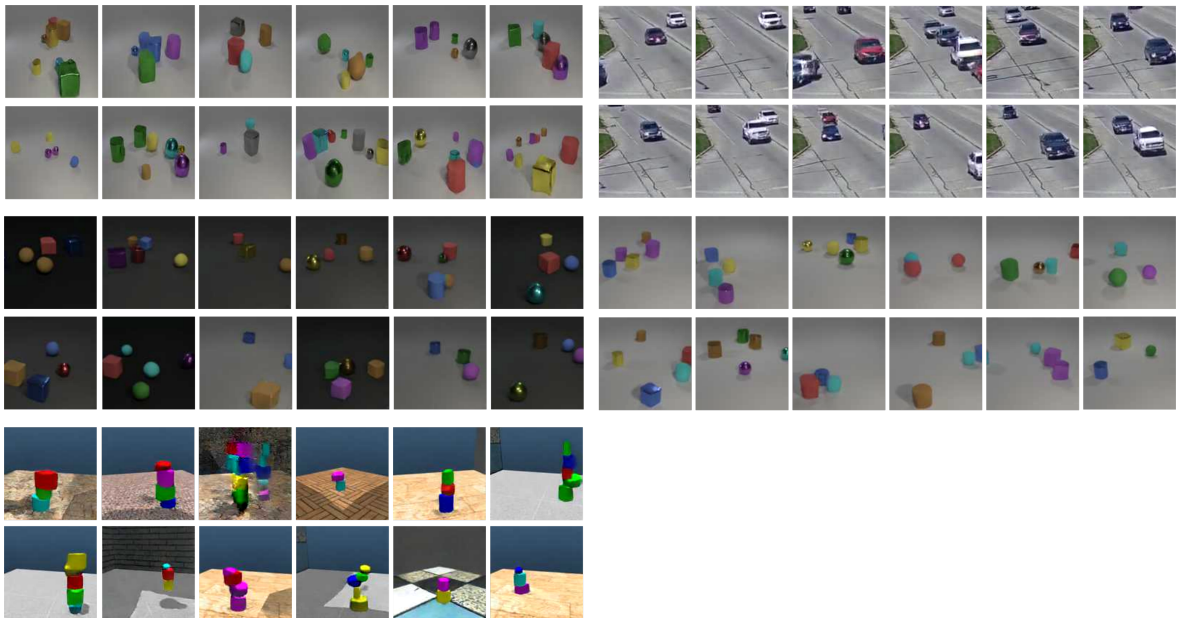


Figure 5.A.1: Samples from the scale augmented RELATE. From left to right and top to bottom we display CLEVR, RealTraffic, CLEVR-5vbg, CLEVR-5, ShapeStacks. Predictions on CLEVR comprise more variety of object sizes compared to our main model (see Fig. 5.G.6). Similarly on RealTraffic we can see that cars can be rendered at arbitrary points in the road.

inserted directly into BlockGAN (Nguyen-Phuoc et al. [150]) – albeit at the expense of significant additional training time. Finally, we acknowledge that the simplified spatial representation of an object as its centroid position is inferior to other object-centric models which predict full object masks, e. g. Anciukevicius et al. [6], Engelcke et al. [40], and Greff et al. [62]. While we believe that an object segmentation mechanism could be easily added based on the already existing rendering of individual latent variables (cf. Fig. 5.4.3), we conjecture that the simplicity of the centroid representation greatly facilitates the learning of spatial correlations within Γ. In contrast, using object

instance masks as spatial representations might introduce unnecessary complications for the neural physics approximation.

5.C Losses

Our final loss is the sum of three losses mentioned in the main text:

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{GAN}}(\hat{I}, I) + \mathcal{L}_{\text{style}}(\hat{I}, I) + \min_{G,\Gamma,P} \|\check{\theta}_k - P(G(W(z_0, z_k, \theta_k)))\|_2^2,$$

where $\mathcal{L}_{\text{GAN}}(\hat{I}, I)$ is the standard GAN loss:

$$\mathcal{L}_{\text{GAN}}(\hat{I}, I) = \min_{G,\Gamma} \max_{D} \mathbb{E}[\log(1 - D(G(W(Z, \Theta))))] + \mathbb{E}[\log(D(I))]$$

The style loss follows the implementation of BlockGAN (Nguyen-Phuoc et al. [150]). The input of the style discriminator D_l are mean μ_l and variance σ_l^2 across spatial dimensions of $\Phi_l(x) \in \mathcal{R}^{W_l \times H_l \times C_l}$, the output of the lth layer of D taken before the normalization step:

$$\mu_l(\Phi_l(x)) = \frac{1}{W_l \times H_l} \sum_i \sum_j \Phi_l(x)_{i,j},$$

$$\sigma_l^2(\Phi_l(x)) = \frac{1}{W_l \times H_l} \sum_i \sum_j (\Phi_l(x)_{i,j} - \mu_l(\Phi_l(x)))^2.$$

The style discriminator D_l for each layer is then implemented as a linear layer followed by a sigmoid activation function. The resulting style loss is:

$$\mathcal{L}_{\text{style}}(\hat{I}, I) = \max_{D} \sum_l \mathbb{E}[\log(1 - D_l(\hat{I}))] + \mathbb{E}[\log(D_l(I))]$$

5.D Implementation Details

Infrastructure and framework. For all experiments we use PyTorch 1.4. We train all models on a single NVIDIA Tesla V100 GPU.

Training hyperparameters. We initialize all weights (including instance normalization ones) by drawing from a random normal distribution $\mathcal{N}(0, 0.02)$. All biases were initialized to 0. For each update of the discriminator we update the generator M number of times. We use Adam parameters $(\beta_1, \beta_2) = (0., 0.999)$ for all datasets except BallsInBowl where $\beta_1 = 0.5$. Similarly W was a max-pooling operator in all datasets except BallsInBowl where we used a sum pooling operator. As in BlockGAN (Nguyen-Phuoc et al. [150]), background and foreground decoders each start from a learned constant tensors T_b and T_f respectively with sizes $H \times H \times 256$ and $H' \times H' \times 512$. For BallsInBowl we use a tensor T_f for each object and use a constant style vector of one.

In the case of dynamic scenarios we reuse same hyperparameters as in the static case except that we use a learning rate of 0.0001 and $\beta_1 = 0$.

Full details of the parameters for each dataset can be found in Table 5.D.1

5.D.1 Evaluation Details

For FID scores computation we draw 10 000 samples from our model which we compare against the same number of images drawn from the test set. To compute FVD score on each dataset, we sample 500 videos of 30 frames from our model and compare them against the videos of the respective test sets (500 for BallsInBowl and 275 videos for RealTraffic). This also applies to the time shuffled baseline.

To be able to compare with other methods we resize our generated images to 96×96 on CLEVR5 and CLEVR5-vbg and 64×64 for ShapeStacks. For the simple generative baselines, DRAGAN and DCGAN we evaluate FID score on the generated 64×64 images from these models. We evaluate on the generated 128×128 images otherwise.

We empirically found that background was rendered with better quality for lower values of z_0. Hence at test time we sampled z_0 from $\mathcal{U}([-0.5, 0.5]^{N_b})$ for optimal results.

Dataset	Learning rate	Epoch nums	M	$K_{min} - K_{max}$	H'	N_b	N_f	H''/H sampling range
BallsInBowl	0.001	60	1	2-2	8	3	1	$[-0.8, 0.8]^2$
CLEVR5	0.0001	40	2	2-5	4	1	90	$[-0.6, 0.6]^2$
CLEVR5-vbg	0.0001	30	2	2-5	4	1	90	$[-0.6, 0.6]^2$
CLEVR	0.0001	40	2	3-6	4	1	90	$[-0.6, 0.6]^2$
ShapeStacks	0.001	30	2	2-5	4	12	64	$[-0.6, 0.6] \times [0, 0.6]$
RealTraffic	0.0001	20	2	1-5	6	1	20	$[-0.6, 0.6]^2$

Table 5.D.1: Hyperparameters for each datasets. Epoch nums are the number of epochs we trained for.

5.D.2 Architecture Details

Generator. In this work we maintain the core of our architecture fixed as much as possible. Since the dimension of the sample z_i does not necessarily match the channel dimension where it is injected before applying Adaptive Instance Normalisation (AdaIN) to a layer l we map z_i to a vector $\hat{z}_i$ transformed such that

$$\hat{z}_i = \max(W_l^T z_i + b_l, 0)$$

Where (W_l, b_l) are learnable parameters. AdaIN is applied at the end of the layers (after the activation). All LeakyReLU layers are using a parameter of 0.2.

Layer name	Layer Type	Input size	Output size	Kernel Size	Stride	Activation	Norm.
Style_f	Id	$H' \times H' \times 512$	$H' \times H' \times 512$	-	-	Id	AdaIn
Convtf_1	ConvTranspose	$H' \times H' \times 512$	$H' \times H' \times 512$	3×3	1	LeakyReLU	AdaIn
Convtf_2	ConvTranspose	$H' \times H' \times 512$	$H' \times H' \times 256$	3×3	1	LeakyReLU	AdaIn
Pad	Padding	$H' \times H' \times 512$	$H \times H \times 256$	-	-	-	-

Table 5.D.2: Network architecture for the foreground object generator Ψ_f.

Layer name	Layer Type	Input size	Output size	Kernel Size	Stride	Activation	Norm.
Style_b	Id	$H \times H \times 256$	$H \times H \times 512$	-	-	Id	AdaIn
Convtb_1	ConvTranspose	$H \times H \times 512$	$H \times H \times 512$	3×3	1	LeakyReLU	AdaIn
Convtb_2	ConvTranspose	$H \times H \times 512$	$H \times H \times 256$	3×3	1	LeakyReLU	AdaIn

Table 5.D.3: Network architecture for the background object generator Ψ_b.

Layer name	Layer Type	Input size	Output size	Kernel Size	Stride	Activation
Ψ_f	-	$H' \times H' \times 512$	$16 \times 16 \times 256$	-	-	-
Ψ_b	-	$H \times H \times 512$	$16 \times 16 \times 256$	-	-	-
W	Max/Sum Pool	$(K+1) \times 16 \times 16 \times 256$	$16 \times 16 \times 256$	-	-	-
Convtg_1	ConvTranspose	$16 \times 16 \times 256$	$32 \times 32 \times 128$	4×4	2	LeakyReLU
Convtg_2	ConvTranspose	$32 \times 32 \times 128$	$64 \times 64 \times 64$	4×4	2	LeakyReLU
Convtg_3	ConvTranspose	$64 \times 64 \times 64$	$64 \times 64 \times 64$	3×3	1	LeakyReLU
Convtg_4	ConvTranspose	$64 \times 64 \times 64$	$128 \times 128 \times 64$	4×4	2	LeakyReLU
Convtg_5	ConvTranspose	$128 \times 128 \times 64$	$128 \times 128 \times 3$	3×3	1	Tanh

Table 5.D.4: Network architecture for the generator G. Outputs of all K foreground object generators Ψ_f (cf. Table 5.D.2) and background generator Ψ_b (cf. Table 5.D.3) are stacked on the first dimension before entering layer *W* (third row).

Layer name	Layer Type	Input size	Output size	Activation
FCf_1	Linear	$2 \times (N_f + 2 + 2^*)$	32	LeakyReLU
FCf_2	Linear	32	32	LeakyReLU
FCf_3	Linear	32	32	None

Table 5.D.5: Network architecture for module f and f_v. * indicates modification of f_v

Table 5.D.8: Network architecture for module f_0.

Layer name	Layer Type	Input size	Output size	Activation
FCf_0_1	Linear	$N_f + N_b + 2$	128	LeakyReLU
FCf_0_2	Linear	128	64	LeakyReLU
FCf_0_3	Linear	64	2	Tanh

Table 5.D.9: Network architecture for module f_1.

Layer name	Layer Type	Input size	Output size	Activation
FCf_1_1	Linear	$N_f + N_b$	128	LeakyReLU
FCf_1_2	Linear	128	64	LeakyReLU
FCf_1_3	Linear	64	2	None
Pos_{out}	Sigmoid(x) Tanh(y)	2	2	None

Discriminator We describe the architecture of the discriminator network in more details in Table 5.D.10. We use spectral normalization (Miyato et al. [138]) at almost every layer. Positions are directly regressed from the last feature output of the discriminator (see last line P_{end}). Therefore in practice P and D share the same backbone D_b (see table Table 5.D.10 until flatten) for every image I:

$$P(I) = P_{end}(D_b(I)), \qquad D(I) = Disc(D_b(I)).$$

Input for style discriminator are taken after the convolution of (Convd_2, Convd_3, Convd_4, Convd_5) in Table 5.D.10 before the normalization. Spectral Normalization

Layer name	Layer Type	Input size	Output size	Activation
FCg_1	Linear	$32 + N_f + 2 + 2^* + N_b$	32	LeakyReLU
FCg_2	Linear	32	32	LeakyReLU
FCg_3	Linear	32	2	Tanh

Table 5.D.6: Network architecture for module g and g_v. * indicates modification of g_v

Layer name	Layer Type	Input size	Output size	Activation
FCe_v_1	Linear	$N_f + 2 + N_b$	128	LeakyReLU
FCe_v_2	Linear	128	128	LeakyReLU
FCe_v_3	Linear	128	3×2	Tanh

Table 5.D.7: Network architecture for module e_v.

was *not* applied to any D_l.

Layer name	Layer Type	Input size	Output size	Kernel Size	Stride	Activation	Norm.
Convd_1	Conv	$128 \times 128 \times 3$	$64 \times 64 \times 64$	5×5	2	LeakyReLU	-
Convd_2	Conv	$64 \times 64 \times 64$	$32 \times 32 \times 128$	5×5	2	LeakyReLU	IN/SN
Convd_3	Conv	$32 \times 32 \times 128$	$16 \times 16 \times 256$	5×5	2	LeakyReLU	IN/SN
Convd_4	Conv	$16 \times 16 \times 256$	$8 \times 8 \times 512$	5×5	2	LeakyReLU	IN/SN
Convd_5	Conv	$8 \times 8 \times 512$	$4 \times 4 \times 1024$	5×5	2	LeakyReLU	IN/SN
Flatten	Id	$4 \times 4 \times 1024$	$1 \times 1 \times 16384$	-	-	-	-
Disc	Linear	$1 \times 1 \times 16384$	1	-	-	Sigmoid	None/SN
P_{end}	Linear	$1 \times 1 \times 16384$	2	-	-	Tanh	None/SN

Table 5.D.10: Network architecture for the discriminators. Note that the instance normalization (IN) weights were also subjected to spectral normalization (SN). P and D shares weights until Flatten layer.

5.E Baselines

DCGAN (Radford et al. [159]) and DRAGAN (Kodali et al. [106]). We used an online pytorch implementation[3] with default hyperparameters. We trained these models to generate 64×64 images and therefore only evaluated FID score at the same resolution (see Section 5.D).

OCF. OCF results were copied from the original paper of Anciukevicius et al. [6].

[3] https://github.com/LynnHo/DCGAN-LSGAN-WGAN-GP-DRAGAN-Pytorch

BlockGAN2D. We use the same hyperparameters and network architecture as RELATE except for learning rate and M. In all cases we report the best results over models trained with variations of learning rate in (0.001, 0.0001) and M in (2,3).

GENESIS. We use the official implementation[4] of GENESIS for all experiments. For the ShapeStacks dataset, we use the official model snapshot released with the original paper[5]. For all other datasets, we train GENESIS for 500,000 iterations with the default learning parameters and select the last model checkpoint for evaluation. When training GENESIS we use *constrained ELBO optimization* (Rezende and Viola [160]) controlled via `g_goal` in the training script which influences the decomposition capability of GENESIS. We perform a grid search over `g_goal` in the range of 0.5635 to 0.5655 and select the model with the lowest ELBO after 500,000 iterations.

5.F Datasets

BallsInBowl. This dataset is a replica of the two balls synthetic dataset of Ehrhardt et al. [37]. It consists of 2500 training sequences and 500 test sequences of two balls of different fixed colour rolling in bowls of various shapes. We count an epoch as 10,000 iterations over the data. In Fig. 5.F.1 we show some sample data from this dataset.

CLEVR. We used the official CLEVR from Johnson et al. [92]. We train on data from train and validation set and evaluate on the test set. Both ours and BlockGAN2D were trained on the subset containing 3 to 6 objects and evaluated on the entire test set.

CLEVR5/CLEVR5-vbg. We use online code provided by the authors[6] to generate CLEVR5 and CLEVR5-vbg. As done in Anciukevicius et al. [6] we generate 100,000 images keep 90,000 for training and 10,000 for testing.

[4] `https://github.com/applied-ai-lab/genesis`

[5] `https://drive.google.com/drive/folders/1uLSV5eV6Iv4BYIyh0R9DUGJT2W6QPDkb?usp=sharing`

[6] `https://github.com/TitasAnciukevicius/clevr-dataset-gen.`

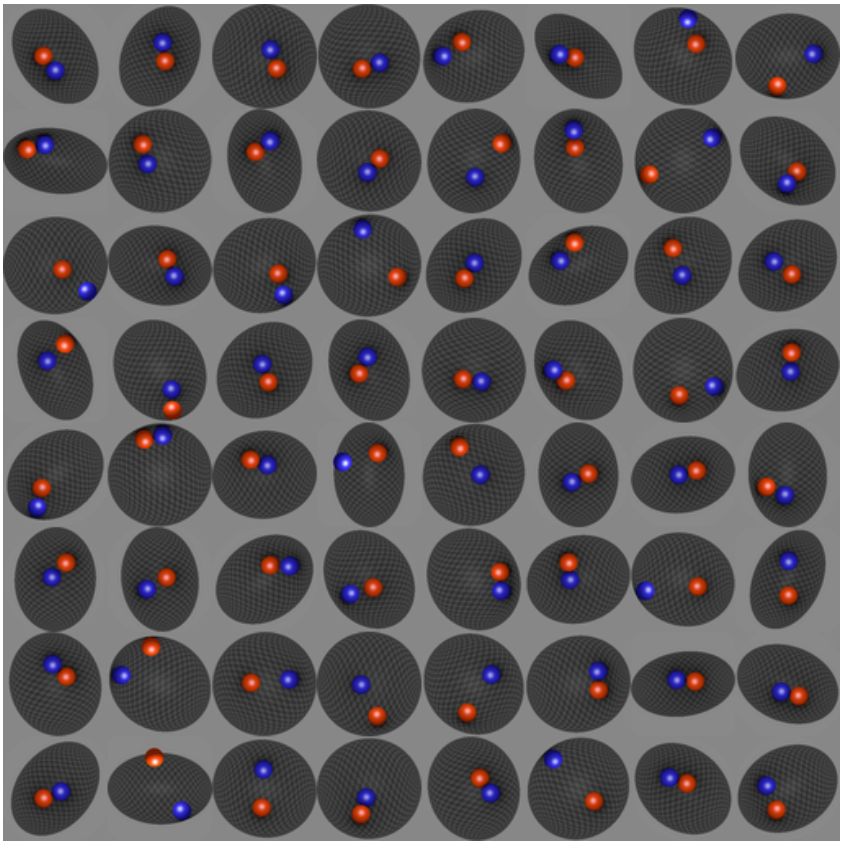


Figure 5.F.1: Sample data from BallsInBowl. The dataset consists of two balls of different colours rolling in elliptical bowls of various shapes.

ShapeStacks. We use the official release of the ShapeStacks dataset[7]. We use the `default` partitioning provided with the dataset and merge the training and validation splits for a total of 264,384 training images. All FID comparisons are made against 10,000 images randomly sampled from the test set which contains 46,560 images in total. Since the original resolution of the images is 224×224 pixels, we re-scale them to 128×128 before feeding them to our network.

RealTraffic. We recorded 5 hours from Youtube[8] of a live traffic camera at a crossing. The video was then unrolled at 10 fps and manually processed to keep only sequences with a number of cars in [1,5]. We kept 560 videos for the training set and 123 in test (80/20 ratio). This dataset will be publicly released.

[7] `https://shapestacks.robots.ox.ac.uk/#data`
[8] `https://www.youtube.com/watch?v=5_XSYlAfJZM`

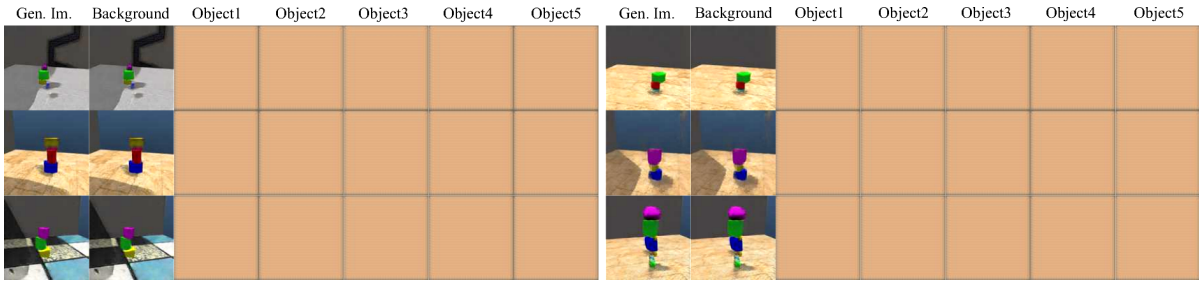
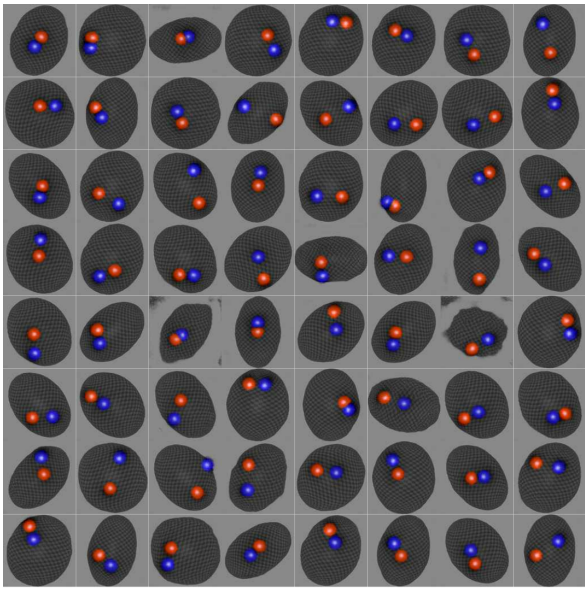


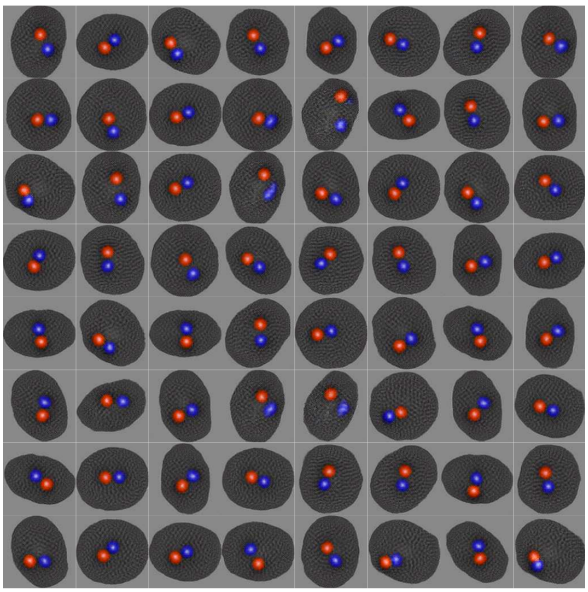
Figure 5.G.1: BlockGAN2D scene decomposition on ShapeStacks. We display in order (generated, background, object_1, ..., object_5) for BlockGAN2D model. We see that in this case BlockGAN doesn't capture objectness at all and render everything in the background. This shows how, for structured scenes, prior work fails to capture correlations between objects.

5.G Qualitative Results

We provide additional qualitative generation results. Figure 5.G.1 shows a failure case of BlockGAN2D mentionned in the paper. In fact, when the scene is more structured BlockGAN2D fails to be object centric and let the background render the entire scene. In addition Figs. 5.G.2 and 5.G.4 to 5.G.7 provide more samples on every dataset for all the models we trained. In particular we can see that when inter-objects relations are weak in CLEVR5 or CLEVR5-vbg, BlockGAN2D performs qualitatively similar to ours (see Figs. 5.G.4 and 5.G.5). However when the scene is more crowded and the objects have higher correlation BlockGAN2D quality decreseases significantly (see Figs. 5.G.2, 5.G.6 and 5.G.7).



RELATE



BLOCKGAN2D

Figure 5.G.2: Generated scenes for models trained on BallsInBowl. Qualitatively RELATE generates images of higher quality compared to BlockGAN2D (Nguyen-Phuoc et al. [150]).

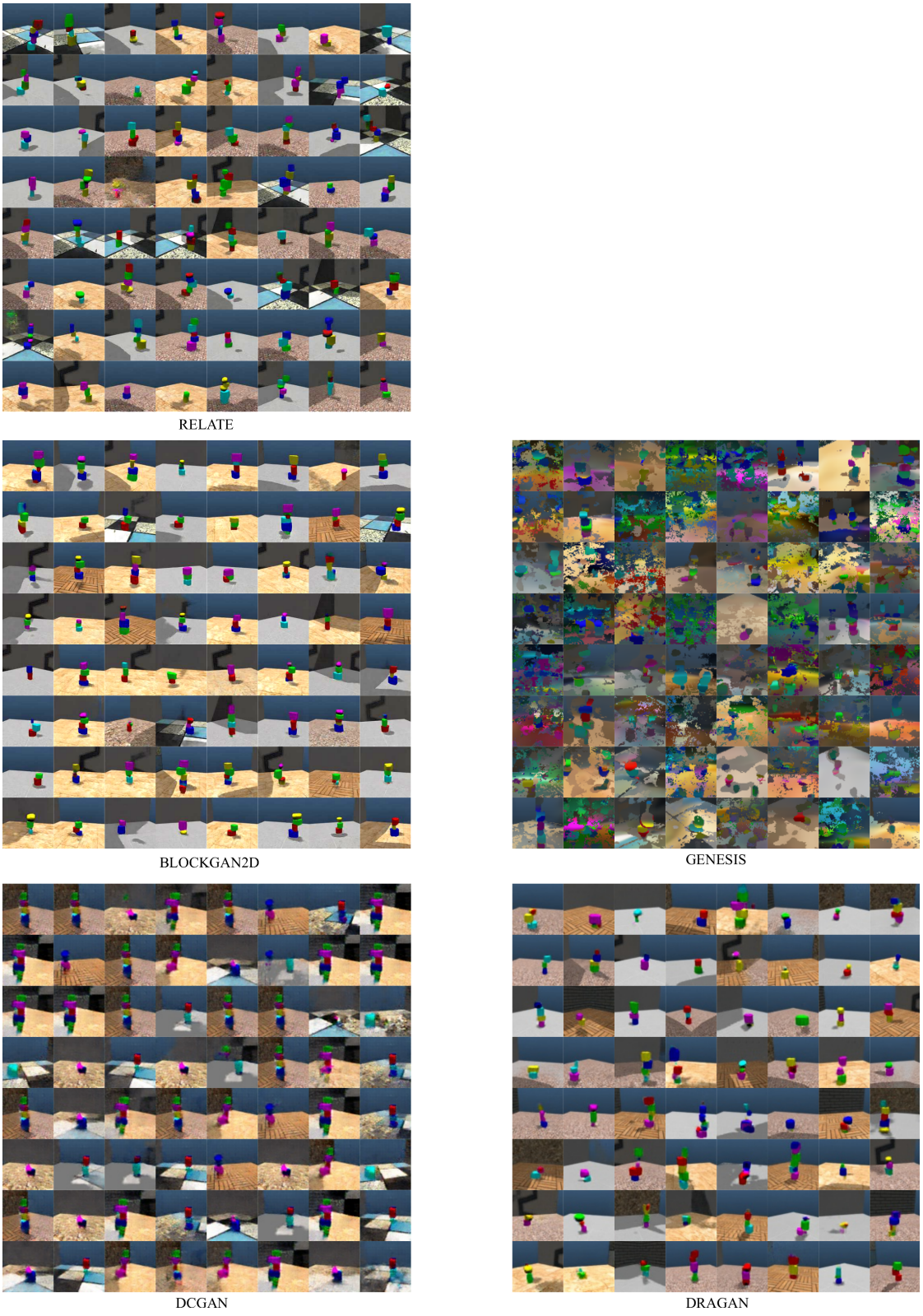


Figure 5.G.3: Generated scenes for models trained on ShapeStacks. Despite qualitative similar rendering, BlockGAN2D is not rendering a scene component-wise as opposed to ours (see Fig. 5.G.1).

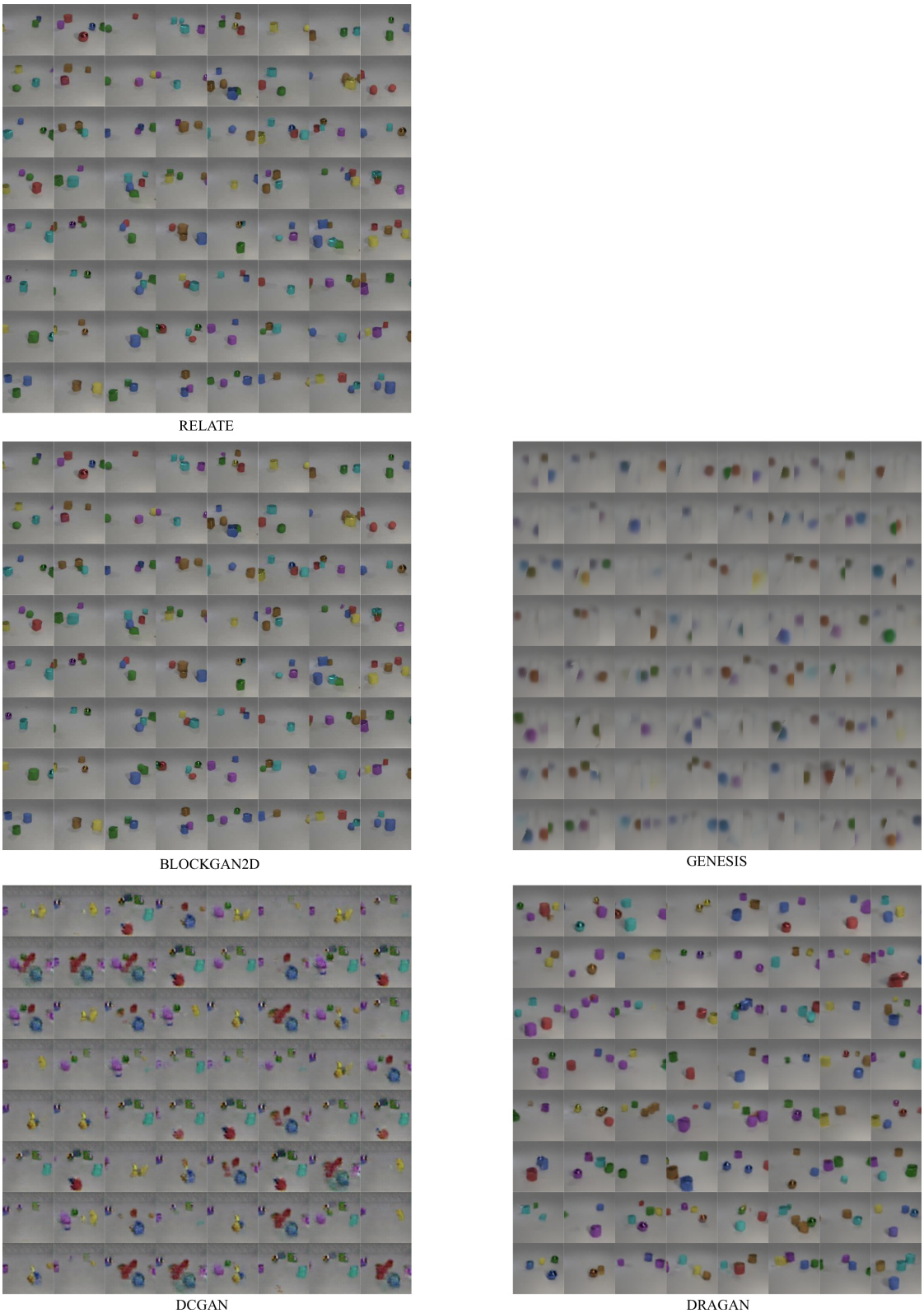


Figure 5.G.4: Generated scenes for models trained on CLEVR5. For less crowded scenes our model and BlockGAN2D reach similar performances.

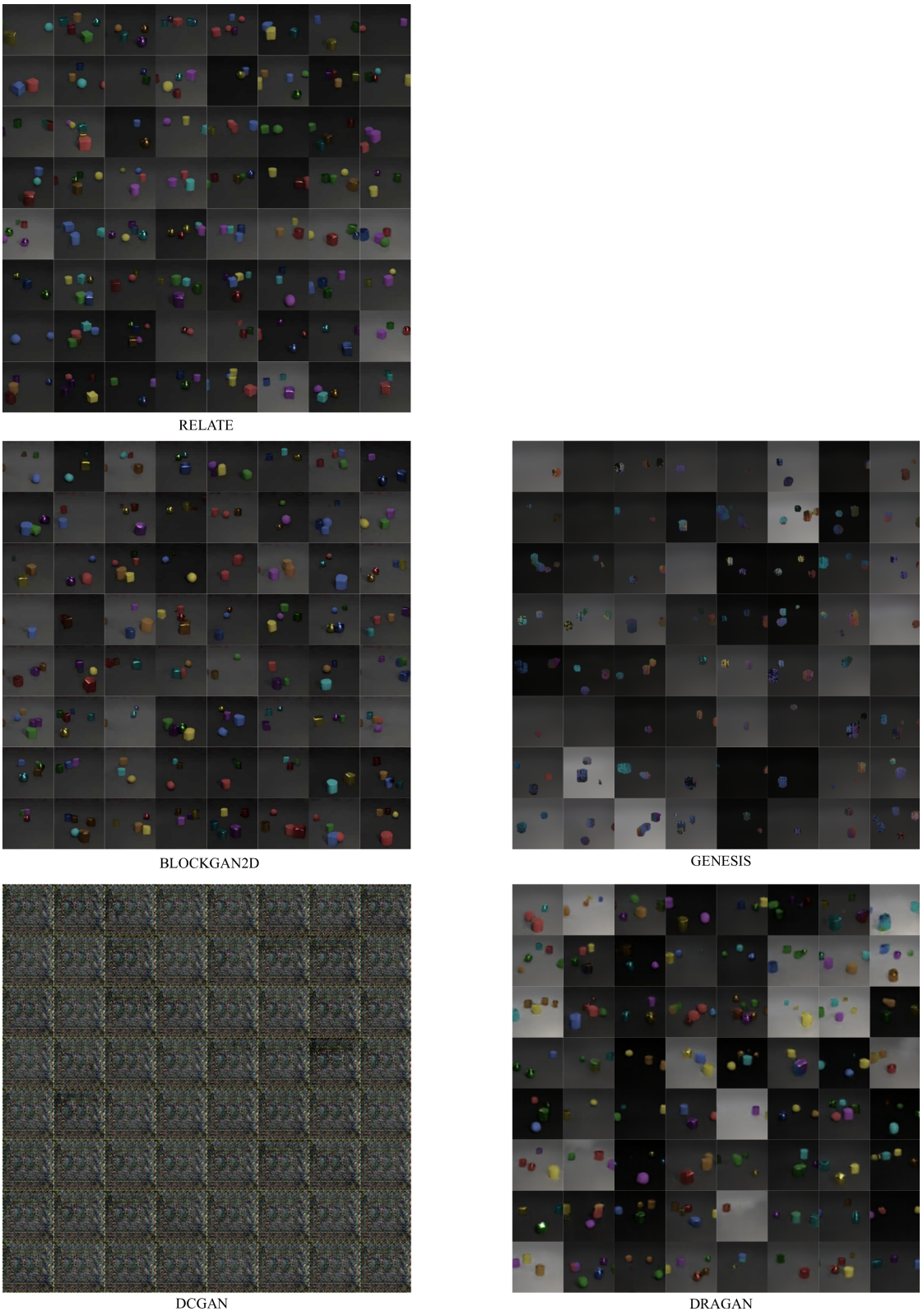


Figure 5.G.5: Generated scenes for models trained on CLEVR5-vbg. This scenario reaches similar conclusion as Fig. 5.G.4.

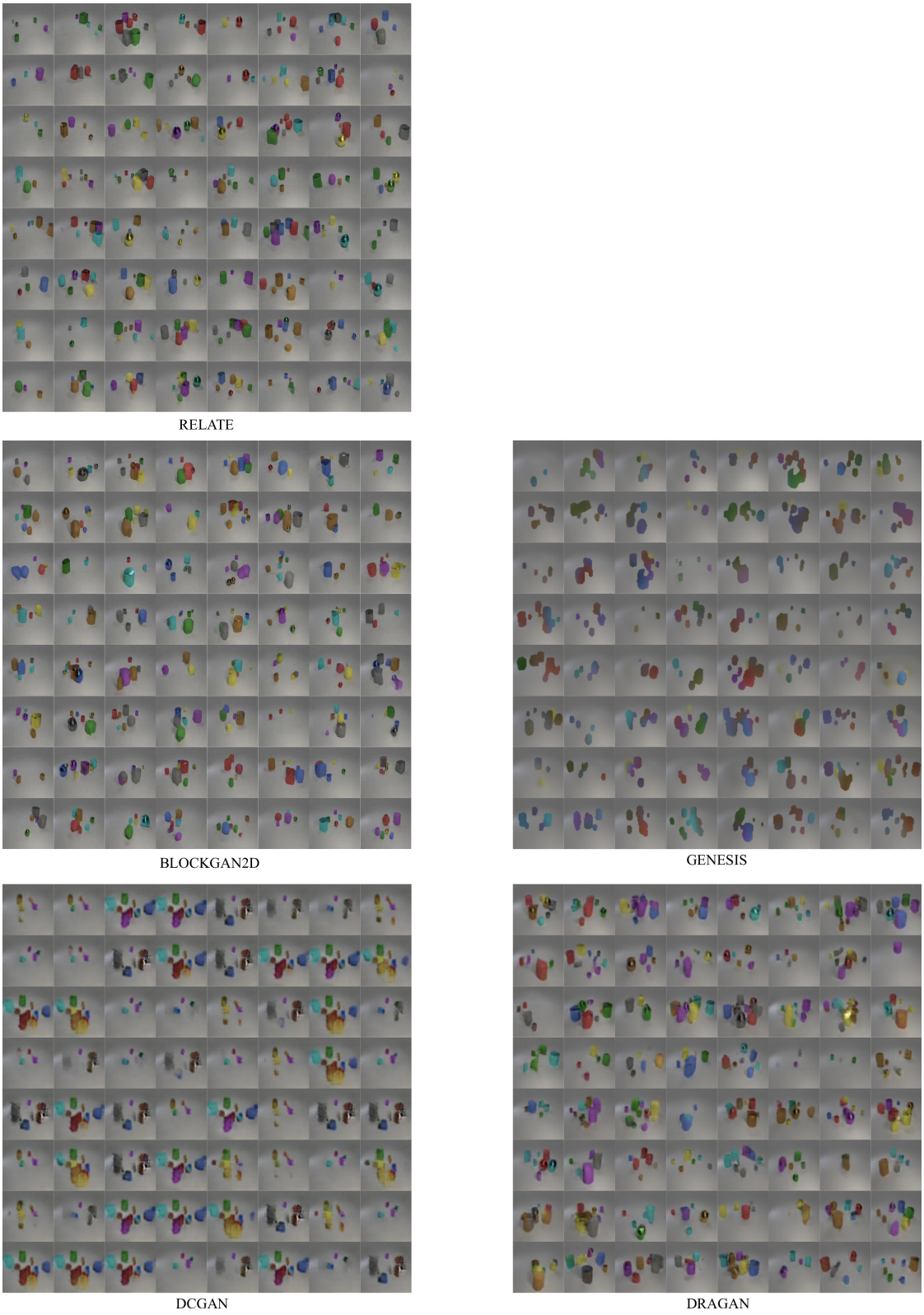


Figure 5.G.6: Generated scenes for models trained on CLEVR. When the scene gets more crowded RELATE gets an advantage as it can push objects apart resulting in higher qualitative rendering.

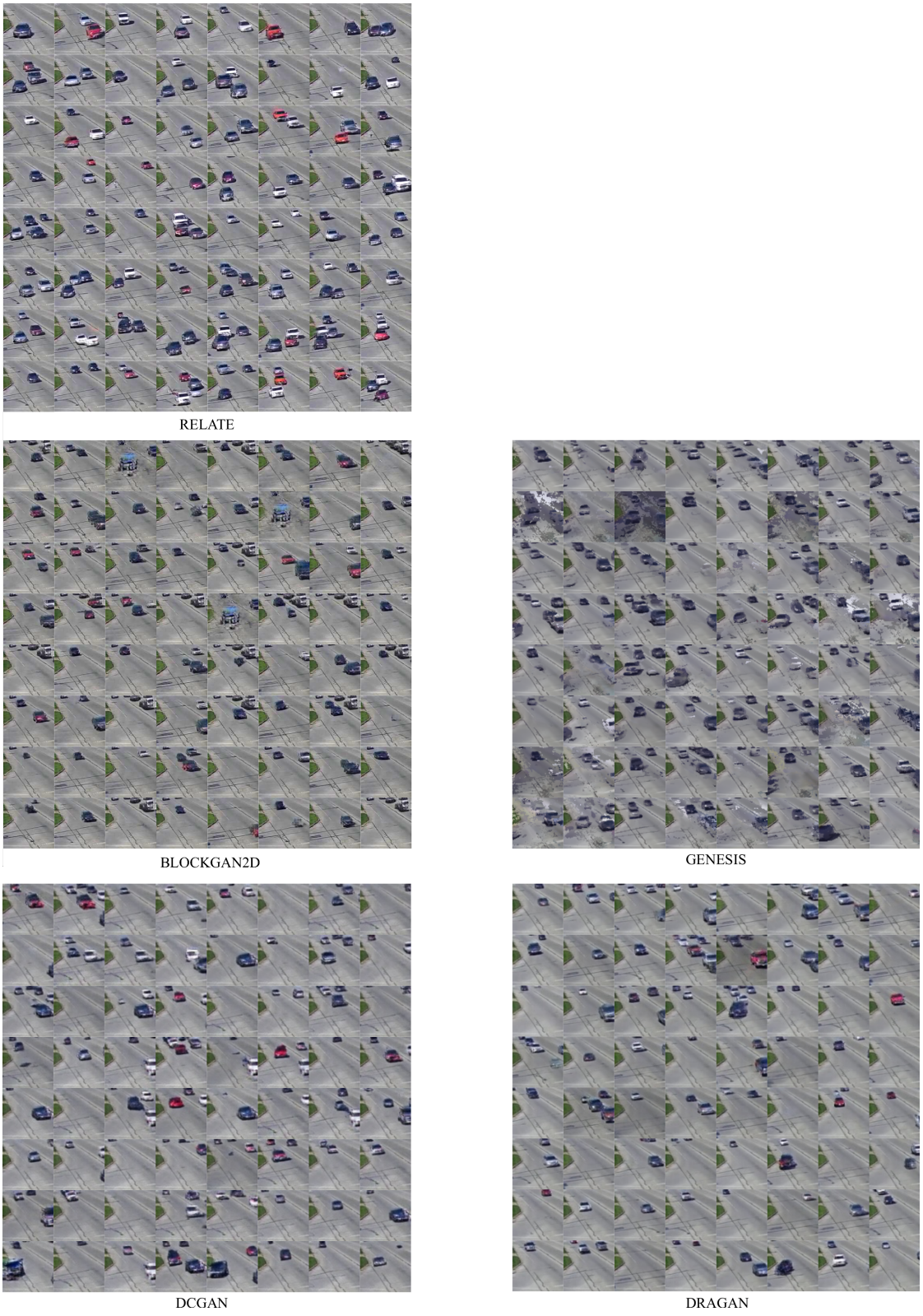


Figure 5.G.7: Generated scenes for models trained on RealTraffic. Our model qualitatively renders higher fidelity images. BlockGAN2D sometimes suffers from background mode collapse (see first, second and fourth rows of second block).

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	
Publication Status	**x** Published □ Accepted for Publication □ Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Sébastien Ehrhardt*, Oliver Groth*, Aron Monszpart, Martin Engelcke, Ingmar Posner, Niloy Mitra, Andrea Vedaldi "RELATE: Physically Plausible Multi-Object Scene Synthesis Using Structured Latent Spaces". In: Advances in Neural Information Processing Systems (NeurIPS). Dec. 2020

Student Confirmation

Student Name:	Oliver Groth		
Contribution to the Paper	- performed related work survey - conceived the idea of integrating neural dynamics approximation with an object-centric representation - designed ablation experiments on the influence of the relationship module Gamma - trained image generation baseline models - implemented video generation baseline and FVD evaluation of video fidelity - performed object disentanglement analysis - contributed to the paper writing, specifically: Related Work, Experiments, Broader Impact Statement		
Signature	Oliver Groth	Date	24/09/2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	PROFESSOR ANDREA VEDALDI		
Supervisor comments	This is an accurate description of the contributions.		
Signature	Andrea Vedaldi	Date	24/9/2021

This completed form should be included in the thesis, at the end of the relevant chapter.

6

Goal-Conditioned End-to-End Visuomotor Control for Versatile Skill Primitives

In this chapter we propose an efficient network architecture for end-to-end, goal-conditioned visuomotor control. We extend prior art in this area by devising a controller which can infer a new manipulation task from a given target image without the need of additional demonstrations or fine-tuning as otherwise common in imitation or meta learning. We demonstrate the efficacy, robustness and versatility of our controller architecture benchmarking its performance on complex, simulated pushing and pick-and-place tasks. We observe strong performance gains over comparable imitation learning and model-predictive control approaches. Additionally, we show that our model transfers to novel tasks immediately without domain-randomisation during training or fine-tuning during execution while simultaneously being robust to challenging visual distractions. This work is published as:

O. Groth, C.-M. Hung, A. Vedaldi, and I. Posner. "Goal-Conditioned End-to-End Visuomotor Control for Versatile Skill Primitives". In: *IEEE International Conference on Robotics and Automation (ICRA)*. June 2021

Abstract

Visuomotor control (VMC) is an effective means of achieving basic manipulation tasks such as pushing or pick-and-place from raw images. Conditioning VMC on desired goal states is a promising way of achieving versatile *skill primitives*. However, common conditioning schemes either rely on task-specific fine tuning - e. g. using one-shot imitation learning (IL) - or on sampling approaches using a forward model of scene dynamics i. e. model-predictive control (MPC), leaving deployability and planning horizon severely limited. In this paper we propose a conditioning scheme which avoids these pitfalls by learning the controller and its conditioning in an end-to-end manner. Our model predicts complex action sequences based directly on a dynamic image representation of the robot motion and the distance to a given target observation. In contrast to related works, this enables our approach to efficiently perform complex manipulation tasks from raw image observations without predefined control primitives or test time demonstrations. We report significant improvements in task success over representative MPC and IL baselines. We also demonstrate our model's generalisation capabilities in challenging, unseen tasks featuring visual noise, cluttered scenes and unseen object geometries.

6.1 Introduction

With recent advances in deep learning, we can now learn robotic controllers end-to-end, mapping directly from raw video streams into a robot's command space. The promise of these approaches is to build real-time visuomotor controllers without the need for complex pipelines or predefined macro-actions (e. g. for grasping). End-to-end visuomotor controllers have demonstrated remarkable performance in real systems, e. g. learning to pick up a cube and place it in a basket (S. James et al. [88] and Zhu et al. [233]). However, a common drawback of current visuomotor controllers is their limited *versatility* due to an often very narrow task definition. For example, in the controllers of (S. James et al. [88] and Zhu et al. [233]), which are unconditioned, putting a red cube into a blue basket is a different task than putting

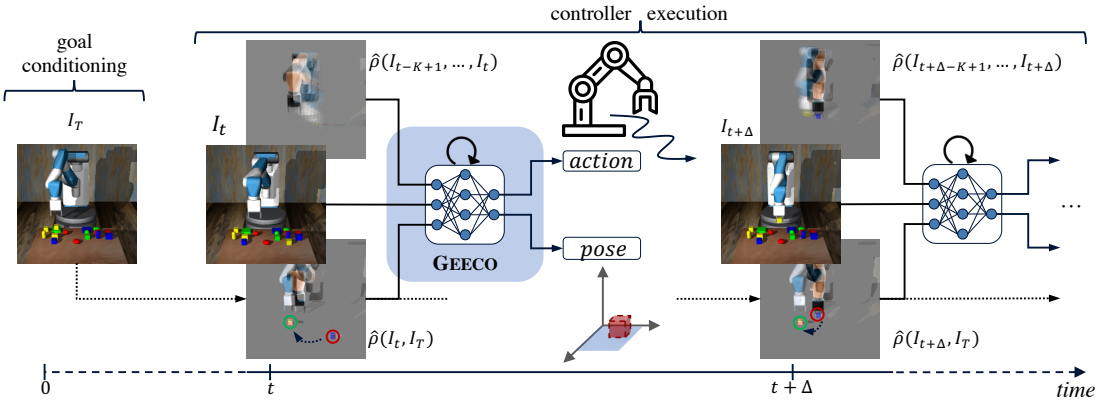


Figure 6.1.1: Our proposed model executes a task given by a target image I_T. In this example, I_T indicates that the small yellow cube from the front right needs to be moved onto the green pad in the back left. Dynamic images are used to (1) represent the difference between the current observation and the target $\hat{\rho}(I_t, I_T)$ and (2) to capture the motion dynamics $\hat{\rho}(I_{t-K+1}, \ldots, I_t)$. Current and target location of the manipulated object are highlighted by red and green circles respectively.

a yellow cube into a green basket. In contrast to that, in this paper we consider a broader definition of task and argue that it should rather be treated as a *skill primitive* (e. g. a policy which can pick up any object and place it anywhere else). Such a policy must thus be conditioned on certain arguments, e. g. specifying the object to be moved and its target.

Several schemes have been proposed to condition visuomotor controllers on a *target image*, e. g. an image depicting how a scene should look like after the robot has executed its task. Established conditioning schemes build on various approaches such as model-predictive control (Ebert et al. [35]), task-embedding (S. James et al. [86]) or meta-learning (Finn et al. [47]) and are discussed in greater detail in Section 6.2. However, the different methods rely on costly sampling techniques, access to prior demonstrations or task-specific fine-tuning during test time restraining their general applicability.

In contrast to prior work, we propose an efficient end-to-end controller which can be conditioned on a single target image without fine-tuning and regresses directly to motor commands of an actuator without any predefined macro-actions. This allows us to learn general *skill primitives*, e. g. pushing and pick-and-place skills,

which are versatile enough to immediately generalise to new tasks, i. e. unseen scene setups and objects to handle. Our model utilises *dynamic images* (Bilen et al. [18]) as a succinct representation of the video dynamics in its observation buffer as well as a visual estimation of the difference between its current observation and the target it is supposed to accomplish. Figure 6.1.1 depicts an example execution of our visuomotor controller, its conditioning scheme and intermediate observation representations.

In summary, our contributions are three-fold: Firstly, we propose a novel architecture for visuomotor control which can be efficiently conditioned on a new task with just one single target image. Secondly, we demonstrate our model's efficacy by outperforming representative MPC and IL baselines in pushing and pick-and-place tasks by significant margins. Lastly, we analyse the impact of the dynamic image representation in visuomotor control providing beneficial perception invariances to facilitate controller resilience and generalisation without the need of sophisticated domain randomisation schemes during training.

6.2 Related Work

The problem of *goal conditioning* constitutes a key challenge in visuomotor control: Given a specific task specification (e. g. putting a red cube onto a blue pad), it needs to be communicated to the robot, which in turn must adapt its control policy in such a way that it can carry out the task. In this paper, we focus on goals which are communicated visually, i. e. through images depicting how objects should be placed on a table. Prior methods which have shown impressive real-world results typically involve dedicated sub-modules for perception and planning (Labbé et al. [114]) or are only loosely goal-conditioned, e. g. on a target shape category (Pashevich et al. [156]). We restrict our survey to end-to-end controllers which can be conditioned on a *single target image* and group related work by their condition schemes and action optimisation methods.

In *visual model-predictive control* one learns a forward model of the world, forecasting the outcome of an action. The learned dynamics model is then explored via sampling or gradient-based methods to compute a sequence of actions which brings the predicted observation closest to a desired goal observation. An established line of work on *Deep Visual Foresight* (VFS) (Ebert et al. [35], Finn and Levine [46], S. Nair and Finn [147], and Xie et al. [217]) learns action-conditioned video predictors and employs CEM-like (Rubinstein and Kroese [166]) sampling methods for trajectory optimisation, successfully applying those models to simulated and real robotic pushing tasks. Instead of low-level video prediction, visual MPC can also be instantiated using higher-level, object-centric models for tasks such as block stacking (Janner et al. [90] and Y. Ye et al. [222]). Another line of work attempts to learn forward dynamics models in suitable latent spaces. After projecting an observation and a goal image into the latent space, a feasible action sequence can then be computed using gradient-based optimisation methods (Byravan et al. [26], Srinivas et al. [183], Watter et al. [203], and Yu et al. [229]). Even though MPC approaches have shown promising results in robot manipulation tasks, they are limited by the quality of the forward model and do not scale well due to the action sampling or gradient optimisation procedures required. In contrast to them our model regresses directly to the next command given a buffer of previous observations.

One-Shot Imitation Learning seeks to learn general task representations which are quickly adaptable to unseen setups. MIL (Finn et al. [47]) is a meta-controller, which requires fine-tuning during test time on one example demonstration of the new task to adapt to it. In contrast to MIL, TecNet (S. James et al. [86]) learns a task embedding from expert demonstrations and requires at least one demonstration of the new task during test time to modulate its policy according to similar task embeddings seen during training. Additionally, a parallel line of work in that domain operates on discrete action spaces (D.-A. Huang et al. [83] and D. Xu et al. [218]) and maps demonstrations of new tasks to known macro actions. Unlike those methods, our model is conditioned on a single target image and does not require any fine-tuning on a new task during test time.

Goal-conditioned reinforcement learning (Kaelbling [93]) is another established paradigm for learning of control policies. However, due to the unwieldy nature of images as state observations, the use of goal images typically comes with limiting assumptions such as being from a previously observed set (Warde-Farley et al. [202]) or living on the manifold of a learned latent space (A. V. Nair et al. [146]). Our proposed utilisation of dynamic images for goal conditioning circumvents such limitations and can be seen as complementary to other work which incorporates demonstrations into goal-conditioned policy learning (Ding et al. [32] and A. Nair et al. [145]) by enabling efficient bootstrapping of a control policy on goal-conditioned demonstrations.

6.3 Goal-Conditioned Visuomotor Control

In order to build a visuomotor controller which can be efficiently conditioned on a target image and is versatile enough to generalise its learned policy to new tasks immediately, we need to address the following problems: Firstly, we need an efficient way to detect scene changes, i. e. answering the question 'Which object has been moved and where from and to?' Secondly, we want to filter the raw visual observation stream such that we only retain information pertinent to the control task; specifically the motion dynamics of the robot. Drawing inspiration from previous work in VMC and action recognition, we propose *GEECO*, a novel architecture for *goal-conditioned end-to-end control* which combines the idea of *dynamic images* (Bilen et al. [18]) with a robust end-to-end controller network (S. James et al. [88]) to learn versatile manipulation skill primitives which can be conditioned on new tasks on the fly. We discuss next the individual components.

Dynamic images. In the domain of action recognition, dynamic images have been developed as a succinct video representation capturing the dynamics of an entire frame sequence in a single image. This enables the treatment of a video with convolutional neural networks as if it was an ordinary RGB image facilitating dynamics-related feature extraction. The core of the dynamic image representation is a ranking machine which learns to sort the frames of a video temporally (Fernando

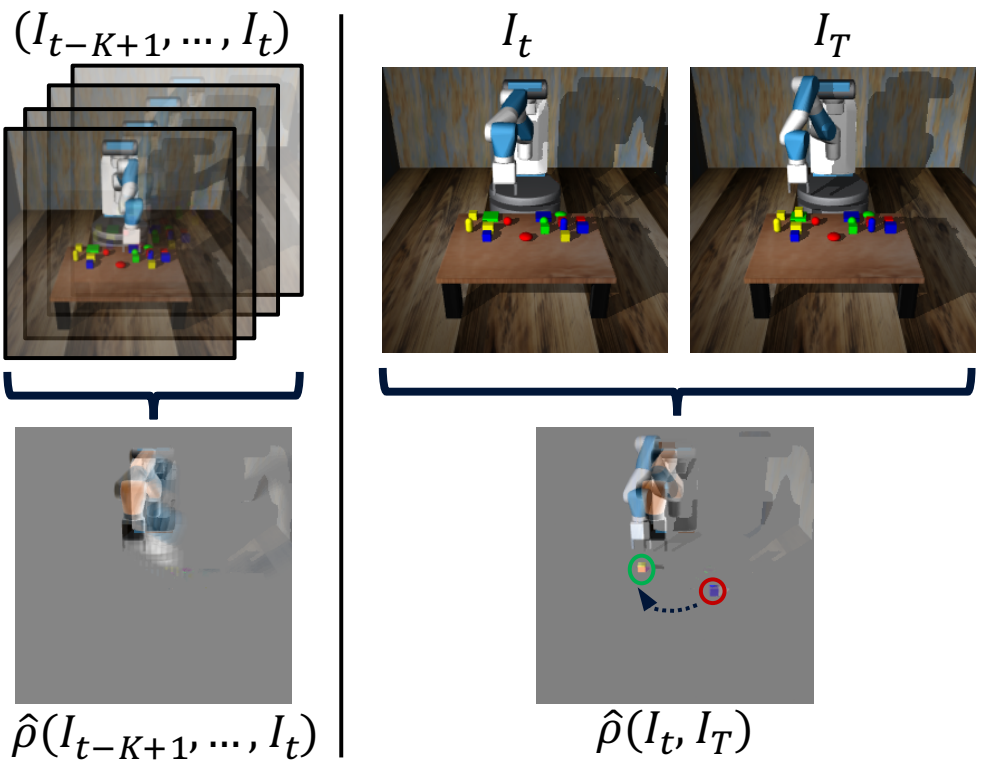


Figure 6.3.1: Utilisation of dynamic images. Left: A dynamic image represents the motion occurring in a sequence of K consecutive RGB observations. Right: A dynamic image represents the changes which occurred between the two images I_t and I_T like the change of object positions as indicated by the red and green circles.

et al. [43]). As shown by prior work (Bilen et al. [18]), an approximate linear ranking operator $\hat{\rho}(\cdot)$ can be applied to any sequence of H temporally ordered frames $(I_1, \dots, I_H)$ and any image feature extraction function $\psi(\cdot)$ to obtain a dynamic feature map according to the following Eq. (6.1):

$$\hat{\rho}(I_1, \dots, I_H; \psi) = \sum_{t=1}^{H} \alpha_t \psi(I_t) \tag{6.1}$$

$$\alpha_t = 2(H - t + 1) - (H + 1)(\mathcal{H}_H - \mathcal{H}_{t-1}) \tag{6.2}$$

Here, $\mathcal{H}_t = \sum_{i=1}^{t} 1/t$ is the t-th Harmonic number and $\mathcal{H}_0 = 0$. Setting $\psi(\cdot)$ to the identity, $\hat{\rho}(I_1, \dots, I_H)$ yields a dynamic image which, after normalisation across all channels, can be treated as a normal RGB image by a downstream network. The employment of dynamic images serves two important purposes in our network, as depicted in Fig. 6.3.1. Firstly, it compresses a window of the last K RGB observations into one image $\hat{\rho}(I_{t-K+1}, \dots, I_t)$ capturing the current motion of the robot arm.

Secondly, given a target image I_T depicting the final state the scene should be in, the dynamic image $\hat{\rho}(I_t, I_T)$ lends itself very naturally to represent the *visual difference* between the current observation I_t and the target state I_T. Another advantage of using dynamic images in these two places is to make the controller network invariant w.r.t. the static scene background and, approximately, the object colour, allowing it to focus on location and geometry of objects involved in the manipulation task.

Observation buffer. During execution, our network maintains a buffer of most recent K observations as a sequence of pairs $((I_{t-K+1}, \mathbf{x}_{t-K+1}), \ldots, (I_t, \mathbf{x}_t))$ where I_t is the RGB frame at time step t and $\mathbf{x}_t$ is the proprioceptive feature of the robot at the same time step represented as a vector of its joint angles. Throughout our experiments we set $K = 4$. The observation buffer breaks long-horizon manipulation trajectories into shorter windows which retain relative independence from each other. This endows the controller with a certain error-correction capacity, e. g. when a grasped object slips from the gripper prematurely, the network can regress back to a pick-up phase.

Goal conditioning. Before executing a trajectory, our controller is *conditioned* on the task to execute, i. e. moving an object from its initial position to a goal position, via a target image I_T depicting the scene after the task has been carried out. As shown in Fig. 6.3.1 (right), the dynamic image representation $\hat{\rho}(I_t, I_T)$ helps this inference process by only retaining the two object positions and the difference in the robot pose while cancelling out all static parts of the scene.

Network architecture. The controller network takes the current observation buffer $((I_{t-K+1}, \mathbf{x}_{t-K+1}), \ldots, (I_t, \mathbf{x}_t))$ and the target image I_T as input and regresses to the following two action outputs: (1) The change in Cartesian coordinates of the end effector $\hat{\mathbf{u}}_{\Delta EE}$ and (2) a discrete signal $\hat{\mathbf{u}}_{GRP}$ for the gripper to either open (-1), close $(+1)$ or stay in position (0). Additionally, the controller regresses two auxiliary outputs: the current position of the end effector $\hat{\mathbf{q}}_{EE}$ and of the object to manipulate $\hat{\mathbf{q}}_{OBJ}$, both in absolute Cartesian world coordinates. While the action vectors $\hat{\mathbf{u}}_{\Delta EE}$ and $\hat{\mathbf{u}}_{GRP}$ are directly used to control the robot, the position predictions serve as an auxiliary signal during the supervised training process to encourage the network to

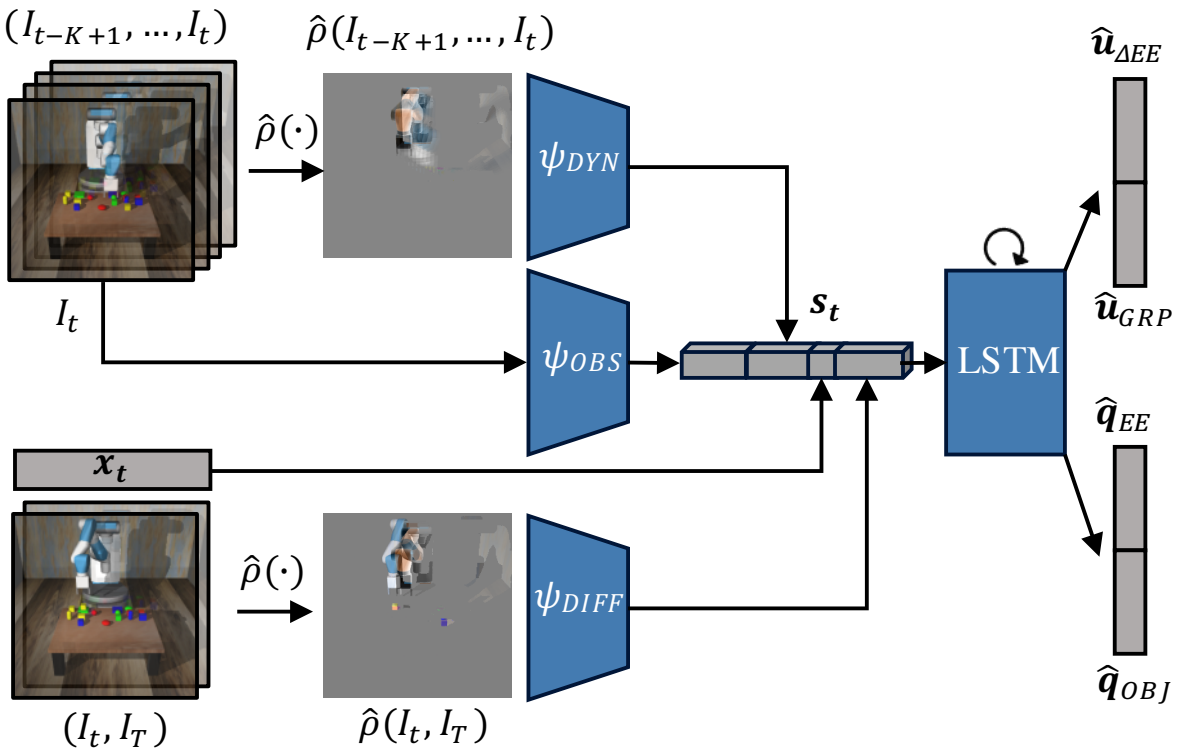


Figure 6.3.2: Network architecture of GEECO-$\mathcal{F}$. The observation buffer $(I_{t-K+1}, \ldots, I_T)$ is compressed into a dynamic image via $\hat{\rho}(.)$ and passed through ψ_{DYN}. The current difference to the target frame I_T is also computed via $\hat{\rho}(.)$ and passed through ψ_{DIFF}. Lastly, the current observation I_t is encoded via ψ_{OBS}. All CNNs compute spatial feature maps which are concatenated to the tiled proprioceptive feature $\mathbf{x}_t$. The LSTM's output is decoded into command actions $\hat{\mathbf{u}}_{\Delta EE}$ and $\hat{\mathbf{u}}_{GRP}$ as well as auxiliary pose predictions $\hat{\mathbf{q}}_{EE}$ and $\hat{\mathbf{q}}_{OBJ}$.

learn intermediate representations correlated to the world coordinates. A sketch of our model architecture can be found in Fig. 6.3.2 and a detailed description of all architectural parameters can be found in Section 6.A.

The full model is trained in an end-to-end fashion on N expert demonstrations of manipulation tasks collected in a simulation environment. Each expert demonstration is a sequence of H time steps indexed by t containing: the RGB frame I_t, the proprioceptive feature $\mathbf{x}_t$, the robot commands $\mathbf{u}^*_{\Delta EE}(t)$, $\mathbf{u}^*_{GRP}(t)$ and the positions of the end effector and the object to manipulate $\mathbf{q}^*_{EE}(t)$, $\mathbf{q}^*_{GRP}(t)$. During training, we

minimise the following loss function:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{N} \Bigg[\sum_{t=1}^{H-K+1} & \mathrm{MSE}(\hat{\mathbf{u}}_{\Delta EE}(\mathbf{fi}_{i,t}), \mathbf{u}^*_{\Delta EE}(i,t)) \\ & + \mathrm{CCE}(\hat{\mathbf{u}}_{GRP}(\mathbf{fi}_{i,t}), \mathbf{u}^*_{GRP}(i,t)) \\ & + \lambda \Big(\mathrm{MSE}(\hat{\mathbf{q}}_{EE}(\mathbf{fi}_{i,t}), \mathbf{q}^*_{EE}(i,t)) \\ & + \mathrm{MSE}(\hat{\mathbf{q}}_{OBJ}(\mathbf{fi}_{i,t}), \mathbf{q}^*_{OBJ}(i,t)) \Big) \Bigg]\end{aligned} \tag{6.3}$$

In Eq. (6.3) MSE and CCE are abbreviations of *Mean-Squared Error* and *Categorical Cross-Entropy* respectively. The hyperparameter λ weighs the auxiliary loss terms for pose prediction. The shorthand notation $\mathbf{fi}_{i,t}$ represents the t-th training window in the i-th expert demonstration of the training dataset comprising of $((I_t, \mathbf{x}_t), \ldots, (I_{t+K-1}, \mathbf{x}_{t+K-1});\ I_T = I_H)$; $\mathbf{u}^*(i,t)$ and $\mathbf{q}^*(i,t)$ are the corresponding ground truth commands, and $\hat{\mathbf{u}}(\mathbf{fi}_{i,t})$ and $\hat{\mathbf{q}}(\mathbf{fi}_{i,t})$ are shorthand notations for the network predictions on that window. During training we always set the target frame I_T to be the last frame of the expert demonstration I_H.

Model ablations. We refer to our *full* model as GEECO-$\mathcal{F}$ (or just $\mathcal{F}$ for short) as depicted in Fig. 6.3.2. However, in order to gauge the effectiveness of our different architecture design decisions, we also consider two ablations of our full model which are briefly described below. We refer the reader to Section 6.B for more comprehensive details and architecture sketches.

GEECO-$\mathcal{R}$: This ablation has ψ_{DYN} and ψ_{DIFF} removed and ψ_{OBS} is responsible for encoding the current observation I_t and the target image I_T and the feature distance is used for goal conditioning. Thus, the state tensor becomes $\mathbf{s}_t = \psi_{OBS}(I_t) \oplus \mathbf{x}_t \oplus (\psi_{OBS}(I_T) - \psi_{OBS}(I_t))$, where $\oplus$ denotes concatenation along the channel dimension. This *residual* state encoding serves as a baseline for learning meaningful goal distances in the feature space induced by ψ_{OBS}.

GEECO-$\mathcal{D}$: This ablation has only the 'motion branch' ψ_{DYN} removed. The state tensor is comprised of $\mathbf{s}_t = \psi_{OBS}(I_t) \oplus \mathbf{x}_t \oplus \psi_{DIFF}(\hat{\rho}(I_t, I_T))$. This gauges the effectiveness of

using an explicitly shaped goal difference function over an implicitly learned one like in GEECO-$\mathcal{R}$.

6.4 Experiments

Our experimental design is guided by the following questions: (1) How do the learned skill primitives compare to representative MPC and IL approaches? (2) Can our controller deliver on its aspired *versatility* by transferring its skills, acquired only on simple cubes, to novel shapes and adverse conditions?

Experimental setup and data collection. We have designed a simulation environment, Goal2Cube2, containing four different tasks to train and evaluate our controller on which are presented in Fig. 6.4.1. In each task one of the small cubes needs to be moved onto one of the larger target pads. The scenario is designed such that the task is ambiguous and the controller needs to infer the object to manipulate and the target location from a given target image depicting the task to perform. We use the MuJoCo physics engine (Todorov et al. [194]) for simulation. We adapt the Gym environment (Brockman et al. [21]) provided by (Duan et al. [34]) featuring a model of a *Fetch Mobile Manipulator* (Wise et al. [206]) with a 7-DoF arm and a 2-point gripper[1]. For each skill, i. e. pushing and pick-and-place, we collect 4,000 unique expert demonstrations of the robot completing the task successfully in simulation according to a pre-computed plan. Each demonstration is four seconds long and is recorded as a list of observation-action tuples at 25 Hz resulting in an episode length of $H = 100$.

Baselines for goal-conditioned VMC. Our first baseline, VFS (Ebert et al. [35]), is a visual MPC which runs on a video prediction backbone. A CEM-based (Rubinstein and Kroese [166]) action sampler proposes command sequences over a short time horizon which are evaluated by the video predictor. The sequence which results in a predicted observation closest to the goal image is executed and the process is repeated until termination. We use SAVP (Lee et al. [116]) as action-conditioned video

[1]Despite the robot having a mobile platform, its base is fixed during all experiments.

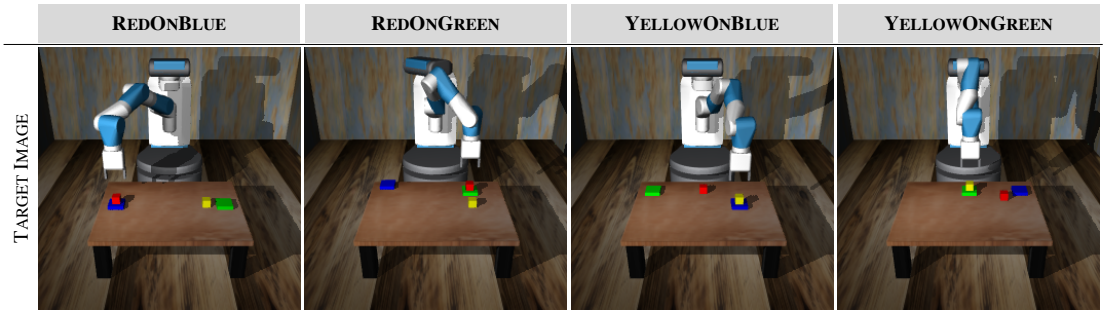


Figure 6.4.1: Basic manipulation tasks in Goal2Cube2. In each task, one of the small cubes (red or yellow) needs to be moved onto one of the target pads (blue or green). The tasks can be accomplished via a pushing or pick-and-place manipulation (target pads are reduced to flat textures for pushing).

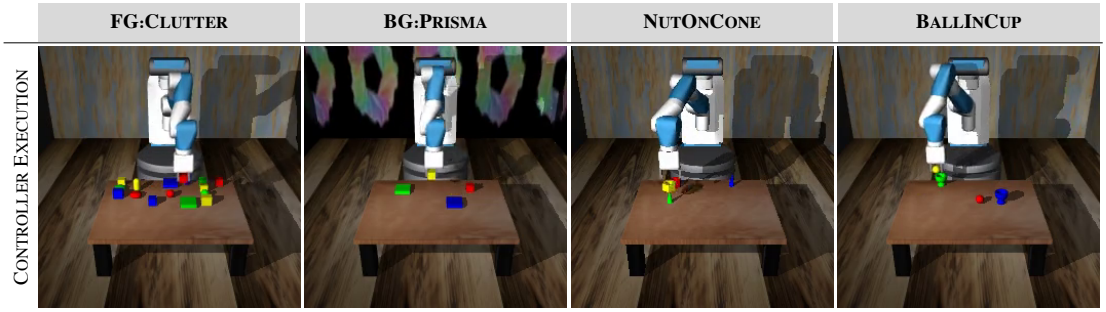


Figure 6.4.2: Generalisation experiments. FG:Clutter and BG:Prisma are variations of Goal2Cube2 with severe physical and visual distractions. The background in BG:Prisma is a video looping rainbow colour patterns. NutOnCone and BallInCup require the application of the pick-and-place skill to novel shapes and target configurations.

predictor and train it on our datasets with the hyperparameters reported for the BAIR robot-pushing dataset (Ebert et al. [36]). Since the time horizon of our scenarios is much longer than SAVP's prediction horizon, we estimate an upper-bound of the performance of VFS by providing intermediate goal images.[2] Our second baseline, TecNet (S. James et al. [86]), is an IL model which is capable of quick adaptation given the demonstration of a new task. The demonstration can be as sparse as start and end image of an executed trajectory making it applicable to our setup. We employ TecNet in its one-shot imitation configuration. We refer to Sections 6.D and 6.E for a comprehensive overview over the hyperparameters used for all baselines.

Training and evaluation protocol. For each skill, we split the demonstrations into

[2]HVF (S. Nair and Finn [147]) employs a similar 'ground truth bottleneck' scheme to upper-bound a visual MPC baseline.

training, validation and test sets with a ratio of 2 : 1 : 1 respectively while keeping the task distributions balanced. We train all models for 300k gradient update steps using the Adam optimiser (Kingma and Ba [103]). After each epoch, we evaluate the controller performance on the validation set and select the best performing model checkpoints for the final evaluation on the test set.

During task execution, we monitor the following performance metrics: REACH (robot touches object to manipulate at least once) PICK (robots palm touches object to manipulate while fingers are closed) PUSH / PLACE (correct object sits on the target pad by the end of the episode). Each evaluation episode is terminated after 200 timesteps (8 seconds).

Basic manipulation results. We start our investigation by comparing the performance of our proposed model and its ablations to VFS and TECNET in the GOAL2CUBE2 scenario and report the results in Table 6.4.1. We treat the tasks (REACH, PUSH, PICK, PLACE) as Bernoulli experiments and report their mean success rates for 1,000 trials as well as their binomial proportion confidence intervals at 0.95. For pushing tasks we measure nearly perfect reaching and very strong pushing performance for GEECO-$\{\mathcal{R}, \mathcal{D}, \mathcal{F}\}$ outperforming the baselines by up to 80% on final task success. We also observe that $\mathcal{D}$ and $\mathcal{F}$, models which employ the dynamic image representation, perform significantly better than the RGB-only ablation $\mathcal{R}$. Likewise, the general ranking of models also applies to pick-and-place tasks with $\mathcal{D}$ outperforming the baselines by up to 60% task success. The almost complete failure of VFS and TECNET for a multi-stage task like pick-and-place is unsurprising given that both models have been originally designed for reaching and pushing tasks.

Qualitatively, we observe that VFS reaches the correct object relatively reliably due to its powerful video predictor but struggles to maintain a firm grasp on an object due to its stochastic action sampling. TECNET fares better in that regard since it is a feedforward regression network like GEECO and can maintain a stable grasp. However, it often approaches the wrong object to manipulate due to the fact that its policy is modulated by the embedding of similar tasks. When confronted with

	Pushing		Pick-and-Place		
	reach [%]	push [%]	reach [%]	pick [%]	place [%]
VFS[a]	66.00 ± 9.28	7.00 ± 5.00	87.00 ± 6.59	40.00 ± 9.60	0.00 ± 0.00
TecNet	54.10 ± 3.09	15.70 ± 2.25	32.00 ± 2.89	15.70 ± 2.25	0.80 ± 0.55
$\mathcal{R}, \lambda = 0.0$	98.70 ± 0.70	79.80 ± 2.49	86.20 ± 2.14	58.90 ± 3.05	42.50 ± 3.06
$\mathcal{D}, \lambda = 0.0$	98.70 ± 0.70	**88.50 ± 1.98**	**95.40 ± 1.30**	65.80 ± 2.94	54.20 ± 3.09
$\mathcal{F}, \lambda = 0.0$	98.00 ± 0.87	78.60 ± 2.54	92.70 ± 1.61	71.00 ± 2.81	45.60 ± 3.09
$\mathcal{R}, \lambda = 1.0$	**98.90 ± 0.65**	79.80 ± 2.49	84.90 ± 2.22	50.40 ± 3.10	33.70 ± 2.93
$\mathcal{D}, \lambda = 1.0$	**99.30 ± 0.52**	**86.60 ± 2.11**	**96.20 ± 1.19**	**79.90 ± 2.48**	**61.40 ± 3.02**
$\mathcal{F}, \lambda = 1.0$	**99.80 ± 0.28**	**89.30 ± 1.92**	**94.80 ± 1.38**	**78.40 ± 2.55**	46.30 ± 3.09

Table 6.4.1: Success rates and confidence intervals for pushing and pick-and-place tasks in the Goal2Cube2 scenarios.

subtle task variations like a colour change in a small object the RGB task embedding becomes less informative and TecNet is prone to inferring the wrong task. An investigation into $\mathcal{F}$'s inferior place performance compared to $\mathcal{D}$ reveals a failure mode of $\mathcal{F}$: The controller sometimes struggles to drop a cube above the target pad presumably due to ambiguity in its depth perception. This suggests that the signal provided by the dynamic frame buffer at relatively motion-less pivot points can be ambiguous without additional treatment. When comparing the versions of GEECO which are trained without auxiliary pose supervision ($\lambda = 0.0$) to their fully supervised counterparts ($\lambda = 1.0$), we observe only mild drops in mean performance of up to 15%. Interestingly, the RGB-only ablation $\mathcal{R}$ is least affected by the pose supervision and even improves performance when trained without auxiliary poses. We hypothesise that this is due to the fact that, in the relatively static simulation, RGB features are very representative of their spatial location. Generally, we conclude from the pose supervision ablation that GEECO's performance is not entirely dependent on accurate pose supervision enabling it to be trained on even less extensively annotated demonstration data.

Generalisation to new scenarios. After validating the efficacy of our proposed approach in basic manipulation scenarios which are close to the training distribution, we investigate its robustness and versatility in two additional sets of experiments. In

	FG:Clutter			BG:Prisma		
	reach [%]	pick [%]	place [%]	reach [%]	pick [%]	place [%]
$\mathcal{R}$	63.80 ± 2.98	29.40 ± 2.82	15.80 ± 2.26	0.50 ± 0.44	0.10 ± 0.20	0.00 ± 0.00
$\mathcal{D}$	77.00 ± 2.61	42.10 ± 3.06	20.70 ± 2.51	**94.10 ± 1.46**	**66.00 ± 2.94**	**26.50 ± 2.74**
$\mathcal{F}$	**85.50 ± 2.18**	**62.60 ± 3.00**	**32.60 ± 2.91**	**93.40 ± 1.54**	**62.40 ± 3.00**	19.30 ± 2.45
	NutOnCone			BallInCup		
	reach [%]	pick [%]	place [%]	reach [%]	pick [%]	place [%]
$\mathcal{R}$	32.30 ± 2.90	6.90 ± 1.57	0.40 ± 0.39	21.50 ± 2.55	3.90 ± 1.20	0.10 ± 0.20
$\mathcal{D}$	**66.60 ± 2.92**	**23.30 ± 2.62**	3.30 ± 1.11	**50.60 ± 3.10**	9.70 ± 1.83	0.20 ± 0.28
$\mathcal{F}$	**72.20 ± 2.78**	**26.90 ± 2.75**	**6.20 ± 1.49**	**54.60 ± 3.09**	**16.30 ± 2.29**	**1.50 ± 0.75**

Table 6.4.2: Pick-and-place success rates and confidence intervals of GEECO models trained on Goal2Cube2 and employed in novel scenarios as depicted in Fig. 6.4.2. All models reported are trained with $\lambda = 1.0$.

the following trials we take models $\mathcal{R}$, $\mathcal{D}$ and $\mathcal{F}$ which have been trained on pick-and-place tasks of Goal2Cube2 and apply them to new scenarios probing different aspects of generalisation. We present examples of the four new scenarios in Fig. 6.4.2 and present quantitative results for 1,000 trials in Table 6.4.2. FG:Clutter and BG:Prisma evaluate whether the pick-and-place skill learned by GEECO is robust enough to be executed in visually challenging circumstances as well. The results reveal that the employment of dynamic images for target difference ($\mathcal{D}$) and additionally buffer representation ($\mathcal{F}$) significantly improves task success over the RGB-baseline ($\mathcal{R}$) in the cluttered tabletop scenario due to the perceptual invariances afforded by the dynamic image representation. The effect is even more apparent when the colours of the scene background are distorted. This leads to a complete failure of $\mathcal{R}$ (which is now chasing after flickering colour patterns in the background) while $\mathcal{D}$ and $\mathcal{F}$ can still accomplish the task in about 20% of the cases. In the second set of experiments (cf. Table 6.4.2, bottom), we evaluate whether the pick-and-place skill is versatile enough to be immediately applicable to new shapes and target configurations. NutOnCone requires to drop a nut onto a cone such that the cone is almost entirely hidden. Conversely, BallInCup requires a ball to be dropped into a cup such that only a fraction of its surface remains visible. Besides the handling of unseen object

[a] Due to computational constraints, VFS has only been tested on 100 tasks from the test set.

geometries, both tasks also pose novel challenges in terms of task inference because they feature much heavier occlusions than the original Goal2Cube2 dataset which the model was trained on. Our full model, $\mathcal{F}$, outperforms the ablations significantly in both new scenarios. The encouraging results in both generalisation experiments shine additional light on GEECO's robustness and versatility and suggest that those capabilities can be achieved without resorting to expensive domain randomisation schemes during model training.

6.5 Conclusion

We introduce GEECO, a novel architecture for goal-conditioned end-to-end visuomotor control utilising dynamic images. GEECO can be immediately conditioned on a new task with the input of a single target image. We demonstrate GEECO's efficacy in complex pushing and pick-and-place tasks involving multiple objects. It also generalises well to challenging, unseen scenarios maintaining strong task performance even when confronted with heavy clutter, visual distortions or novel object geometries. Additionally, its built-in invariances can help to reduce the dependency on sophisticated randomisation schemes during the training of visuomotor controllers. Our results suggest that GEECO can serve as a robust component in robotic manipulation setups providing data-efficient and versatile skill primitives for manipulation of rigid objects.

Acknowledgments

Oliver Groth is funded by the European Research Council under grant ERC 638009-IDIU. Chia-Man Hung is funded by the Clarendon Fund and receives a Keble College Sloane Robinson Scholarship at the University of Oxford. This work is also supported by an EPSRC Programme Grant (EP/M019918/1). The authors acknowledge the use of Hartree Centre resources in this work. TheSTFC Hartree Centre is a research collaboratory in association with IBM providing High Performance Computing platforms funded by the UK's investment in e-Infrastructure. The authors also

acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (`http://dx.doi.org/10.5281/zenodo.22558`). Special thanks goes to Frederik Ebert for his helpful advise on adjusting Visual Foresight to our scenarios and to Ștefan Săftescu for lending a hand in managing experiments on the compute clusters. Lastly, we would also like to thank our dear colleagues Sudhanshu Kasewa, Sébastien Ehrhardt and Olivia Wiles for proofreading and their helpful suggestions and discussions on this draft.

Appendix

6.A GEECO Hyperparameters

In this section, we present additional details regarding the architecture and training hyperparameters of GEECO and all its ablations.

Observation buffer. The observation buffer consists of pairs $(I_j, \mathbf{x}_j)$, $j \in [t-K+1, \ldots, t]$ of images I_j and proprioceptive features $\mathbf{x}_j$ representing the K most recent observations of the model up to the current time step t. The images are RGB with a resolution of 256×256 and the proprioceptive feature is a vector of length seven containing the angles of the robot's seven joints at the respective time step. We have experimented with frame buffer sizes $K \in \{2, 4, 6, 8\}$. Buffer sizes smaller than four result in too coarse approximations of dynamics (because velocities have to be inferred from just two time steps) and consequently in lower controller performance. However, controller performance also does not seem to improve with buffer sizes greater than four. We assume that in our scenarios, four frames are sufficient to capture the robot's motions accurately enough, which is in line with similar experiments in prior work (S. James et al. [88]). Therefore, we keep the buffer hyperparameter $K = 4$ fixed in all our experiments. At the start of the execution of the controller, we pad the observation buffer to the left with copies of the oldest frame, if there are less than K pairs in the buffer assuming that the robot is always starting from complete rest.

Convolutional encoder. All convolutional encoders used in the GEECO architecture have the same structure, which is outlined in Table 6.A.1. However, the parameters between the convolutional encoders are not shared. The rationale behind this decision is that the different stacks of convolutions are processing semantically different inputs:

ψ_{OBS} processes raw RGB observations, ψ_{DYN} processes dynamic images representing the motion captured in the observation buffer and ψ_{DIFF} processes the dynamic image difference between the current observation and the target image.

Layer	Filters	Kernel	Stride	Activation
Conv1	32	3	1	ReLU
Conv2	48	3	2	ReLU
Conv3	64	3	2	ReLU
Conv4	128	3	2	ReLU
Conv5	192	3	2	ReLU
Conv6	256	3	2	ReLU
Conv7	256	3	2	ReLU
Conv8	256	3	2	ReLU

Table 6.A.1: The convolutional encoders used in GEECO all share the same structure of eight consecutive layers of 2D convolutions. They take as inputs RGB images with a resolution of 256×256 and return spatial feature maps with a shape of $2 \times 2 \times 256$.

LSTM decoder. The spatial feature maps $\psi_{OBS}(I_t)$, $\psi_{DYN}(\hat{\rho}(I_{t-K+1}, \ldots, I_t))$, $\psi_{DIFF}(\hat{\rho}(I_t, I_T))$ obtained from the convolutional encoders are concatenated to the proprioceptive feature $\mathbf{x}_t$ containing the current joint angles for the robot's 7 DoF. This concatenated tensor forms the state representation $\mathbf{s}_t$, which, in the full model GEECO-$\mathcal{F}$, has a shape of $2 \times 2 \times (256 + 256 + 7 + 256)$. The state is subsequently fed into an LSTM (cf. Fig. 6.3.2). The LSTM has a hidden state $\mathbf{h}$ of size 128 and produces an output vector $\mathbf{o}_t$ of the same dimension at each time step. As shown in prior work (S. James et al. [88]), maintaining an internal state in the network is crucial for performing multi-stage tasks such as pick-and-place.

At the beginning of each task, i. e. when the target image I_T is set and before the first action is executed, the LSTM state is initialised with a zero vector. The output $\mathbf{o}_t$ at each timestep is passed through a fully connected layer $\phi(\cdot)$ with 128 neurons and a ReLU activation function. This last-layer feature $\phi(\mathbf{o}_t)$ is finally passed through four parallel, fully-connected decoding heads without an activation function to obtain the command vectors and the auxiliary position estimates for the object and the end effector as described in Table 6.A.2.

Head	Units	Output
$\hat{\mathbf{u}}_{\Delta EE}$	3	change in EE position $(\Delta x, \Delta y, \Delta z)$
$\hat{\mathbf{u}}_{GRP}$	3	logits for {open, noop, close}
$\hat{\mathbf{q}}_{EE}$	3	absolute EE position (x, y, z)
$\hat{\mathbf{q}}_{POS}$	3	absolute OBJ position (x, y, z)

Table 6.A.2: The output heads of the LSTM decoder regressing to the commands and auxiliary position estimates.

Training details. We train all versions of GEECO with a batch size of 32 for 300k gradient steps using the Adam optimiser (Kingma and Ba [103]) with a start learning rate of 1e-4. One training run takes approximately 48 hours to complete using a single NVIDIA GTX 1080 Ti with 11 GB of memory.

Execution time. Running one simulated trial with an episode length of eight seconds takes about ten seconds for any version of GEECO using a single NVIDIA GTX 1080 Ti. This timing includes the computational overhead for running and rendering the physics simulation resulting in a lower-bound estimate of GEECO's control frequency at 20 Hz. This indicates that our model is nearly real-time capable of continuous control without major modifications.

6.B GEECO Ablation Details

GEECO-$\mathcal{R}$ Our first ablation, which is presented in Fig. 6.B.1, uses a naïve *residual* target encoding to represent the distance to a given target observation in feature space. The residual feature is the difference $\psi_{OBS}(I_T) - \psi_{OBS}(I_j),\ j \in [t-K+1, \ldots, t]$ and should tend towards zero as the observation I_j approaches the target image I_T. Since the same encoder ψ_{OBS} is used for observation and target image, this architecture should encourage the formation of a feature space which captures the difference between an observation and the target image in a semantically meaningful way. Since the observation buffer is not compressed into a dynamic image via $\hat{\rho}(\cdot)$, it is processed slightly differently in order to retain information about the motion dynamics. For each pair $(I_j, \mathbf{x}_j),\ j \in [t-K+1, \ldots, t]$ containing an observed image and a proprioceptive

feature at time step j, the corresponding state representation $\mathbf{s}_j$ is computed and fed into the LSTM which, in turn, updates its state. However, only after all K pairs of the observation buffer have been fed, the command outputs are decoded from the LSTM's last output vector. This delegates the task of inferring motion dynamics to the LSTM as it processes the observation buffer.

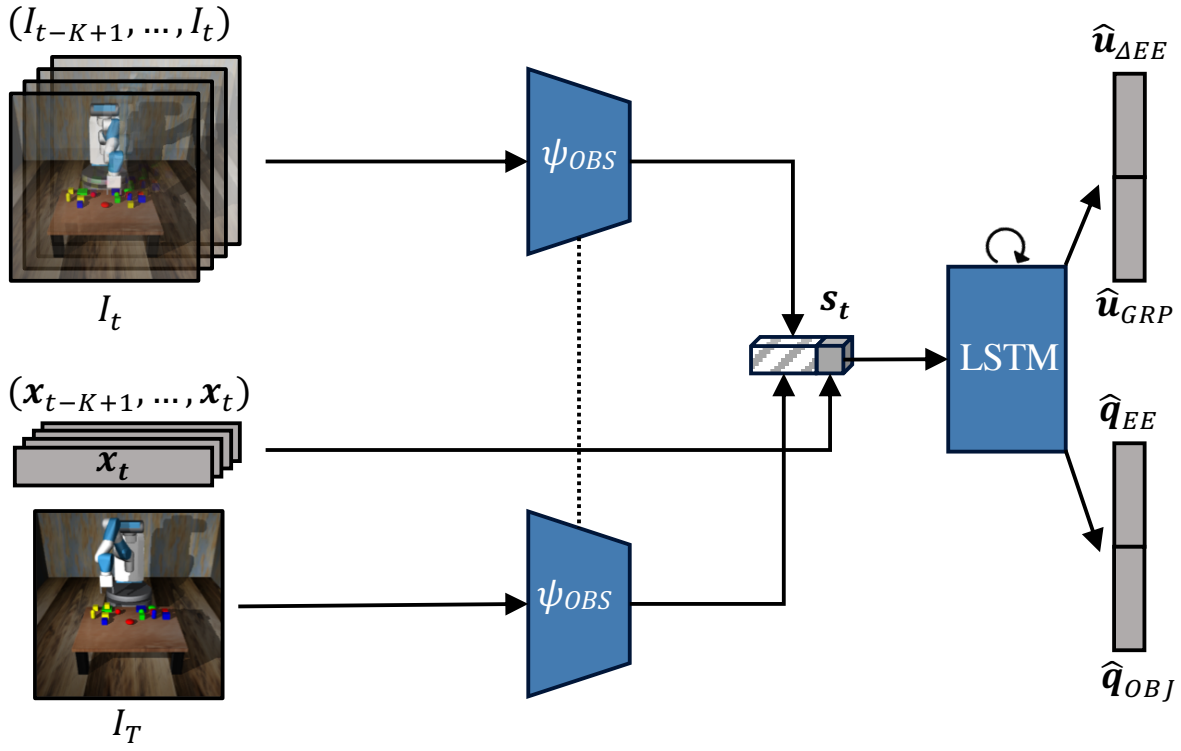


Figure 6.B.1: Model architecture of GEECO-$\mathcal{R}$. The same encoder ψ_{OBS} is used for RGB observations I_t and the target frame I_T. For each observed image I_t, the residual feature w.r.t. to the target image I_T is computed as $\psi_{OBS}(I_T) - \psi_{OBS}(I_t)$, indicated by the striped box in $\mathbf{s}_t$.

GEECO-$\mathcal{D}$ Our second ablation, which is presented in Fig. 6.B.2, uses the dynamic image operator $\hat{\rho}(\cdot)$ to compute the difference between each observed frame I_j, $j \in [t-K+1, \ldots, t]$ and the target image I_T as opposed to GEECO-$\mathcal{R}$ which represents the difference only in feature space. Since the *dynamic difference* $\hat{\rho}(I_t, I_T)$ is semantically different from a normal RGB observation, it is processed with a dedicated convolutional encoder ψ_{DIFF} and the resulting feature is concantenated to the state representation $\mathbf{s}_t$. In order to also capture motion dynamics, the observation buffer is processed sequentially like in GEECO-$\mathcal{R}$ before a control command is issued.

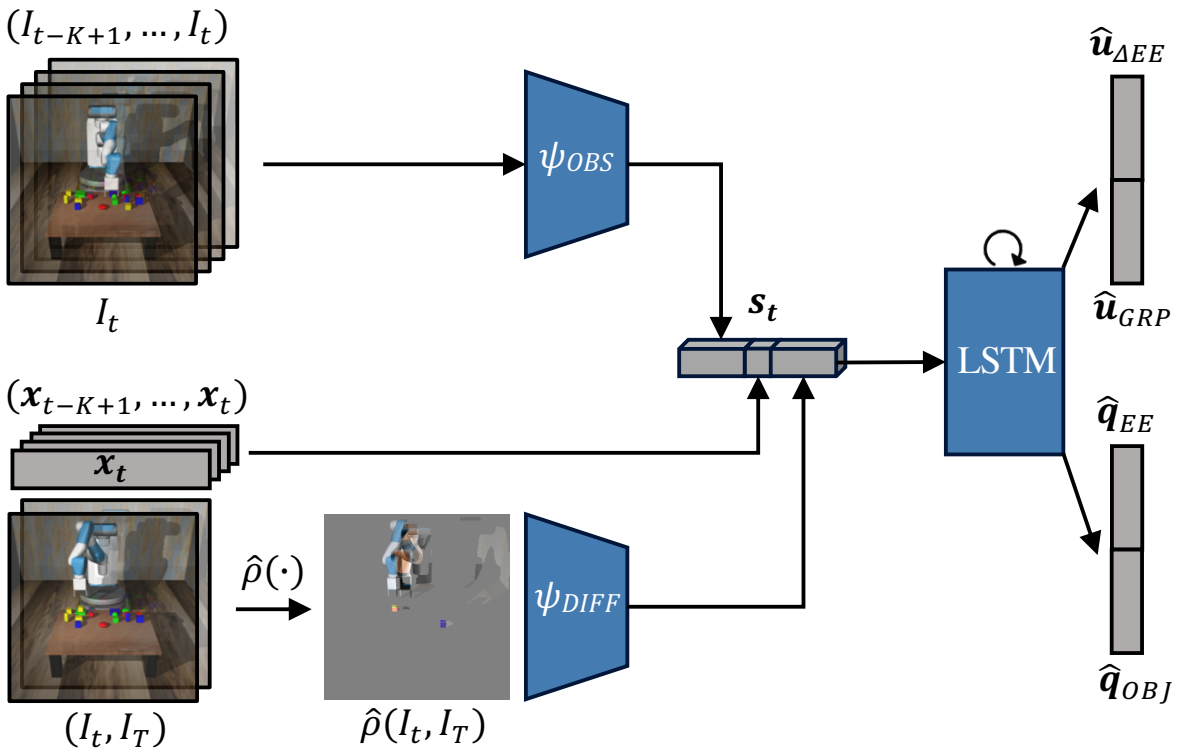


Figure 6.B.2: Model architecture of GEECO-$\mathcal{D}$. For each image I_t in the observation buffer, the *dynamic difference* to the target image I_T is computed using $\hat{\rho}(\cdot)$. The difference image $\hat{\rho}(I_t, I_T)$ is encoded with ψ_{DIFF} before being concatenated to $\mathbf{s}_t$.

6.C E2EVMC Baseline

We compare GEECO to E2EVMC (S. James et al. [88]), an unconditioned visuomotor controller, which we have implemented according to the original paper. We have re-created a similar environment like in the original paper featuring only a red cube and a blue target pad which we call Goal1Cube1. This dataset also consists of 4,000 demonstrations per skill and is split into training, validation and test sets with a ratio of 2 : 1 : 1. Training and testing on this scenario is done to ensure the correct functionality of the model architecture and verify that GEECO performs at least as well as an unconditioned controller in an unambiguous scenario. Even though the task is always the same, i. e. the red cube always goes on top of the blue pad, we still provide GEECO with the target image in every trial. The unconditioned baseline, E2EVMC, runs without a target image since it is trained to perform only one task. We train E2EVMC exactly like GEECO (cf. Section 6.A: Training Details) and select the

best model snapshots according to the task performance on the respective validation sets. We present experimental results for 1,000 trials on the test set of Goal1Cube1 in Table 6.C.1.

model	Pushing		Pick-and-Place		
	reach [%]	push [%]	reach [%]	pick [%]	place [%]
E2EVMC	95.20 ± 1.32	58.60 ± 3.05	**96.10 ± 1.20**	72.00 ± 2.78	**67.60 ± 2.90**
$\mathcal{R}$	**98.80 ± 0.67**	43.70 ± 3.07	**95.30 ± 1.31**	73.00 ± 2.75	60.70 ± 3.03
$\mathcal{D}$	**99.50 ± 0.44**	**87.40 ± 2.06**	**95.80 ± 1.24**	77.40 ± 2.59	**64.90 ± 2.96**
$\mathcal{F}$	**99.20 ± 0.55**	72.90 ± 2.75	**95.90 ± 1.23**	**83.30 ± 2.31**	61.20 ± 3.02

Table 6.C.1: Comparison of pushing and pick-and-place performance of all versions of GEECO with E2EVMC on Goal1Cube1.

We observe that GEECO-$\mathcal{D}$ and -$\mathcal{F}$ perform commensurately with E2EVMC for both pushing and pick-and-place tasks. Both E2EVMC and GEECO reach the red cube nearly perfectly with at least 95% success rate. GEECO-$\mathcal{D}$ performs best on this dataset even outperforming E2EVMC by almost 30% mean success rate for pushing tasks. Again, GEECO-$\mathcal{F}$ sometimes exhibits its failure mode at the pivot point between moving and dropping phase presumably due to ambiguous or uninformative signals from the motion representation around this phase.

6.D Visual Foresight Baseline

In this section, we explain all hyperparameters which have been used during training and evaluation of the Visual Foresight baseline (Ebert et al. [35]).

Video predictor. We use the official implementation[3] of Stochastic Adversarial Video Prediction (SAVP) (Lee et al. [116]) as the video prediction backbone of Visual Foresight. We have not been able to fit the model at a resolution of 256×256 on a single GPU with 11 GB of memory. Hence, we adjusted the image resolution of the video predictor to 128×128 pixels. We use SAVP's hyperparameter set which is

[3] https://github.com/alexlee-gk/video_prediction

reported for the BAIR robot pushing dataset (Ebert et al. [36]) since those scenarios resemble our training setup most closely. We report the hyperparameter setup in Table 6.D.1.

Parameter	Value	Description
`scale_size`	128	image resolution
`use_state`	True	use action conditioning
`sequence_length`	13	prediction horizon
`frame_skip`	0	use entire video
`time_shift`	0	use original frame rate
`l1_weight`	1.0	use L_1 reconstruction loss
`kl_weight`	0.0	make model deterministic
`state_weight`	1e-4	weight of conditioning loss

Table 6.D.1: Hyperparameter setup of SAVP. Hyperparameters not listed here are kept at their respective default values.

Training details. We train SAVP with a batch size of 11 for 300k gradient steps using the Adam optimiser (Kingma and Ba [103]) with a start learning rate of 1e-4. One training run takes approximately 72 hours to complete using a single NVIDIA GTX 1080 Ti with 11 GB of memory.

Action sampling. We use CEM (Rubinstein and Kroese [166]) as in the original VFS paper (Ebert et al. [35]) to sample actions which bring the scene closer to a desired target image under the video prediction model. We set the *planning horizon* of VFS to the prediction length of SAVP, $P = 13$. The action space is identical to the one used in GEECO and consists of a continuous vector representing the position change in the end effector $\mathbf{u}_{\Delta EE} \in \mathbb{R}^3$ and a discrete command for the gripper $\mathbf{u}_{GRP} \in \{-1, 0, 1\}$. Once a target image has been set, we sample action sequences of length P according to the following Eqs. (6.4) and (6.5):

$$\mathbf{u}_{\Delta EE}^{1:P} \sim \mathcal{N}(—, \Sigma) \tag{6.4}$$

$$\mathbf{u}_{GRP}^{1:P} \sim \mathcal{U}\{-1, 0, 1\} \tag{6.5}$$

where $\mathcal{N}(—, \Sigma)$ is a multi-variate Gaussian distribution and $\mathcal{U}\{-1, 0, 1\}$ is a uniform distribution over the gripper states. For each planning step, we run CEM for four iterations drawing 200 samples at each step and re-fit the distributions to the ten best action sequences according to the video predictor, i. e. the action sequences which transform the scene closest to the next goal image. Finally, we execute the best action sequence yielded from the last CEM iteration and re-plan after P steps.

Goal distance. We use L_2 distance in image space to determine the distance between an image forecast by the video predictor and a target image (cf. (Ebert et al. [35])). Since this goal distance is dominated by large image regions (e. g. the robot arm), it is ill suited to capture position differences of the comparatively small objects on the table or provide a good signal when a trajectory is required which is not a straight line. Therefore, we resort to a 'ground truth bottleneck' scheme (S. Nair and Finn [147]) for a fairer comparison. Instead of providing just a single target image from the end of an expert demonstration, we give the model ten *intermediate target frames* taken every ten steps during the expert demonstration. This breaks down the long-horizon planning problem into multiple short-horizon ones with approximately straight-line trajectories between any two intermediate targets. This gives an upper-bound estimate of VFS's performance, if it had access to a perfect keyframe predictor splitting the long-horizon problem. An example execution of VFS being guided along intermediate target frames is presented in Fig. 6.D.1.

Execution time. To account for VFS's sampling-based nature and the guided control process using intermediate target images, we give VFS some additional time to execute a task during test time. We set the total test episode length to 400 time steps as opposed to 200 used during the evaluation of GEECO. VFS is given 40 time steps to 'complete' each sub-goal presented via the ten intermediate target images. However, the intermediate target image is updated to the next sub-goal strictly every 40 time steps, irrespective of how 'close' the controller has come to achieving the previous sub-goal. Running one simulated trial with an episode length of 16 seconds takes about ten minutes using a single NVIDIA GTX 1080 Ti. This timing includes the computational

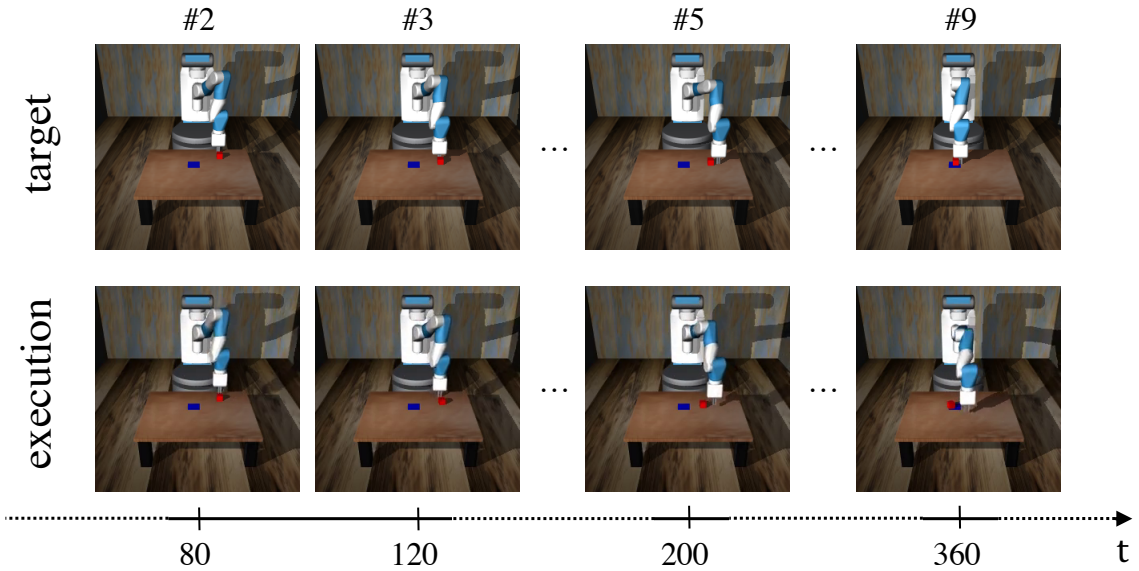


Figure 6.D.1: An execution of VFS with the 'ground truth bottleneck' scheme. The top row depicts intermediate target images from an expert demonstration. The bottom row shows the corresponding state of execution via VFS at time step t.

overhead for running and rendering the physics simulation. While this results in an effective control frequency of 0.7 Hz, a like-for-like comparison between VFS and GEECO can not be made in that regard because we have not tuned VFS for runtime efficiency in our scenarios. Potential speedups can be gained from lowering the image resolution and frame rate of the video predictor, predicting shorter time horizons and pipelining the re-planning procedure in a separate thread. However, the fundamental computational bottlenecks of visual MPC can not be overcome with hyper-paramter tuning: Action-conditioned video prediction remains an expensive operation for dynamics forecasting although pixel-level prediction accuracy is presumably not needed to control a robot. Additionally, the action sampling process is a separate part of the model which requires tuning and trades off accuracy versus execution time. In contrast to that, GEECO provides a compelling alternative by reducing the action computation to a single forward pass through the controller network.

Parameter	Value	Description
iterations	300000	number of gradient updates
batch_size	64	batch size
lr	5e-4	start learning rate of Adam optimiser
img_shape	(125, 125, 3)	image resolution
support	1	k-shot support of new task
query	5	query examples per task during training
embedding	20	size of task embedding vector
activation	elu	layer activation function
filters	16,16,16,16	#(filters) in conv-layers of embedding network
kernels	5,5,5,5	filter size in conv-layers of embedding network
strides	2,2,2,2	stride size in conv-layers of embedding network
fc_layers	200,200,200	neurons of fc-layers of control network
lambda_embedding	1.0	weight of embedding loss
lambda_support	0.1	weight of support loss
lambda_query	0.1	weight of query loss
margin	0.1	margin of the hinge rank loss
norm	layer	using Layer-Norm throughout the network

Table 6.D.2: Hyperparameter setup of TecNet. Hyperparameters not listed here are kept at their respective default values.

6.E TecNet Baseline

We use the official implementation of TecNet for all our experiments[4]. In Table 6.D.2 we provide a comprehensive list of all hyperparameters used in our experiments with TecNet.

Training details. We train TecNet in the one-shot imitation setup and provide one 'demonstration' before the start of the controller execution consisting of only the first observation and the target image. During training and evaluation, we resize all images fed into TecNet to 125×125 pixels as per the original paper. We train TecNet with a batch size of 64 for 300k gradient update steps using the Adam optimiser (Kingma and Ba [103]) with a start learning rate of 5e-4. One training run takes about 72 hours to complete using a single NVIDIA GTX 1080 Ti with 11 GB of memory.

[4] https://github.com/stepjam/TecNets

Execution time. Running one simulated trial with TecNet with an episode length of eight seconds takes about eight seconds using a single NVIDIA GTX 1080 Ti. This timing includes the computational overhead for running and rendering the physics simulation resulting in a lower-bound estimate of TecNet's control frequency at 25 Hz. This makes TecNet also a viable option for real-time visuomotor control without any system modifications.

7

Is Curiosity All You Need? On the Utility of Emergent Behaviours from Curious Exploration

In this chapter we investigate behaviours which emerge in an embodied agent when exploring its environment to learn a forward predictive model of the environment's dynamics. To this end, we propose a novel, off-policy curiosity learning setup and apply it to two robotic control domains: a 9-DoF robot arm and a 20-DoF humanoid. We demonstrate that complex, human-interpretable behaviour emerges and vanishes over time when the agent optimises its policy according to the shifting curioisty objective. Our observations give rise to the conjecture that this emergent behaviour can serve as a useful skill set for an agent to learn new downstream tasks. We analyse the utility of the emergent behaviour in a hierarchical learning setup where self-discovered skills from curious exploration are reused to learn a new manipulation task more quickly. We find that such a setup can perform commensurately with a specifically designed reward curriculum in terms of learning speed which suggests that the discovery and retention of emerging behaviour can open a fruitful new avenue in unsupervised reinforcement learning. This work was presented at the 4th

Robot Learning Workshop at NeurIPS 2021 and is published as a preprint:

O. Groth, M. Wulfmeier, G. Vezzani, V. Dasagi, T. Hertweck, R. Hafner, N. Heess, and M. Riedmiller. "Is Curiosity All You Need? On the Utility of Emergent Behaviours from Curious Exploration". In: *arXiv preprint arXiv:2109.08603* (Sept. 2021)

Abstract

Curiosity-based reward schemes can present powerful exploration mechanisms which facilitate the discovery of solutions for complex, sparse or long-horizon tasks. However, as the agent learns to reach previously unexplored spaces and the objective adapts to reward new areas, many behaviours emerge only to disappear due to being overwritten by the constantly shifting objective. We argue that merely using curiosity for fast environment exploration or as a bonus reward for a specific task does not harness the full potential of this technique and misses useful skills. Instead, we propose to shift the focus towards retaining the behaviours which emerge *during* curiosity-based learning. We posit that these self-discovered behaviours serve as valuable skills in an agent's repertoire to solve related tasks. Our experiments demonstrate the continuous shift in behaviour throughout training and the benefits of a simple policy snapshot method to reuse discovered behaviour for transfer tasks.

7.1 Introduction

Intrinsic motivation (Baranes and Oudeyer [11], Oudeyer et al. [154], Oudeyer and Kaplan [155], and Schmidhuber [170, 171]) can be a powerful concept to endow an agent with an automated mechanism to continuously explore its environment in the absence of task information. One common way to implement intrinsic motivation is to train a predictive model alongside the agent's policy and use the model's prediction error as a reward signal for the agent encouraging the exploration of previously unfamiliar transitions in the environment - a method also known as *curiosity learning* (Pathak et al. [157]). Curiosity-esque reward schemes have been used in different ways to facilitate exploration in sparse tasks (Burda et al. [24] and Houthooft et al. [82]) or pre-train policy networks before fine-tuning them on difficult downstream tasks (Sekar et al. [174]). In environments where the main task objective is highly correlated with thorough exploration, curiosity-based approaches have also

been shown to solve the main task without any additional reward signal (Burda et al. [23]).

However, in environments with multiple possible tasks – e. g. in manipulation scenarios where objects could be interacted with or re-arranged in different ways – not only the final behaviour of a curious exploration run might be of interest, but intermediate behaviours can correlate with solutions to different tasks. Naturally, the constantly changing curiosity objective leads to the emergence of diverse behaviours during training – much akin to the learning process of infants which develop useful skills by playing (Haber et al. [70]). Yet, only a fraction of this diversity is ultimately retained in a downstream, task-specific policy – which might also be biased towards exploration as a side-effect – or it is even completely overwritten after fine-tuning. This problem of *catastrophic forgetting* (McCloskey and Cohen [136]) is well-known for any neural network which operates under a shifting data distribution. However, the intermediate behaviours which emerge and disappear *during* learning based on curiosity can be of relevance for different tasks of interest. If we were able to extract and leverage emergent behaviour, we could turn the process of exploration from a service for task-driven reinforcement learning into a rich continual learning setup in its own right (Hadsell et al. [71]).

Despite technical challenges, the discovery of self-induced curricula of skills holds a tantalising prospect for an agent's ability to solve broad sets of long-horizon tasks when it is able to draw upon potentially useful skills. For instance, a robotic arm which has already discovered how to reach, grasp and lift objects in its workspace has a much easier time exploring and learning a policy to stack objects later on, if it recombined its previously acquired skills. Recent works (Hertweck et al. [77], Riedmiller et al. [161], and Wulfmeier et al. [215]) have studied the influence of specific tasks across a spectrum of manual engineering effort like in the stacking example on the learning success of complex manipulation policies. A curiosity-based approach could further reduce the effort of designing a curriculum of tasks and reward functions with a set of self-discovered skills.

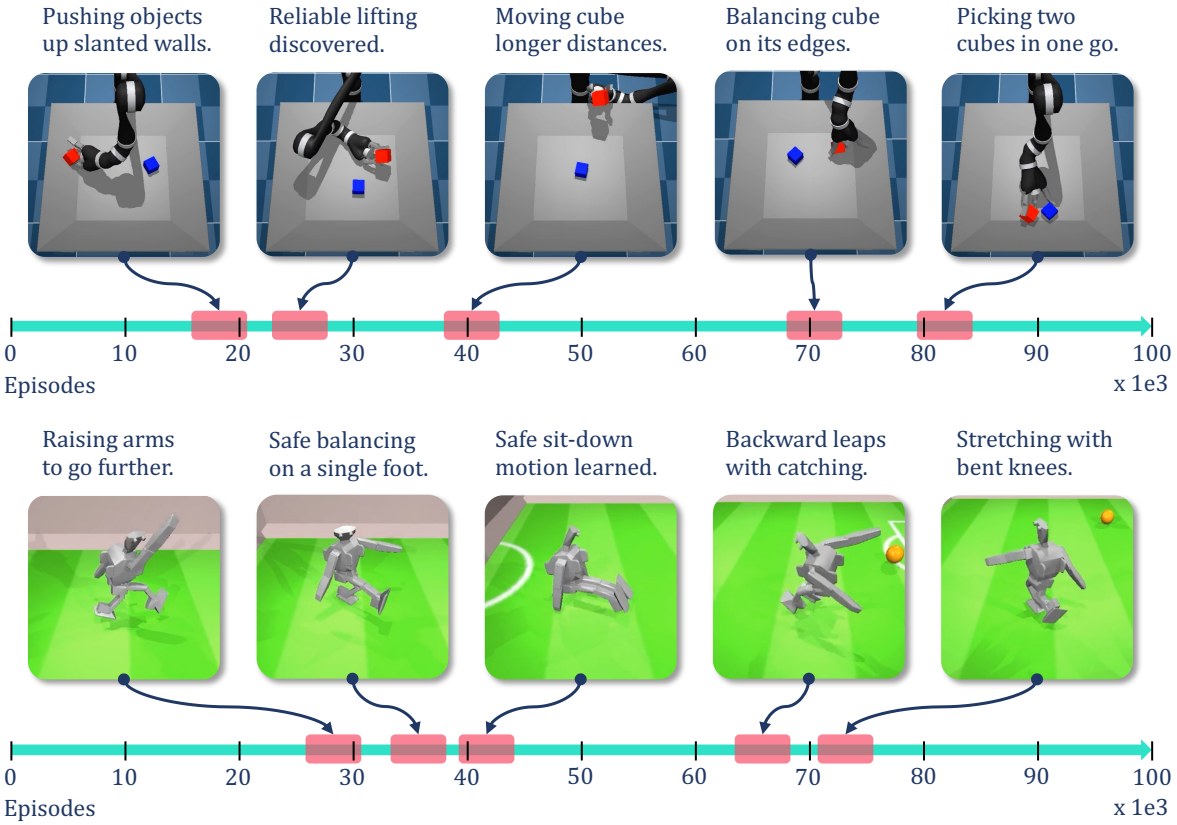


Figure 7.1.1: Two example timelines depicting the emergence of behaviour while pursuing a curiosity objective on a 9-DoF JACO arm (top) and on a 20-DoF OP3 humanoid robot (bottom). Each timeline represents the evolution of behaviour from a single random seed on a single simulated actor. At each point in time, the agent exhibits a single behaviour which slowly evolves over time as the curiosity objective changes. A detailed description of the emergent behaviour in this experiment is provided in Section 7.4.1 and corresponding quantitative results are shown in Figs. 7.4.1 and 7.4.2. The corresponding videos can be found at: `https://dpmd.ai/dm-selmo`.

In this paper, we study behaviour which emerges based on a curiosity objective in two continuous control settings: manipulation and locomotion. In contrast to prior work in this domain, we implement curiosity-based exploration in an off-policy learning setting which improves upon on-policy implementations in terms of data-efficiency and presumably increases the diversity of emerging behaviours. Furthermore, we look at the utilisation of the self-discovered behaviour for learning new downstream tasks. In particular, we find that even a naïve baseline which treats snapshots of a curiosity policy as fixed exploration skills in a hierarchical learning setup can perform commensurately with a hand-designed curriculum learning setup. Our findings suggest that the identification and exploitation of self-discovered behaviours can be a

fruitful avenue for future work in the learning of complex robotics tasks.

In summary, we make the following two contributions: First, we introduce *SelMo*, an off-policy realisation of a self-motivated, curiosity-based method for exploration which is applied to two robotic manipulation and locomotion domains in simulation. We show that even in those complex, 3D environments, meaningful and diverse behaviour emerges solely based on the optimisation of the curiosity objective. Second, we propose to extend the focus in the application of curiosity learning towards the identification and retention of emerging intermediate behaviours and support this conjecture with a baseline experiment which uses self-discovered behaviours as auxiliary skills in a hierarchical reinforcement learning setup.

7.2 Related Work

The utilisation of forward modelling error as reward signal has been implemented in deterministic and probabilistic settings (Achiam and Sastry [2] and Shelhamer et al. [178]) and is commonly known as *curiosity learning*. The error reward signal encourages the exploration of unfamiliar parts of the state-action space which are not yet well-predictable. Pathak et al. [157] derive the curiosity reward from an inverse dynamics model which is simultaneously less prone to be confused by unpredictable elements of the environment (cf. *white-noise problem* (Schmidhuber [171])). Burda et al. [24] use a randomly initialised projection from observation into latent space and the predictor driving the learning process is tasked with learning this projection. S. H. Huang et al. [84] use the error of a predictive model of penalties in dexterous manipulation and find that the additional intrinsically motivated exploration helps in the development of gentle grasping policies. Curiosity learning can also be seen through an information theoretic lens using an information gain objective like in Still and Precup [187] or Houthooft et al. [82]. Lastly, the *disagreement* between an ensemble of forward models (Pathak et al. [158]) can also be treated as a proxy for model uncertainty and exploited as a reward signal. In addition to serving as a reward generator, the predictors can also be used for targeted exploration. Lowrey

et al. [131] and Sekar et al. [174] employ world models to deliberately explore in regions where the expected prediction error is high *a priori* - as opposed to realising that something surprising has been observed *a posteriori* like in standard curiosity approaches.

Besides curiosity learning, the literature features an extensive body of work on structured exploration methods in reinforcement learning. Classical methods like count-based exploration schemes have been revisited in continuous settings, e. g. in Bellemare et al. [16] changes in state density estimations are used as exploration bonus rewards. Another line of work revolves around the central notion of *empowerment* (Klyubin et al. [105]) which aims to find new behaviours which are increasingly controllable by the agent (Gregor et al. [63] and Mohamed and Rezende [140]). The *estimation of model learning progress* has also been discussed as a reward signal for exploration but this approach is much harder to implement as it relies on a measure of model improvement (Lopes et al. [130]). Closely related to model-improvement-based exploration is the idea of *PowerPlay* (Schmidhuber [172]) which describes an inductive task proposal scheme equipping the agent with a mechanism which infers the current frontier of soluble tasks and inductively creates a novel task which can be solved by employing the agent's current knowledge. However, this exploration scheme has mostly remained conceptual so far with experiments limited to simple pattern recognition tasks (Srivastava et al. [184]). Related to PowerPlay's idea of task proposition but supposedly more tractable is the concept of *self-play* (Sukhbaatar et al. [189]). In this paradigm, two agents play a competitive game where one player is rewarded for inventing a behaviour which the other agent cannot imitate. This approach has recently led to the emergence of highly complex object manipulation behaviour on a simulated robotic arm (OpenAI et al. [152]).

The importance of diverse interactions with the environment – similar to the playful behaviour exhibited by children – has been stressed in recent works in reinforcement learning, e. g. in Haber et al. [70] or Lynch et al. [133]. Such intrinsically motivated

exploration has been shown to lead to the discovery of diverse behaviour and re-usable skills as a natural 'byproduct' of interacting with the environment (Singh et al. [181]). Several recent works have identified the *diversity of behaviour* as a central objective to optimise for during unsupervised exploration (Eysenbach et al. [42] and Sharma et al. [177]). Sharma et al. [176] have extended this line of work and also showed that the acquired latent 'skill space' can be leveraged in model-predictive control fashion for goal-conditioned navigation on a real-world quadruped.

Our work builds upon the curiosity learning approach utilising the forward prediction error of a dynamics model as a reward signal. However, in contrast to typical curiosity setups (Burda et al. [23]) which are optimised *on-policy* we employ an *off-policy* method to train the agent. Our method is also set apart from prior art with regards to the utilisation of self-discovered behaviour. Instead of using *model-predictive control* (Sharma et al. [176]), we leverage emergent behaviour directly by employing policy snapshots as modular skills in a mixture policy (Wulfmeier et al. [214, 216]).

7.3 Method

In this section, we present SelMo – a self-motivated exploration method which optimises a curiosity objective in an off-policy fashion. Our system is designed around two key components: A forward dynamics model $f_{dyn}: S \times A \mapsto S$ which aims to approximate the state transition function of the environment and a policy $\pi(a_t|s_t)$ which aims to take transitions in the environment for which the prediction error of f_{dyn} is high.

Given potentially different requirements for the dynamics model and the exploration policy, the implementation builds on two separate learning processes. We have decided to implement the system in this distributed way to more easily realise the data labeling process and have convenient control over learning rate and data flow parameters. However, other implementations which fuse both learning processes and data buffers are also conceivable. Crucially, our setup deviates from recent

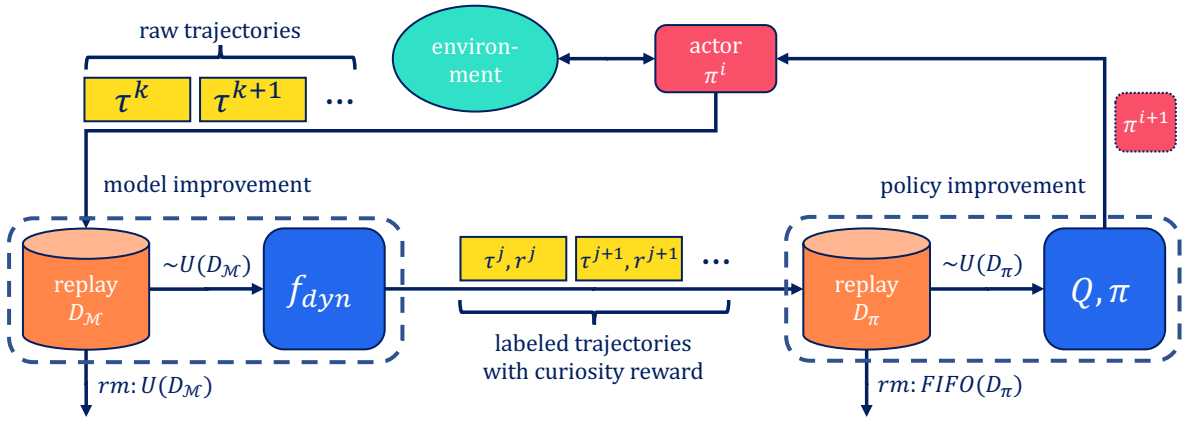


Figure 7.2.1: An overview over the SelMo system architecture. The agent collects trajectories $\tau^k, \tau^{k+1}, \ldots$ in the environment using its current policy π^i and stores it in a model replay buffer $D_{\mathcal{M}}$. When $D_{\mathcal{M}}$ is full, trajectories are replaced with a uniform removal strategy. The dynamics model f_{dyn} samples uniformly from this buffer and updates its parameters for forward prediction using *stochastic gradient descent* (SGD). The sampled trajectories $\tau^j, \tau^{j+1}, \ldots$ are then assigned a curiosity reward $r^j, r^{j+1}, \ldots$ scaled by their respective prediction error under the current $f_{dyn}^{(j)}$. The labeled trajectories are passed on to the policy replay buffer D_π which runs a FIFO removal strategy. *Maximum a posteriori policy optimisation* (MPO) is used to fit Q-function and policy π based on uniformly drawn samples from the policy replay. The resulting policy π^{i+1} is then synced back into the actor. Note that both model and policy learning is executed in independent loops.

on-policy approaches for curiosity-based rewards in two important aspects: First, we optimise in an off-policy fashion based on a diverse set of experienced transitions in the environment. In this way, the policy is encouraged to to cover a broader spectrum of exploration avenues as the learning evolves. Second, we employ an approximate but efficient data labelling strategy which only assigns curiosity rewards to trajectories when they are used to update the dynamics model, but refrain from relabeling all trajectories in the policy replay after every model update. We provide a general overview over the whole system in Fig. 7.2.1 and explain each component in detail in the paragraphs below.

Model replay. The fixed-size model replay buffer $D_{\mathcal{M}}$ stores trajectories $\tau^k = [(s_0^k, a_0^k, s_1^k), \ldots, (s_{T-1}^k, a_{T-1}^k, s_T^k)]$ collected by the actor in the environment. The values of all environment transitions (s_t, a_t, s_{t+1}) are normalised in the range $[-1, 1]$. Batches of training data for the world model are sampled uniformly from this buffer as $\mathcal{B}_{\mathcal{M}} := \{\tau^1, \ldots, \tau^B\} \sim \mathcal{U}(D_{\mathcal{M}})$. In order to preserve a diverse sampling of environment

transitions during the whole learning process, each trajectory in $D_{\mathcal{M}}$ can be sampled at most n_{max} times. Additionally, old trajectories in the buffer are replaced by new ones at random when the buffer size limit is exceeded.

World model. We describe the environment in which the agent operates as $\mathcal{E} = (S, A, P)$ with state and action spaces S and A as well as a state transition function $s_{t+1} = P(s_t, a_t)$ over discrete time steps which describes the environment's *transition dynamics*. The world model is a forward-predictive model $f_{dyn} \colon S \times A \mapsto S$ which approximates the environment's transition dynamics as:

$$\hat{s}_{t+1} = f_{dyn}(s_t, a_t; \theta) \tag{7.1}$$

In our case, Eq. (7.1) is implemented as a two-layer MLP with parameters θ. Besides estimating the transition dynamics from observed data, the world model plays a crucial role in assigning the reward for each observed transition (s_t, a_t, s_{t+1}). When a transition is evaluated by the world model, the assigned reward is scaled by the model's current prediction error.

$$r^{(j)}(s_t, a_t, s_{t+1}) = \tanh(\eta_r * (f^{(j)}_{dyn}(s_t, a_t) - s_{t+1})^2) \tag{7.2}$$

The 'state' of the world model is indicated by the number of gradient updates j which have been performed on it so far. We scale the reward via a hyperparameter η_r and pass it through a tanh to keep it bounded for the downstream policy learning procedure. When a new batch of data $\mathcal{B}$ is sampled from the model replay, the world model performs two operations. First, it labels each $\tau \in \mathcal{B}$ by assigning curiosity rewards r^C according to Eq. (7.2). Second, it performs one gradient update $\theta^{(j+1)} \leftarrow \theta^{(j)} + \eta_{\mathcal{M}} \partial \mathcal{L}^{(j)}_{dyn}(\mathcal{B}) / \partial \theta^{(j)}$ by minimising its prediction loss:

$$\mathcal{L}^{(j)}_{dyn}(\mathcal{B}) = \sum_{\tau \in \mathcal{B}} \sum_{(s_t, a_t, s_{t+1}) \in \tau} (f^{(j)}_{dyn}(s_t, a_t) - s_{t+1})^2 \tag{7.3}$$

After one world model update, the relabeled batch of trajectories $\tilde{\mathcal{B}}$ is stored in the policy replay buffer D_π.

Policy replay. The fixed-size policy replay D_π stores tuples (τ^j, r^j) representing trajectories which have been labeled with curiosity rewards by the world model. This off-policy setup provides the policy learner with a diverse training set to optimise for useful exploration actions *globally* and not only in the vicinity of the most recent experience. During policy learning, data batches are sampled uniformly from this buffer as $\mathcal{B}_\pi := \{(\tau^1, r^1), \ldots, (\tau^B, r^B)\} \sim \mathcal{U}(D_\pi)$. Similar to the model replay, each tuple can be sampled up to m_{max} times to increase the utilisation of each data point during policy learning. However, the removal strategy of the policy replay is FIFO to ensure that trajectories with the most outdated curiosity rewards get replaced first to reflect the change in the world model, albeit with a certain delay.

Policy. The *Markov Decision Process* (MDP) which is induced by this setup can be written as: $\mathcal{M}^{(j)} = (S, A, P, r^{(j)})$. Since the world model $f^{(j)}_{dyn}$ changes with every gradient update, the reward function $r^{(j)}$ changes continuously. Consequently, the reward varies with the model training timesteps j and the policy π is required to keep adapting. The policy replay D_π is filled by the world model's training loop. It contains data of the most recent $\kappa = |D_\pi|/|\mathcal{B}|$ versions of the MDP. Hence, the resulting policy is optimising for a mixture of MDPs $\{(S, A, P, r^{(\iota)} \mid \iota \in [j - \kappa, \ldots, j]\}$. Policy and critic network are implemented as two separate MLPs. The policy is optimised off-policy using MPO (Abdolmaleki et al. [1]) and a separate learning rate η_π.

7.4 Experiments

This section is split in two parts. First, we report our analysis of the emergence of behaviours when optimising only for a curiosity reward in Section 7.4.1. Second, we present an empirical utilisation of self-discovered behaviour for accelerated learning of new downstream tasks in Section 7.4.2. For details regarding the simulation domains and model hyperparameters, we refer the reader to Section 7.A and Section 7.B respectively.

7.4.1 Emergence of Behaviour

We start our investigation with an analysis of the behaviour which emerges in our two simulated domains as depicted in Fig. 7.1.1. The JACO domain features a 9 DoF robotic arm and two cubes; the OP3 domain features a 20 DoF humanoid robot. For this experiment, we run the SelMo learning loop (cf. Fig. 7.2.1) with a single actor for 100K episodes on each of the environments. During training, the agent solely optimises its curiosity objective which is defined by Eq. (7.2). The visual inspection of the experiments reveal that in both cases, diverse sets of human-interpretable behaviour emerge consistently and are exhibited by the agent for extended periods of time before the ever-changing curiosity reward function shifts the learning towards a new behaviour. Below, we describe the observed behaviours in each domain in greater detail.

Emergent Manipulation Behaviour on JACO

A qualitative example timeline of emerging behaviours on the JACO arm is depicted in Fig. 7.1.1 (top) and supplemented by a plot evaluating reaching and lifting behaviour during the run in Fig. 7.4.1. We find that the agent is very quickly driven towards both cubes with equal attention and starts interacting with them by pushing them around. Soon thereafter, it discovers that pushing them up the slanted walls of the bin facilitates picking them up before it stably latches onto a mode in which it prefers manipulating the red cube over the blue one after approximately 15K episodes. This also coincides with a first period of sustained lifting of the red cube. We hypothesise that the discovery of lifting said cube reinforces the interaction with it as it opens up a new dimension along which model prediction error can be rewarded: the height of the object. After about 25K episodes, it has learned to pick up an object reliably even without the help of the slopes.

After approximately 40K episodes, the curiosity objective pushes the agent to deliberately take objects outside of the workspace and perform pick-and-place operations which move a cube over a long distance but at a lower height. Interestingly, the policy does not degenerate into extreme behaviours like spinning motions which have been

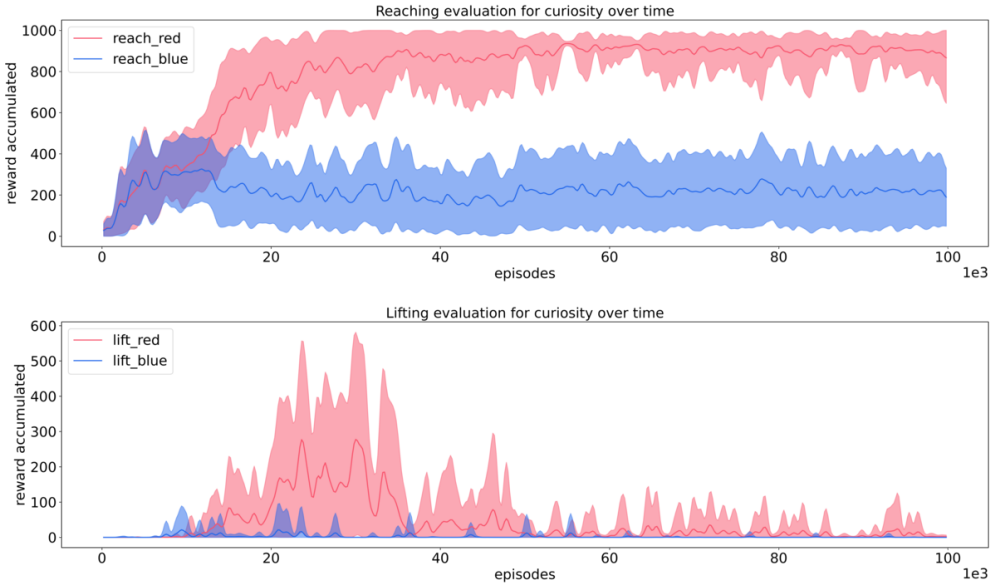


Figure 7.4.1: Manipulation task evaluation in the JACO environment over the lifetime of one experiment while the agent is only trained on the curiosity objective (cf. Eq. (7.2)). A snapshot of the curiosity policy is saved every 100 episodes and evaluated on reaching and lifting the red and blue cubes respectively. Mean and standard deviation are plotted over 20 evaluation runs per policy snapshot and the plot is smoothed with an exponential filter of $\sigma = 1.5$.

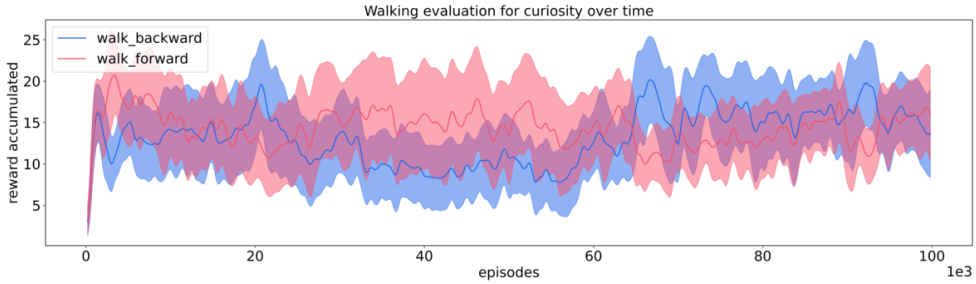


Figure 7.4.2: Task evaluation in the OP3 environment over the lifetime of one experiment while the agent is only trained on the curiosity objective (cf. Eq. (7.2)). A snapshot of the curiosity policy is saved every 100 episodes and evaluated on locomotion (`walk_{forward,backward}`) tasks. Mean and standard deviation are plotted over 20 evaluation runs per policy snapshot and the plot is smoothed with an exponential filter of $\sigma = 1.5$.

observed in related work (Sekar et al. [174]) but stays focused on the objects and keeps exploring their physical properties. For instance, at around 70K episodes, the policy investigates the stability of cube poses in a targeted way by balancing them on their edges and corners. Finally, after about 80K episodes, it starts exploring the possibilities of moving both cubes simultaneously.

Emergent Locomotion Behaviour on OP3

Similar to the JACO arm, we present a timeline of emerging behaviour on the OP3 in Fig. 7.1.1 (bottom) and a corresponding evaluation of locomotion behaviour in Fig. 7.4.2. Unsurprisingly, the agent spends roughly the first 2K episodes – indicated by the steep rise in walking rewards – just on learning a sense of balance because an episode is terminated early when the torso constraint (cf. Section 7.A) is violated and the agent is about to fall. This also corresponds to maximising the experienced episode length because this increases the chances of further increasing the accumulated reward. This finding is in line with earlier work (Pathak et al. [157]) which has shown that the avoidance of a 'death' event is a natural byproduct of curiosity-driven learning with a positive reward and favourably shapes the emerging policy.

Once the agent has learned to stay upright, it slowly starts to develop basic locomotion in the form of stumbling forwards and backwards with only small foot lifting heights which is also reflected in minor oscillations during the evaluation of the walking rewards in Fig. 7.4.2. Interestingly, after around 30K episodes, the agent has discovered to swing its arms to take bigger steps. This is most impressively first demonstrated after nearly 40K episodes when the agent balances on one foot while stretching out the other leg using its arms for counterbalancing moves. Using the arms also opens new avenues for exploration. Approximately 40K episodes into training, the agent has learned to catch itself when falling backwards. This leads to the discovery of a sit-down behaviour which does not violate the environment's torso constraints. After the agent has explored various 'ground exercises' it switches back to walking gaits at around 55K episodes. Then, the whole body movement has become considerably more nimble and its movement repertoire now features quick turns, stumbling reflexes and even safe backward leaps. After about 70K episodes the agent starts revisiting earlier behaviour, e. g. the balancing skill, but keeps adding variations to it like knee-bending or stretching.

7.4.2 Utilisation of Emergent Behaviour

As we have discussed in the previous section, the constantly evolving curiosity policy develops behaviours which correspond to the solution of concrete tasks (cf. Fig. 7.4.1, Fig. 7.4.2). In order to retain those diverse behaviours to accelerate the learning of new tasks, we have devised the following experiment: Assuming that the curiosity policies exhibit undirected yet versatile exploratory behaviour, we investigate how well they could serve as auxiliary skills in a modular learning setup of a downstream task.

We employ *Regularized Hierarchical Policy Optimization* (RHPO) (Wulfmeier et al. [214]) as this framework allows us to compose multiple policies in a hierarchical manner. In each environment, we define a downstream target task which we are interested in learning and provide five policy snapshots from a SelMo experiment as auxiliary exploration skills. During the SelMo experiment, we optimise solely for the curiosity objective and save a snapshot of the curiosity policy every 100 episodes. The RHPO experiment subsequently samples SelMo policy snapshots and utilises the behaviour exhibited by them to assist the exploration for the desired downstream task.

For this experiment, we refer again to the JACO environment where we define the target task to be `lift_red`. While the policy for the target task is randomly initialised, the auxiliary policies are randomly sampled from the SelMo snapshots and kept fixed during the entire RHPO training. We differentiate between three different phases from which the SelMo snapshots are chosen: `early` comprises of snapshots which have been trained on up to 10K episodes, `mid` snapshots are from the interval between 10K and 20K episodes and `late` snapshots are sampled between 20K and 30K training episodes. In Fig. 7.4.3 we compare the learning progress of the downstream task with the sampled SelMo auxiliary skills against a baseline featuring hand-designed task curricula in an SAC-X framework (Riedmiller et al. [161]). In the case of the JACO environment, the agent is given a curriculum of reward functions which help to `reach` and `move` the `red` cube.

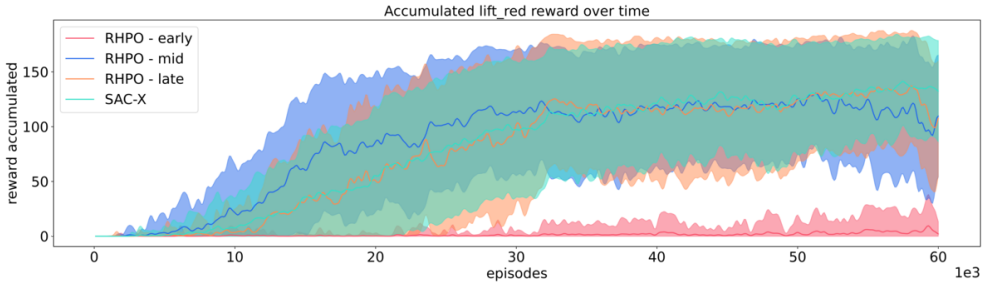


Figure 7.4.3: Learning curves for hierarchical skill learning of `lift_red` using SelMo auxiliary policies in the JACO environment. The SAC-X baseline uses auxiliary reward functions `{reach,move}_red`. Each RHPO run uses five randomly sampled SelMo policies from the respective intervals as auxiliary skills (cf. Section 7.4.2). Mean and standard deviation are plotted for five random seeds for each model and the plot is smoothed with an exponential filter of $\sigma = 1.5$.

In the case of `lift_red` on JACO, we find that SelMo auxiliaries from the `mid` and `late` exploration periods give the learning of the lifting policy a significant boost which is commensurate with a tuned SAC-X baseline featuring multiple auxiliary rewards which have been hand-designed to facilitate the learning of `lift_red`. This result is in line with the performance observed in Fig. 7.4.1 where the curiosity-based policy has developed a sustained lifting behaviour between 20K and 35K episodes. Consequently, snapshot auxiliaries sampled from that range are particularly useful when a targeted lifting policy is to be learned.

This experiment shows that even a simple behaviour retention strategy like policy snapshotting can already provide clear benefits for downstream learning of new tasks. The self-discovered behaviours from a curiosity training phase afford a task learning scaffold which can be commensurate with a specifically designed set of auxiliary reward functions. This is a promising result suggesting that independent curious exploration could be used in lieu of human-engineered task curricula in complex manipulation scenarios.

7.5 Discussion

Our experiments have shown that complex manipulation and locomotion behaviour such as grasping, lifting, balancing, sitting and leaping emerges completely unsuper-

vised in an off-policy curiosity learning setup on a 9 DoF robot arm and a 20 DoF humanoid. This observation supports our hypothesis that self-discovered behaviour can provide a valuable skill repertoire for the learning of new downstream tasks. We add further evidence to this hypothesis by utilising randomly selected policy snapshots from a curiosity training as auxiliary skills in a modular learning setup and show that they provide a learning scaffold commensurate with hand-designed auxiliary reward functions for the respective tasks. This suggests that curiosity-based exploration should be treated as an independent aspect of a learning system as opposed to a mere bonus reward or policy pre-training phase. In this study, we provide a baseline for harnessing self-discovered behaviour by randomly sampling policy snapshots and treating them as fixed skills in a modular learning setup. However, more sophisticated techniques for identification, retention and utilisation of behaviour in curious exploration settings are conceivable and will be briefly touched upon in this discussion section.

Identification of emerging behaviour. As we have shown in Section 7.4.1, complex behaviour can emerge in an unsupervised way when optimising a policy for a curiosity objective. However, it is revealed in Fig. 7.4.1 and Fig. 7.4.2 that pre-conceived reward functions are only able to capture a fraction of the emergent behaviour like reaching and lifting of individual objects or basic walking gaits. More involved behaviours are not covered by basic reward functions and implementing reward functions to identify a broad set of behaviour a priori does not scale well with the unsupervised nature of curiosity learning. Diversity-based approaches (Eysenbach et al. [42] and Sharma et al. [176, 177]) address this challenge via a latent 'skill space' which modulates the policy network to capture different modes of operation like jumping and walking. The temporal and hierarchical abstraction provided by the latent vector also facilitates planning over the skill space. However, the identification of particularly useful skills worth retaining and comparatively useless skills which could be overwritten is still an open question, especially in never-ending learning settings where a curiosity-driven exploration keeps discovering new behaviours or revisiting old ones.

Retention and utilisation of self-discovered behaviour. Closely related to the question of behaviour identification is the question of its retention and utilisation. In our setup we have treated snapshots of the same curiosity policy as different skills as they represent different behaviour over time. Using a latent skill space as an arbiter for different behaviours in the same policy network (Eysenbach et al. [42] and Sharma et al. [177]) has also been shown to work effectively as this formulation also enables planning over the learned space of skills. However, Lynch et al. [133] have also demonstrated the utility of raw *play data* in the training of versatile goal-conditioned policies which would position the curiosity-driven exploration as a data collector for a downstream policy distillation instead of a re-usable behaviour in itself. Lastly, Riedmiller et al. [161] and Hertweck et al. [77] have shown the benefits of curricula of reward functions for the learning of complex manipulation tasks which opens another avenue for the utilisation of curiosity-based learning: Instead of using frozen snapshots of the policy as a repertoire of skills one could also treat different versions of the world model as a set of distinct reward functions to encourage the optimisation for diverse behaviour during learning of new downstream tasks.

7.6 Conclusion

In this paper we have studied the emerging behaviour when optimising an exploration policy for a curiosity objective derived from a forward-predictive world model. To this end, we have presented SelMo, a curiosity-based, off-policy exploration method and applied it in two continuous control domains: a simulated robotic arm and humanoid robot. We have observed that complex behaviour emerges in both settings and provided a baseline for the utilisation of this self-discovered behaviour in a modular downstream learning scenario. Despite the remaining technical challenges, we believe that the automatic identification and retention of useful emerging behaviour from curious exploration is a fruitful avenue of future investigation in unsupervised reinforcement learning.

Acknowledgements

We would like to thank Arunkumar Byravan, Dushyant Rao, Abbas Abdolmaleki, Yuval Tassa and Nathan Lambert for the discussions and feedback during the conception and execution of this project. We would also like to thank Andrea Huber for facilitating all organisational aspects of this collaboration.

Author Contributions

Oliver Groth developed and implemented the SelMo model, conducted the curiosity experiments, studied related work on self-motivated reinforcement learning and wrote the paper. **Markus Wulfmeier** advised during the implementation and experimentation phase and helped writing the paper. **Giulia Vezzani** conducted the RHPO experiments and created the experiment evaluation plots. **Vibhavari Dasagi** assisted in compiling related work and provided feedback on the method and discussion section. **Tim Hertweck** provided technical support during the implementation and experimentation phase and created the supplementary videos. **Roland Hafner** conducted the SAC-X baseline experiments and provided support in setting up the JACO and OP3 environments. **Nicolas Heess** provided feedback on the experimental results and during writing and revision of the paper. **Martin Riedmiller** conceived the idea of self-motivated skill learning, provided conceptual feedback and supervised the project.

Appendix

7.A Simulation Environments

In this section we provide details about the two robotic simulation domains used in our experiments.

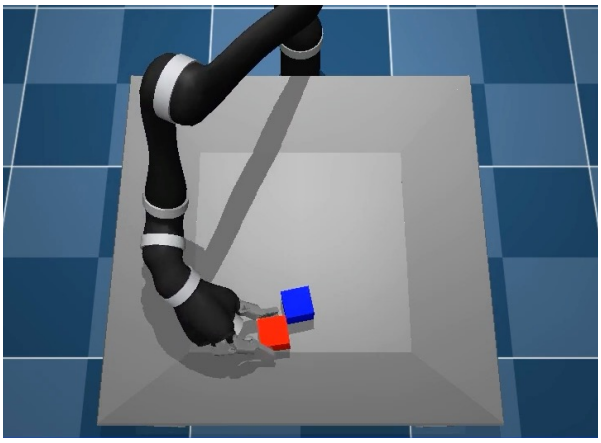


Figure 7.A.1: The JACO manipulation environment. The 6 DoF robot arm with a 3 DoF gripper can interact with multiple same-sized cubes in its workspace.

7.A.1 JACO Manipulation Environment

This environment is designed to study manipulation tasks like object lifting and stacking with a robotic arm (cf. Fig. 7.A.1). The state observation space consists of 72 dimensions: 24 features are used to represent the robot's proprioception as well as the state of each object. The action space spans 9 dimensions: 6 concerning the arm and 3 concerning the three-point gripper. A detailed description of the environment is provided in Table 7.A.1. Interactions with the two objects (`01` = red cube, `02` = blue cube) are evaluated using the sparse reward functions `reach_{red,blue}` and

Feature	Dimension
`arm/joints_pos`	6
`arm/joints_vel`	6
`arm/hand/finger_joints_pos`	3
`arm/hand/finger_joints_vel`	3
`arm/hand/fingertip_sensors`	3
`arm/hand/pinch_site_pos`	3
proprioception	$\sum = 24$
`O<i>/rel_pos_wrt_tcp`	3
`O<i>/pos`	3
`O<i>/orientation`	4
`O<i>/linvel`	3
`O<i>/angvel`	3
`O<i>/proptype`	5
`O<i>/dimensions`	3
perception per object <i>	$\sum = 24$
`arm`	6
`hand`	3
action space	$\sum = 9$

Table 7.A.1: State and action space semantics of JACO environment.

`lift_{red,blue}`. Each episode in this environment lasts 20 seconds or 400 control timesteps.

7.A.2 OP3 Locomotion Environment

This environment is designed to study locomotion with a humanoid robot (cf. Fig. 7.A.2). The state observation consists of 49 proprioceptive features. The 20-dimensional action space controls the orientations of the robot's head, ankle, elbow, hip, knee and shoulder. Actions passed to the robot are smoothed with an exponential filter to reduce motion jerk. A detailed description of the environment is provided in Table 7.A.2. The robot always spawns in a standing, upright position. Locomotion is evaluated by the dense reward functions `walk_{forward,backward}` for forward and backward walking gaits respectively. Each episode in this environment lasts 10 seconds or 200 control timesteps. If the robot's hip angle deviates more than 15° from an upright orientation, the simulation terminates early effectively preventing

the robot from falling over.

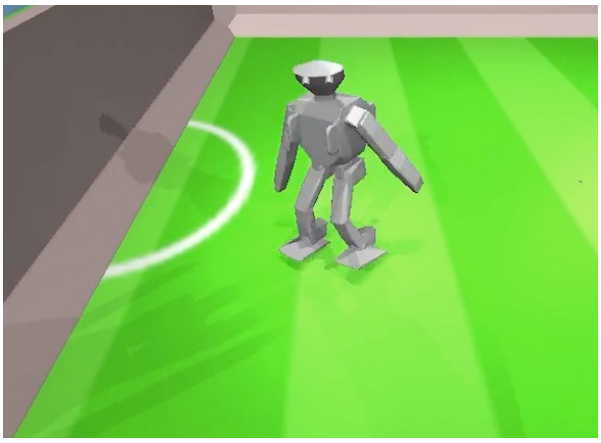


Figure 7.A.2: The OP3 locomotion environment. The 20 DoF humanoid robot can walk around on a plane. The episode terminates early, if the OP3 is about to fall over.

Feature	Dimension
`walker/joints/pos`	20
`walker/imu/linear_acc`	3
`walker/imu/angular_vel`	3
`walker/imu/gravity`	3
`scaled/action_filter/state`	20
proprioception	$\sum = 49$
`head_{pan,tilt}`	2
`{l,r}_ankle_{pitch,roll}`	4
`{l,r}_elbow`	2
`{l,r}_hip_{pitch,roll,yaw}`	6
`{l,r}_knee`	2
`{l,r}_shoulder_{pitch,roll}`	4
action space	$\sum = 20$

Table 7.A.2: State and action space semantics of OP3 environment.

7.B Model and Training Details

Across all experiments with the SelMo architecture, we consistently use the following hyperparameters and model architectures. Each SelMo experiment is run with a single actor for N = 1e5 episodes.

World model f_{dyn}**.** The world model is implemented as a *multi-layer perceptron* (MLP) with the following layer sizes and activation functions: `[FC(256), elu(·), FC(256), elu(·), FC(size_state)]` For each environment, `size_state` is the sum of the dimensions for proprioception and perception (cf. Section 7.A). The world model is optimised using Adam (Kingma and Ba [103]) with a learning rate of $\eta_{\mathcal{M}} = 3e-4$.

Policy. Both policy π and critic Q are implemented as two independent MLPs with the following layer sizes and activation functions:

- Q: `[tanh(·), FC(512), elu(·), FC(512), elu(·), FC(256), FC(1)]`
- π: `[FC(256), elu(·), FC(256), elu(·), FC(128), FC(size_action)]`

The `size_action` is different for each environment (cf. Section 7.A). The policy is optimised using Adam (Kingma and Ba [103]) with a learning rate of $\eta_{\pi} = 3e-4$. The reward scale is set to $\eta_{r} = 10.0$ across all experiments.

Replays. The replays $D_{\mathcal{M}}$ and D_{π} store trajectories with a length of $T = 50$ transitions. The buffer sizes used are $|D_{\mathcal{M}}| = |D_{\pi}| = 5e4$ and each trajectory in the buffers can be sampled at most $n_{max} = m_{max} = 32$ times (cf. Section 7.3). The batch size of samples drawn from the replays is set to $B = 64$.

8 Discussion

The investigation of physical intuition has been an established field of research in Cognitive Science for many decades and has been identified as an important pillar of human learning and cognition (McCloskey [135]). In contrast, its adoption in Robotics and Machine Learning is relatively nascent, facilitated in large part by the advent of deep learning models. In the previous chapters of this thesis, we have implemented different aspects of physical intuition and investigated their conduciveness in robotic manipulation scenarios. In this final chapter we revisit our central hypothesis postulated in Chapter 1 and evaluate how it is corroborated by the results obtained in the experiments conducted. We also reflect on the limitations of the models proposed here and point out avenues of future work to improve upon the identified shortcomings.

8.1 Key Contributions

In Chapter 1 we proposed that a 'tight integration of intuitive physics models with robotics control renders possible sophisticated manipulation skills'. As we have discussed in Chapter 2, the task of scene rearrangement (Batra et al. [12]) provides an ideal testbed for this research as it encompasses planning, visuomotor control and

representation learning aspects. In Sections 3.5.2 and 3.5.3 we demonstrate that an intuitive visual stability classifier facilitates the planning of stack construction as it can be employed to infer a suitable ranking of objects according to the stable support they provide and identify placement locations for counterweights. Extending the work on rearrangement planning with physical intuition, we show in Section 5.4 that our object-centric, generative model is capable of visualising physically plausible scene states and enables the targeted editing of object locations or the forward prediction of the scene's dynamics. This makes the model useful for planning object position changes in a scene in advance as well as providing a visualisation of the altered scene which could serve as an imagined target image for a downstream visuomotor controller. Both contributions corroborate the conduciveness of physical intuition – especially the aspect of physically plausible imagination – to planning in robotic manipulation.

The second area of robotic manipulation that this thesis contributes to is visuomotor control (VMC). As shown in the stacking experiment in Section 3.5.2, our data-driven model for visual stability prediction can be successfully employed in a closed control loop to guide the placement of an object to a stable stacking position. In Section 6.3 we also explore how a deeper integration of visual dynamics approximation with the VMC architecture affects the performance in manipulation tasks. The corresponding experiments in Section 6.4 confirm that the utilisation of dynamic images facilitates the goal conditioning of the VMC policy network while also making it significantly more resistant against visual perturbations during execution. In summary, the VMC experiments in Chapters 3 and 6 suggest that high- and low-level approximations of rigid-body motions can successfully be integrated into closed control loops to obtain robust manipulation policies from raw visual input without the need for exact measurements of the scene.

Orthogonal to our particular contributions to rearrangement planning and VMC, our experiments also provide some insight into the aspect of representation learning for

manipulation tasks[1]. The tool imagination task studied in Section 4.4 gives a fresh perspective on the representation of a reaching affordance: Instead of treating the affordance as a label to predict, it is represented as a latent space manifold which supports a classifier predicting the feasibility of a given tool for a given reaching task. In a similar vein, our experiments with RELATE in Sections 5.A.1 and 5.4.1 shed additional light on the formation of a disentangled latent space in connection to physical relationship modelling.

The technical contributions of Chapters 3 to 6 corroborate the claim about the conduciveness of learned physical intuitions to various aspects of robotic manipulation. However, our central hypothesis posits that a 'tight integration' of intuition and control renders possible sophisticated manipulation. Hence, we combine both intuition and control modules into a single system in Chapter 7 which is jointly trained from scratch. The results presented in Section 7.4.1 ultimately demonstrate that complex embodied behaviour can emerge as a natural byproduct when learning a forward-predictive model about the environment's dynamics with curious exploration. In particular, it is worth noting that this employment of physical intuition goes beyond the established scope of applying a learned intuition in a control loop to facilitate a certain task (e. g. stacking) and demonstrates that the acquisition of the intuition itself shapes control policies which naturally solve tasks without any prior definition. In addition, this finding connects the acquisition of physical intuition to the wider problem of structured exploration in Machine Learning and Robotics. Finally, it opens the research avenue of employing the acquisition of physical intuition as a potential driver for continual robot learning and unsupervised skill discovery.

8.2 Limitations and Future Work

In the previous section, we have reiterated on the contributions made in this thesis and evaluated them with respect to our central hypothesis postulated in Chapter 1. We believe that our experiments provide empirical evidence to support the hypothesis that

[1] A more comprehensive analysis on representation learning for robotic manipulation can be found at Wulfmeier et al. [215].

physical intuition enables sophisticated object manipulation by facilitating two of its core problems: planning and visuomotor control. In addition, we have demonstrated that the joint learning of a physics approximator and a control policy yields complex, emerging behaviours in embodied agents further emphasising that physical intuition and robotic control need to be tightly integrated. However, the benefits of physical intuition presented in this thesis need to be put into perspective with its limitations to specify the scope of its application and identify avenues of future work.

Firstly, it is important to note that machine-learned physical intuition – regardless of its concrete instantiation as e. g. a visual stability predictor or a dynamics model – is always but an approximation of the true underlying physical phenomenon. Hence, when implemented using contemporary deep learning methodology, it is prone to the common drawbacks of these techniques. Most importantly, the models will only exploit correlations in the observed training data which do not necessarily correspond to the true physical causations. For instance, in Section 3.4.2 we observe that in 80% of the instable stacks, the attention peak of the visual stability classifier rests on the object violating the stability constraint. However, the CNN does not invoke any reasoning about the true centres of masses or their respective support in those cases but rather relies on visually discernible cues such as bent surfaces or protruding objects which are discriminative visual features of instability in *most* cases.

Secondly, machine-learned physical intuitions typically exhibit limited generalisation beyond their training domains if this aspect has not been explicitly mitigated during the model design. For instance, our model for tool imagination in Chapter 4 is able to generate 'novel' tool shapes for unseen tasks but only in so far as those shapes are interpolations within the learned factors of variation during training, i. e. width, handle length and hook length. While this already affords generalisation to the space of tools induced by the learned factors of variation, it is still unable to go beyond – e. g. imagining a tool with multiple hooks – without any prior example. RELATE (cf. Section 5.3) is a step up in this direction as it is able to generalise its dynamics modelling to different numbers of objects because the spatial relationship

approximation is implemented in a symmetric and permutation-invariant way. In a similar vein, the task transfer capabilities of GEECO (cf. Section 6.3) are afforded by the explicit use of dynamic images which reduces the model's dependency on visual textures and focuses the controller on motion features instead which are invariant between different manipulation tasks. In summary, these examples highlight the importance of the underlying model assumptions, even in the case when physical phenomena are learned in a data-driven way.

Lastly, a common drawback of the approaches presented in Chapters 3, 4 and 6 is their dependency on task demonstrations in the form of built stacks, feasible tool examples or successful object rearrangements. While such demonstrations are easily obtainable in procedurally generated simulations, the transfer of simulation-trained models to real world scenarios – also known as the *sim2real problem* – requires additional effort and scrutiny. Especially for the applications investigated here, it is important to bear in mind that no contemporary physics simulator captures real physics without any errors. Therefore, sim2real domain adaptation and efficient real world data collection remain crucial tasks when deploying any of these approaches onto a real robotic platform.

Reflecting upon the current limitations of the intuitive physics models investigated here, multiple avenues of future work can be identified to improve on the data-driven learning of physical dynamics and facilitate the adoption of physical intuition in robotics. To mitigate the problems related to correlation exploitation and limited generality, we believe that the modelling of approximate physics requires architectural innovations beyond established deep learning architectures. Our results in Sections 3.4.1 and 5.4.3 suggest that CNNs and MLPs are sufficient to approximate certain aspects of physical intuition like structural stability or rigid-body trajectories, respectively. However, in order to increase the generality and robustness of the models obtained their latent spaces need to be equipped with inductive biases which facilitate generalisation such as (disentangled) factors of variation (cf. Section 4.3) or set-like representations (cf. Section 5.3). In that regard, the lines of work on object-centric

scene representation (e. g. Burgess et al. [25]) and graph networks for physical interaction (e. g. P. Battaglia et al. [14]) are very promising avenues of research whose results are likely to feed back into improvements of intuitive physics models.

Reducing the amount of demonstrations and supervision needed for the training of intuitive physics models is another thread which could follow from work in this thesis. The curiosity-based method presented in Chapter 7 explores how an approximation of the environment dynamics can be learned jointly with a control policy. This opens up an avenue of research to learn entire world models (Ha and Schmidhuber [69]) in an unsupervised way using structured exploration. Interestingly, the curiosity-driven acquisition of world models also connects to the thread of structured models for physics approximation: Instead of using a simple MLP to approximate the next state vector as in Section 7.3, more sophisticated, object-centric dynamics approximators like (visual) interaction networks (P. Battaglia et al. [14], Van Steenkiste et al. [196], and Watters et al. [204]) could be leveraged. This would enable more faithful predictions multiple steps into the future which facilitates more data-efficient structured exploration as in Sekar et al. [174] and also connects the model learning to raw visual input. In addition to their use in planning or MPC, world models which capture a physical intuition about the environment's dynamics also unlock the tantalising possibility of training entire policies within a model's 'imagination' as shown by Ha and Schmidhuber [69] and Schrittwieser et al. [173].

Beyond the research threads which address the immediate limitations of the approaches presented here, other avenues of future work could follow more open-ended questions arising from work in this thesis. Of particular importance in that regard would be methods for an automatic identification and retention of novel and useful behaviour as discussed in Section 7.5. As our experiments in Section 7.4.1 have shown, complex embodied behaviour emerges as a 'byproduct' of learning a model about the environment's dynamics. However, the continual learning setup and the constant overwriting of old behaviours with new ones in the policy network still pose

a significant challenge. If this challenge could be overcome though, the acquisition of physical intuition could be leveraged as an important driver for learning intricate behaviour for which the design of targeted reward functions is impractical.

8.3 Conclusion

Inspired by the concept of physical intuition in Cognitive Sciene and its alleged influence on our embodied behaviour and object manipulation capabilities, we have set out in this thesis to investigate how machine-learned physical intuition could be leveraged to improve object manipulation in Robotics. We have posited the hypothesis that a tight integration between physical intuition – i. e. a predictive capability about the evolution of the environment over time – and robotic control can unlock sophisticated manipulation skills in artificial agents. We started our investigation in Chapters 3 and 4 with two experimental setups borrowed from the Cognitive Science literature: visual stability prediction and tool manufacturing in a reaching task. In both cases we demonstrated how high-level intuition classifiers can be leveraged to stably stack objects or synthesise appropriate tools. Next, in Chapters 5 and 6 we looked into low-level physical intuition, i. e. the modelling and approximation of rigid-body motions from visual input. We show that these models facilitate crucial aspects of object manipulation such as scene imagination and goal-conditioned visuomotor control. Finally, we integrated the dynamics model and the control policy into a joint learning setup in Chapter 7. Our final experiments demonstrate that the curiosity-driven exploration of an environment's dynamics leads to the emergence of complex skills in manipulation and locomotion which ultimately emphasises the close link between the acquisition of physical intuition and the quality of embodied behaviour.

The results in Chapter 7 also provide an interesting spin on the original motivation of this thesis: While the earlier experiments in Chapters 3 and 6 investigated how physical intuition can be leveraged to enhance object manipulation capabilities, the results of Chapter 7 suggest that sophisticated manipulation skills could also be

explained as a 'byproduct' of learning physical intuition. Additionally, this finding also sheds some light on questions raised during the discussion of physical intuition in Cognitive Science in Chapter 2. If the assumption holds that our embodied skills result at least partially from the acquisition of physical intuition, it could explain that our own physical intuition is only developed to the point which allowed for the emergence of 'useful' skills. And by the time our set of skills is useful enough to competently deal with everyday situations, the learning process slows down leaving our understanding of physics at a useful but imperfect state.

In conclusion, we believe that the investigations in this thesis contribute to the transfer of Cognitive Science concepts into the field Robotics and provide insightful and novel perspectives on the utilisation of approximate physical models in the learning process of object manipulation skills. Our findings could potentially inspire future areas of Robotics research, especially with respect to world modelling, structured exploration and continual learning. Finally, our research into the nature and utility of physical intuition can be summarised elegantly by quoting one of the central tenents of sciene in a slightly adapted form: 'All models are wrong. But the acquisition of some models is useful.'

References

[1] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller. "Maximum a Posteriori Policy Optimisation". In: *arXiv preprint arXiv:1806.06920* (2018).

[2] J. Achiam and S. Sastry. "Surprise-based Intrinsic Motivation for Deep Reinforcement Learning". In: *arXiv preprint arXiv:1703.01732* (2017).

[3] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine. "Learning to Poke by Poking: Experiential Learning of Intuitive Physics". In: *Advances in Neural Information Processing Systems* NIPS (2016). ISSN: 10495258. arXiv: 1606.07419. URL: http://arxiv.org/abs/1606.07419.

[4] K. R. Allen, K. A. Smith, and J. B. Tenenbaum. "Rapid Trial-and-error Learning with Simulation Supports Flexible Tool Use and Physical Reasoning". In: *Proceedings of the National Academy of Sciences* 117.47 (2020), pp. 29302–29310.

[5] S. H. Ambrose. "Paleolithic Technology and Human Evolution". In: *Science* 291.5509 (2001), pp. 1748–1753.

[6] T. Anciukevicius, C. H. Lampert, and P. Henderson. "Object-Centric Image Generation with Factored Depths, Locations, and Appearances". In: *arXiv preprint arXiv:2004.00642* (2020).

[7] Aristotle. *Physics*. Trans. by D. W. Graham. Hackett Publishing Company, 1999.

[8] R. Baillargeon. "The Acquisition of Physical Knowledge in Infancy: A Summary in Eight Lessons". In: *Blackwell Handbook of Childhood Cognitive Development* 1.46-83 (2002), p. 1.

[9] R. Baillargeon, E. S. Spelke, and S. Wasserman. "Object Permanence in Five-month-old Infants". In: *Cognition* 20.3 (1985), pp. 191–208.

[10] A. Bakhtin, L. van der Maaten, J. Johnson, L. Gustafson, and R. Girshick. "Phyre: A New Benchmark for Physical Reasoning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5083–5094.

[11] A. Baranes and P.-Y. Oudeyer. "R-iac: Robust Intrinsically Motivated Exploration and Active Learning". In: *IEEE Transactions on Autonomous Mental Development* 1.3 (2009), pp. 155–169.

[12] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, et al. "Rearrangement: A Challenge for Embodied AI". In: *arXiv preprint arXiv:2011.01975* (2020).

[13] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum. "Simulation as an Engine of Physical Scene Understanding". In: *Proceedings of the National Academy of Sciences* 110.45 (2013), pp. 18327–18332. ISSN: 0027-8424. DOI: 10.1073/pnas.1306572110. arXiv: arXiv:1404.2263v1. URL: http://www.pnas.org/cgi/doi/10.1073/pnas.1306572110.

[14] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. "Interaction Networks for Learning about Objects, Relations and Physics". In: *Advances in Neural Information Processing Systems*. 2016, pp. 4502–4510.

[15] S. R. Beck, I. A. Apperly, J. Chappell, C. Guthrie, and N. Cutting. "Making Tools isn't Child's Play". In: *Cognition* 119.2 (2011), pp. 301–306.

[16] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. "Unifying Count-based Exploration and Intrinsic Motivation". In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 1471–1479.

[17] Y. Bengio, A. Courville, and P. Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828.

[18] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi. "Action Recognition with Dynamic Image Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.12 (2017), pp. 2799–2813.

[19] J. Bohg, A. Morales, T. Asfour, and D. Kragic. "Data-driven Grasp Synthesis – a Survey". In: *IEEE Transactions on Robotics* 30.2 (2013), pp. 289–309.

[20] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. "Learning 6D Object Pose Estimation Using 3D Object Coordinates". In: *European conference on computer vision*. Springer. 2014, pp. 536–551.

[21] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. *OpenAI Gym*. 2016. eprint: arXiv:1606.01540.

[22] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. "Language Models are Few-Shot Learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[23] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. "Large-scale Study of Curiosity-driven Learning". In: *arXiv preprint arXiv:1808.04355* (2018).

[24] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. "Exploration by Random Network Distillation". In: *arXiv preprint arXiv:1810.12894* (2018).

[25] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. "MONet: Unsupervised Scene Decomposition and Representation". In: *arXiv preprint arXiv:1901.11390* (2019).

[26] A. Byravan, F. Lceb, F. Meier, and D. Fox. "Se3-Pose-Nets: Structured Deep Dynamics Models for Visuomotor Control". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–8.

[27] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum. "A Compositional Object-Based Approach to Learning Physical Dynamics". In: (2017).

[28] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning Phrase Representations Using RNN Encoder-decoder for Statistical Machine Translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[29] A. Collet, M. Martinez, and S. S. Srinivasa. "The MOPED Framework: Object Recognition and Pose Estimation for Manipulation". In: *The International Journal of Robotics Research* 30.10 (2011), pp. 1284–1306.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A Large-Scale Hierarchical Image Database". In: *IEEE Conference on Computer Vision and Pattern Recognition, 2009*. IEEE. 2009, pp. 248–255.

[31] E. L. Denton and v. Birodkar. "Unsupervised Learning of Disentangled Representations from Video". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 4414–4423. URL: http://papers.nips.cc/paper/7028-unsupervised-learning-of-disentangled-representations-from-video.pdf.

[32] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp. "Goal-Conditioned Imitation Learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 15324–15335.

[33] T. Do, A. Nguyen, I. D. Reid, D. G. Caldwell, and N. G. Tsagarakis. "AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018.

[34] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba. "One-Shot Imitation Learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1087–1098.

[35] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine. "Visual Foresight: Model-based Deep Reinforcement Learning for Vision-based Robotic Control". In: *arXiv preprint arXiv:1812.00568* (2018).

[36] F. Ebert, C. Finn, A. X. Lee, and S. Levine. "Self-Supervised Visual Planning with Temporal Skip Connections". In: *Conference on Robot Learning*. 2017.

[37] S. Ehrhardt, A. Monszpart, N. J. Mitra, and A. Vedaldi. "Taking Visual Motion Prediction to New Heightfields". In: *Computer Vision and Image Understanding* 181 (2019), pp. 14–25.

[38] S. Ehrhardt, O. Groth, A. Monszpart, M. Engelcke, I. Posner, N. J. Mitra, and A. Vedaldi. "RELATE: Physically Plausible Multi-Object Scene Synthesis Using Structured Latent Spaces". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Dec. 2020.

[39] N. J. Emery and N. S. Clayton. "Tool Use and Physical Cognition in Birds and Mammals". In: *Current Opinion in Neurobiology* 19.1 (2009), pp. 27–33.

[40] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner. "Genesis: Generative Scene Inference and Sampling with Object-centric Latent Representations". In: *International Conference on Learning Representations*. 2020.

[41] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. *Visualizing Higher-Layer Features of a Deep Network*. Tech. rep. University of Montreal, 2009.

[42] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. "Diversity Is All You Need: Learning Skills without a Reward Function". In: *arXiv preprint arXiv:1802.06070* (2018).

[43] B. Fernando, E. Gavves, M. José Oramas, A. Ghodrati, and T. Tuytelaars. "Modeling Video Evolution for Action Recognition". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 5378–5387. DOI: 10.1109/CVPR.2015.7299176.

[44] C. Finn, P. Abbeel, and S. Levine. "Model-agnostic Meta-learning for Fast Adaptation of Deep Networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1126–1135.

[45] C. Finn, I. Goodfellow, and S. Levine. "Unsupervised Learning for Physical Interaction Through Video Prediction". In: *Advances in Neural Information Processing Systems*. 2016, pp. 64–72.

[46] C. Finn and S. Levine. "Deep Visual Foresight for Planning Robot Motion". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2786–2793.

[47] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. "One-Shot Visual Imitation Learning via Meta-Learning". In: *Conference on Robot Learning*. 2017, pp. 357–368.

[48] J. Fischer, J. G. Mikhael, J. B. Tenenbaum, and N. Kanwisher. "Functional Neuroanatomy of Intuitive Physical Inference". In: *Proceedings of the National Academy of Sciences* 113.34 (2016), E5072–E5081. ISSN: 0027-8424. DOI: 10.1073/pnas.1610344113. URL: http://www.pnas.org/lookup/doi/10.1073/pnas.1610344113.

[49] J. W. Forrester. "Counterintuitive Behavior of Social Systems". In: *Theory and Decision* 2.2 (1971), pp. 109–140.

[50] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik. "Learning Visual Predictive Models of Physics for Playing Billiards". In: *International Conference on Learning Representations*. 2016.

[51] F. B. Fuchs, O. Groth, A. R. Kosiorek, A. Bewley, M. Wulfmeier, A. Vedaldi, and I. Posner. "Scrutinizing and De-Biasing Intuitive Physics with Neural Stethoscopes." In: *British Machine Vision Conference (BMVC)*. Sept. 2019.

[52] F. B. Fuchs, A. R. Kosiorek, L. Sun, O. P. Jones, and I. Posner. "End-to-end Recurrent Multi-Object Tracking and Trajectory Prediction with Relational Reasoning". In: *Sets and Partitions Workshop at NeurIPS 2019* (2019).

[53] F. Furrer, M. Wermelinger, H. Yoshida, F. Gramazio, M. Kohler, R. Siegwart, and M. Hutter. "Autonomous Robotic Stone Stacking with Online Next Best Object Target Pose Planning". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2017), pp. 2350–2356. ISSN: 10504729. DOI: 10.1109/ICRA.2017.7989272.

[54] J. J. Gibson. "The Theory of Affordances". In: *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*. Hilldale, USA: Lawrence Erlbaum, 1977.

[55] R. Girdhar, D. Fouhey, M. Rodriguez, and A. Gupta. "Learning a Predictable and Generative Vector Representation for Objects". In: *ECCV*. 2016.

[56] R. C. Goertz. "Fundamentals of General-purpose Remote Manipulators". In: *Nucleonics* 10.11 (1952), pp. 36–42.

[57] R. C. Goertz and W. M. Thompson. "Electronically Controlled Manipulator". In: *Nucleonics (US) Ceased publication* 12 (1954).

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

[59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[60] U. E. Goswami. *Blackwell Handbook of Childhood Cognitive Development*. Blackwell Publishing, 2002.

[61] H. Grabner, J. Gall, and L. Van Gool. "What Makes a Chair a Chair?" In: *Computer Vision and Pattern Recognition (CVPR)*. 2011, pp. 1529–1536.

[62] K. Greff, R. L. Kaufmann, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. "Multi-object Representation Learning with Iterative Variational Inference". In: *Proceedings of the 36th International Conference on Machine Learning*. 2019.

[63] K. Gregor, D. J. Rezende, and D. Wierstra. "Variational Intrinsic Control". In: *arXiv preprint arXiv:1611.07507* (2016).

[64] O. Groth, F. B. Fuchs, I. Posner, and A. Vedaldi. "ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking". In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018.

[65] O. Groth, C.-M. Hung, A. Vedaldi, and I. Posner. "Goal-Conditioned End-to-End Visuomotor Control for Versatile Skill Primitives". In: *IEEE International Conference on Robotics and Automation (ICRA)*. June 2021.

[66] O. Groth, M. Wulfmeier, G. Vezzani, V. Dasagi, T. Hertweck, R. Hafner, N. Heess, and M. Riedmiller. "Is Curiosity All You Need? On the Utility of Emergent Behaviours from Curious Exploration". In: *arXiv preprint arXiv:2109.08603* (Sept. 2021).

[67] A. Gupta, A. A. Efros, and M. Hebert. "Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics". In: *European Conference on Computer Vision (ECCV)*. 2010.

[68] D. Ha and J. Schmidhuber. "Recurrent World Models Facilitate Policy Evolution". In: *arXiv preprint arXiv:1809.01999* (2018).

[69] D. Ha and J. Schmidhuber. "World Models". In: *arXiv preprint arXiv:1803.10122* (2018).

[70] N. Haber, D. Mrowca, S. Wang, F.-F. Li, and D. L. Yamins. "Learning to Play With Intrinsically-Motivated, Self-Aware Agents". In: *NeurIPS*. 2018, pp. 8398–8409.

[71] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu. "Embracing Change: Continual Learning in Deep Neural Networks". In: *Trends in Cognitive Sciences* (2020).

[72] J. Hamrick, P. Battaglia, and J. B. Tenenbaum. "Internal Physics Models Guide Probabilistic Judgments about Object Dynamics". In: *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*. Vol. 2. Citeseer. 2011.

[73] J. B. Hamrick. "Analogues of Mental Simulation and Imagination in Deep Learning". In: *Current Opinion in Behavioral Sciences* 29 (2019), pp. 8–16.

[74] J. B. Hamrick, P. W. Battaglia, T. L. Griffiths, and J. B. Tenenbaum. "Inferring Mass in Complex Scenes by Mental Simulation". In: *Cognition* 157 (2016). ISSN: 18737838. DOI: 10.1016/j.cognition.2016.08.012.

[75] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[76] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger, et al. "Team Delfts Robot Winner of the Amazon Picking Challenge 2016". In: *Robot World Cup*. Springer. 2016, pp. 613–624.

[77] T. Hertweck, M. Riedmiller, M. Bloesch, J. T. Springenberg, N. Siegel, M. Wulfmeier, R. Hafner, and N. Heess. "Simple Sensor Intentions for Exploration". In: *arXiv preprint arXiv:2005.07541* (2020).

[78] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. "GANs Trained by a two Time-scale Update Rule Converge to a Local Nash Equilibrium". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6626–6637.

[79] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *ICLR*. 2017.

[80] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes". In: *2011 international conference on computer vision*. IEEE. 2011, pp. 858–865.

[81] G. Hinton, N. Srivastava, and K. Swersky. "Coursera, Neural Networks for Machine Learning, Lecture 6e". In: (2014). URL: https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf.

[82] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. "VIME: Variational Information Maximizing Exploration". In: *arXiv preprint arXiv:1605.09674* (2016).

[83] D.-A. Huang, S. Nair, D. Xu, Y. Zhu, A. Garg, L. Fei-Fei, S. Savarese, and J. C. Niebles. "Neural Task Graphs: Generalizing to Unseen Tasks from a Single Video Demonstration". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8565–8574.

[84] S. H. Huang, M. Zambelli, J. Kay, M. F. Martins, Y. Tassa, P. M. Pilarski, and R. Hadsell. "Learning Gentle Object Manipulation with Curiosity-driven Deep Reinforcement Learning". In: *arXiv preprint arXiv:1903.08542* (2019).

[85] X. Huang and S. Belongie. "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1501–1510.

[86] S. James, M. Bloesch, and A. J. Davison. "Task-Embedded Control Networks for Few-Shot Imitation Learning". In: *Conference on Robot Learning*. 2018, pp. 783–795.

[87] S. James and A. Davison. "Attention-driven Robotic Manipulation". In: (2020).

[88] S. James, A. J. Davison, and E. Johns. "Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task". In: *Conference on Robot Learning*. 2017, pp. 334–343.

[89] S. L. James. "Tightly-coupled Manipulation Pipelines: Combining Traditional Pipelines and End-to-end Learning". PhD thesis. Imperial College London, 2021.

[90] M. Janner, S. Levine, W. T. Freeman, J. B. Tenenbaum, C. Finn, and J. Wu. "Reasoning About Physical Interactions with Object-Oriented Prediction and Planning". In: *International Conference on Learning Representations*. 2019.

[91] Z. Jia, A. C. Gallagher, A. Saxena, and T. Chen. "3D Reasoning from Blocks to Stability". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.5 (2015), pp. 905–918. ISSN: 01628828. DOI: 10.1109/TPAMI.2014.2359435.

[92] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning". In: (2017), pp. 2901–2910.

[93] L. P. Kaelbling. "Learning to Achieve Goals". In: *IJCAI*. Citeseer. 1993, pp. 1094–1099.

[94] D. Kahneman. *Thinking, Fast and Slow*. Macmillan, 2011.

[95] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. "Video Pixel Networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 1771–1779.

[96] H. Kato, Y. Ushiku, and T. Harada. "Neural 3D Mesh Renderer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3907–3916.

[97] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. "Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces". In: *IEEE transactions on Robotics and Automation* 12.4 (1996), pp. 566–580.

[98] I. K. Kim and E. S. Spelke. "Infants' Sensitivity to Effects of Gravity on Visible Object Motion." In: *Journal of Experimental Psychology: Human Perception and Performance* 18.2 (1992), p. 385.

[99] S. Kimura, T. Watanabe, and Y. Aiyama. "Force based Manipulation of Jenga Blocks". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, pp. 4287–4292.

[100] J. E. King, M. Cognetti, and S. S. Srinivasa. "Rearrangement Planning Using Object-centric and Robot-centric Action Spaces". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 3940–3947.

[101] D. P. Kingma and M. Welling. "Auto-encoding Variational Bayes". In: *International Conference on Learning Representations*. 2014.

[102] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (2015).

[103] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[104] H. Kjellström, J. Romero, and D. Kragić. "Visual Object-Action Recognition: Inferring Object Affordances from Human Demonstration". In: *Computer Vision and Image Understanding* 115.1 (2011), pp. 81–90. ISSN: 10773142. DOI: 10.1016/j.cviu.2010.08.002.

[105] A. S. Klyubin, D. Polani, and C. L. Nehaniv. "Empowerment: A Universal Agent-centric Measure of Control". In: *2005 IEEE Congress on Evolutionary Computation*. Vol. 1. IEEE. 2005, pp. 128–135.

[106] N. Kodali, J. Abernethy, J. Hays, and Z. Kira. "On Convergence and Stability of GANs". In: *arXiv preprint arXiv:1705.07215* (2017).

[107] A. Kohli, V. Sitzmann, and G. Wetzstein. "Inferring Semantic Information with 3D Neural Scene Representations". In: *arXiv preprint arXiv:2003.12673* (2020).

[108] H. S. Koppula and A. Saxena. "Anticipating Human Activities Using Object Affordances for Reactive Robotic Response". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (2016), pp. 14–29. ISSN: 01628828. DOI: 10.1109/TPAMI.2015.2430335.

[109] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner. "Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects". In: *Advances in Neural Information Processing Systems*. 2018, pp. 8606–8616.

[110] J. Kossen, K. Stelzner, M. Hussing, C. Voelcker, and K. Kersting. "Structured Object-Aware Physics Prediction for Video Modeling and Planning". In: *International Conference on Learning Representations*. 2020.

[111] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances In Neural Information Processing Systems* (2012), pp. 1–9. ISSN: 10495258. DOI: http://dx.doi.org/10.1016/j.protcy.2014.09.007. arXiv: 1102.0183.

[112] J. R. Kubricht, K. J. Holyoak, and H. Lu. "Intuitive Physics: Current Research and Controversies". In: *Trends in Cognitive Sciences* 21.10 (2017), pp. 749–759. ISSN: 1879307X. DOI: 10.1016/j.tics.2017.06.002. URL: http://dx.doi.org/10.1016/j.tics.2017.06.002.

[113] T. D. Kulkarni, P. Kohli, J. B. Tenenbaum, and V. Mansinghka. "Picture: A Probabilistic Programming Language for Scene Perception". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4390–4399.

[114] Y. Labbé, S. Zagoruyko, I. Kalevatykh, I. Laptev, J. Carpentier, M. Aubry, and J. Sivic. "Monte-Carlo Tree Search for Efficient Visually Guided Rearrangement Planning". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3715–3722.

[115] Y. LeCun. "The MNIST Database of Handwritten Digits". In: *http://yann.lecun.com/exdb/mnist/* (1998). URL: https://ci.nii.ac.jp/naid/10027939599/en/.

[116] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. "Stochastic Adversarial Video Prediction". In: *arXiv preprint arXiv:1804.01523* (2018).

[117] A. Lerer, S. Gross, and R. Fergus. "Learning Physical Intuition of Block Towers by Example". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 430–438. URL: http://dl.acm.org/citation.cfm?id=3045390.3045437.

[118] S. Levine, C. Finn, T. Darrell, and P. Abbeel. "End-to-end Training of Deep Visuomotor Policies". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1334–1373.

[119] S. Levine and V. Koltun. "Guided Policy Search". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1–9.

[120] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. "Learning Hand-eye Coordination for Robotic Grasping with Deep Learning and Large-scale Data Collection". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 421–436.

[121] W. Li, S. Azimi, A. Leonardis, and M. Fritz. "To Fall or Not to Fall: A Visual Approach to Physical Stability Prediction". In: *arXiv preprint arXiv:1604.00066* (2016).

[122] W. Li, A. Leonardis, and M. Fritz. "Visual Stability Prediction for Robotic Manipulation". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2017), pp. 2606–2613. ISSN: 10504729. DOI: 10.1109/ICRA.2017.7989304.

[123] Y. Liao, K. Schwarz, L. Mescheder, and A. Geiger. "Towards Unsupervised Learning of Generative Models for 3D Controllable Image Synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5871–5880.

[124] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. "Continuous Control with Deep Reinforcement Learning". In: *arXiv preprint arXiv:1509.02971* (2015).

[125] D. Lisle. "Making safe: The dirty history of a bomb disposal robot". In: *Security dialogue* 51.2-3 (2020), pp. 174–193.

[126] S. Liu, T. Li, W. Chen, and H. Li. "Soft Rasterizer: A Differentiable Renderer for Image-Based 3D Reasoning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7708–7717.

[127] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. "Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1118–1125.

[128] Z. Liu, W. T. Freeman, J. B. Tenenbaum, and J. Wu. "Physical Primitive Decomposition". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 3–19.

[129] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem. "On the Fairness of Disentangled Representations". In: *Advances in Neural Information Processing Systems*. 2019, pp. 14611–14624.

[130] M. Lopes, T. Lang, M. Toussaint, and P.-Y. Oudeyer. "Exploration in Model-based Reinforcement Learning by Empirically Estimating Learning Progress". In: *Neural Information Processing Systems (NIPS)*. 2012.

[131] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch. "Plan Online, Learn Offline: Efficient Learning and Exploration via Model-based Control". In: *arXiv preprint arXiv:1811.01848* (2018).

[132] T. Lozano-Pérez and M. A. Wesley. "An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles". In: *Communications of the ACM* 22.10 (1979), pp. 560–570.

[133] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. "Learning Latent Plans from Play". In: *Conference on Robot Learning*. PMLR. 2020, pp. 1113–1132.

[134] T. Mar, V. Tikhanoff, G. Metta, and L. Natale. "Self-supervised Learning of Grasp Dependent Tool Affordances on the iCub Humanoid Robot". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 3200–3206.

[135] M. McCloskey. "Intuitive Physics". In: *Scientific American* 248.4 (1983), pp. 122–131.

[136] M. McCloskey and N. J. Cohen. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". In: *Psychology of Learning and Motivation*. Vol. 24. Elsevier, 1989, pp. 109–165.

[137] A. L. Mitchell, M. Engelcke, O. P. Jones, D. Surovik, S. Gangapurwala, O. Melon, I. Havoutis, and I. Posner. "First Steps: Latent-Space Control with Semantic Constraints for Quadruped Locomotion". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5343–5350.

[138] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. "Spectral Normalization for Generative Adversarial Networks". In: *International Conference on Learning Representations*. 2018.

[139] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. "Playing Atari with Deep Reinforcement Learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[140] S. Mohamed and D. J. Rezende. "Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning". In: *arXiv preprint arXiv:1509.08731* (2015).

[141] A. Mordvintsev, C. Olah, and M. Tyka. *Inceptionism: Going Deeper into Neural Networks*. 2015. URL: `https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`.

[142] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi. "Newtonian Image Understanding: Unfolding the Dynamics of Objects in Static Images". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.383. arXiv: 1511.04048.

[143] R. Mottaghi, M. Rastegari, A. Gupta, and A. Farhadi. ""What happens if..." Learning to Predict the Effect of Forces in Images". In: *European Conference on Computer Vision*. Springer. 2016, pp. 269–285.

[144] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. "Affordance Detection of Tool Parts from Geometric Features". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 1374–1381.

[145] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel. "Overcoming Exploration in Reinforcement Learning with Demonstrations". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 6292–6299.

[146] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine. "Visual Reinforcement Learning with Imagined Goals". In: *Advances in Neural Information Processing Systems*. 2018, pp. 9191–9200.

[147] S. Nair and C. Finn. "Hierarchical Foresight: Self-Supervised Learning of Long-Horizon Tasks via Visual Subgoal Generation". In: *arXiv preprint arXiv:1909.05829* (2019).

[148] I. Newton. *Philosophiae Naturalis Principia Mathematica*. Vol. 2. typis A. et JM Duncan, 1833.

[149] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang. "HoloGAN: Unsupervised Learning of 3D Representations from Natural Images". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7588–7597.

[150] T. Nguyen-Phuoc, C. Richardt, L. Mai, Y.-L. Yang, and N. Mitra. "BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images". In: *arXiv preprint arXiv:2002.08988* (2020).

[151] N. Nilson. *Shakey the Robot, SRI International*. Tech. rep. Technical Note 323, 1984.

[152] O. OpenAI, M. Plappert, R. Sampedro, T. Xu, I. Akkaya, V. Kosaraju, P. Welinder, R. D'Sa, A. Petron, H. P. d. O. Pinto, et al. "Asymmetric Self-play for Automatic Goal Discovery in Robotic Manipulation". In: *arXiv preprint arXiv:2101.04882* (2021).

[153] O. Ornan and A. Degani. "Toward Autonomous Disassembling of Randomly Piled Objects with Minimal Perturbation". In: *IEEE International Conference on Intelligent Robots and Systems* (2013), pp. 4983–4989. ISSN: 21530858. DOI: 10.1109/IROS.2013.6697076.

[154] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner. "Intrinsic Motivation Systems for Autonomous Mental Development". In: *IEEE Transactions on Evolutionary Computation* 11.2 (2007), pp. 265–286.

[155] P.-Y. Oudeyer and F. Kaplan. "What is Intrinsic Motivation? A Typology of Computational Approaches". In: *Frontiers in Neurorobotics* 1 (2009), p. 6.

[156] A. Pashevich, I. Kalevatykh, I. Laptev, and C. Schmid. "Learning Visual Policies for Building 3D Shape Categories". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 8073–8080.

[157] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. "Curiosity-driven Exploration by Self-supervised Prediction". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2778–2787.

[158] D. Pathak, D. Gandhi, and A. Gupta. "Self-Supervised Exploration via Disagreement". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 5062–5071. URL: http://proceedings.mlr.press/v97/pathak19a.html.

[159] A. Radford, L. Metz, and S. Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *International Conference on Learning Representations*. 2016.

[160] D. J. Rezende and F. Viola. "Generalized ELBO with Constrained Optimization, GECO". In: *Workshop on Bayesian Deep Learning, NeurIPS*. 2018.

[161] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg. "Learning by Playing – Solving Sparse Reward Tasks from Scratch". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4344–4353.

[162] M. B. Ring. "Continual Learning in Reinforcement Environments". In: (1994).

[163] R. Riochet, M. Y. Castro, M. Bernard, A. Lerer, R. Fergus, V. Izard, and E. Dupoux. "IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[164] I. Rock. "The Logic of Perception". In: (1983).

[165] D. A. Rosenbaum, K. M. Chapman, M. Weigelt, D. J. Weiss, and R. Van Der Wel. "Cognition, Action, and Object Manipulation." In: *Psychological Bulletin* 138.5 (2012), p. 924.

[166] R. Y. Rubinstein and D. P. Kroese. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2004. ISBN: 038721240X.

[167] M. Saito, E. Matsumoto, and S. Saito. "Temporal Generative Adversarial Nets With Singular Value Clipping". In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[168] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. "A Simple Neural Network Module for Relational Reasoning". In: *Advances in neural information processing systems*. 2017, pp. 4967–4976.

[169] T. Schaul, D. Horgan, K. Gregor, and D. Silver. "Universal Value Function Approximators". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1312–1320. URL: https://proceedings.mlr.press/v37/schaul15.html.

[170] J. Schmidhuber. "A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers". In: *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*. 1991, pp. 222–227.

[171] J. Schmidhuber. "Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010)". In: *IEEE Transactions on Autonomous Mental Development* 2.3 (2010), pp. 230–247.

[172] J. Schmidhuber. "PowerPlay: Training an Increasingly General Problem Solver by Continually Searching for the Simplest Still Unsolvable Problem". In: *Frontiers in Psychology* 4 (2013), p. 313.

[173] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model". In: *Nature* 588.7839 (2020), pp. 604–609.

[174] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. "Planning to Explore via Self-supervised World Models". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8583–8592.

[175] ShadowRobot. *ShadowRobot Dexterous Hand*. 2005. URL: https://fetchrobotics.com/wp-content/uploads/2018/04/Fetch-and-Freight-Workshop-Paper.pdf.

[176] A. Sharma, M. Ahn, S. Levine, V. Kumar, K. Hausman, and S. Gu. "Emergent Real-World Robotic Skills via Unsupervised Off-Policy Reinforcement Learning". In: *arXiv preprint arXiv:2004.12974* (2020).

[177] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. "Dynamics-aware Unsupervised Discovery of Skills". In: *arXiv preprint arXiv:1907.01657* (2019).

[178] E. Shelhamer, P. Mahmoudieh, M. Argus, and T. Darrell. "Loss is its own Reward: Self-Supervision for Reinforcement Learning". In: *arXiv preprint arXiv:1612.07307* (2016).

[179] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *Workshop at International Conference on Learning Representations*. 2014.

[180] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-scale Image Recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[181] S. Singh, A. G. Barto, and N. Chentanez. *Intrinsically Motivated Reinforcement Learning*. Tech. rep. MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE, 2005.

[182] R. Smith, M. Self, and P. Cheeseman. "Estimating Uncertain Spatial Relationships in Robotics". In: *Autonomous Robot Vehicles*. Springer, 1990, pp. 167–193.

[183] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn. "Universal Planning Networks". In: *International Conference on Machine Learning (ICML)*. 2018.

[184] R. K. Srivastava, B. R. Steunebrink, and J. Schmidhuber. "First Experiments with PowerPlay". In: *Neural Networks* 41 (2013), pp. 130–136.

[185] T. Standley, O. Sener, D. Chen, and S. Savarese. "image2mass: Estimating the Mass of an Object from Its Image". In: *Proceedings of the 1st Annual Conference on Robot Learning* 78.CoRL (2017), pp. 324–333. URL: http://proceedings.mlr.press/v78/standley17a.html.

[186] S. V. Steenkiste, K. Kurach, J. Schmidhuber, and S. Gelly. "Investigating Object Compositionality in Generative Adversarial Networks". In: *CoRR* abs/1810.10340 (2019). arXiv: 1810.10340. URL: http://arxiv.org/abs/1810.10340.

[187] S. Still and D. Precup. "An Information-theoretic Approach to Curiosity-driven Reinforcement Learning". In: *Theory in Biosciences* 131.3 (2012), pp. 139–148.

[188] A. Stoytchev. "Behavior-Grounded Representation of Tool Affordances". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2005, pp. 3071–3076.

[189] S. Sukhbaatar, Z. Lin, I. Kostrikov, G. Synnaeve, A. Szlam, and R. Fergus. "Intrinsic Motivation and Automatic Curricula via Ssymmetric Self-play". In: *arXiv preprint arXiv:1703.05407* (2017).

[190] I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3104–3112.

[191] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[192] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." In: *AAAI*. Vol. 4. 2017, p. 12.

[193] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta. "Exploring Affordances and Tool Use on the iCub". In: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. 2013, pp. 130–137.

[194] E. Todorov, T. Erez, and Y. Tassa. "MuJoCo: A Physics Engine for Model-Based Control". In: *IEEE International Conference on Intelligent Robots and Systems* (2012), pp. 5026–5033. ISSN: 21530858. DOI: 10.1109/IROS.2012.6386109.

[195] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly. "Towards Accurate Generative Models of Video: A New Metric & Challenges". In: *arXiv preprint arXiv:1812.01717* (2018).

[196] S. Van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber. "Relational Neural Expectation Maximization: Unsupervised Discovery of Objects and their Interactions". In: *International Conference on Learning Representations*. 2018.

[197] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[198] R. Veerapaneni, J. D. Co-Reyes, M. Chang, M. Janner, C. Finn, J. Wu, J. B. Tenenbaum, and S. Levine. "Entity Abstraction in Visual Model-Based Reinforcement Learning". In: *International Conference on Learning Representations*. 2019.

[199] C. Vondrick, H. Pirsiavash, and A. Torralba. "Generating Videos with Scene Dynamics". In: *Advances in Neural Information Processing Systems*. 2016, pp. 613–621.

[200] J. Wang, P. Rogers, L. Parker, D. Brooks, and M. Stilman. "Robot Jenga: Autonomous and Strategic Block Extraction". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009* (2009), pp. 5248–5253. DOI: 10.1109/IROS.2009.5354303.

[201] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. "Pixel2mesh: Generating 3D Mesh Models from Single RGB Images". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 52–67.

[202] D. Warde-Farley, T. V. de Wiele, T. Kulkarni, C. Ionescu, S. Hansen, and V. Mnih. "Unsupervised Control Through Non-Parametric Discriminative Rewards". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=r1eVMnA9K7.

[203] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller. "Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2746–2754.

[204] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti. "Visual Interaction Networks: Learning a Physics Simulator from Video". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4539–4547.

[205] P.-B. Wieber. "On the Stability of Walking Systems". In: *Proceedings of the Third IARP International Workshop on Humanoid and Human Friendly Robotics* (2002), pp. 1–7. ISSN: 02649381. DOI: 10.1088/0264-9381/12/2/003. arXiv: 9412172 [hep-th].

[206] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich. *Fetch & Freight: Standard Platforms for Service Robot Applications*. 2018. URL: https://fetchrobotics.com/wp-content/uploads/2018/04/Fetch-and-Freight-Workshop-Paper.pdf.

[207] J. Wu, J. J. Lim, H. Zhang, J. B. Tenenbaum, and W. T. Freeman. "Physics 101: Learning Physical Object Properties from Unlabeled Videos." In: *BMVC*. Vol. 2. 6. 2016, p. 7.

[208] J. Wu, E. Lu, P. Kohli, W. T. Freeman, and J. B. Tenenbaum. "Learning to See Physics via Visual De-animation". In: *Advances in Neural Information Processing Systems* NIPS (2017).

[209] J. Wu, J. B. Tenenbaum, and P. Kohli. "Neural Scene De-rendering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 699–707.

[210] J. Wu, I. Yildirim, J. Lim, W. Freeman, and J. Tenenbaum. "Galileo : Perceiving Physical Object Properties by Integrating a Physics Engine with Deep Learning". In: *Advances in Neural Information Processing Systems 28 (NIPS 2015)* (2015), pp. 1–9. ISSN: 10495258.

[211] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. 2016, pp. 82–90.

[212] Y. Wu, S. Kasewa, O. Groth, S. Salter, L. Sun, O. Parker Jones, and I. Posner. "Learning Affordances in Object-Centric Generative Models". In: *Workshop on Object-Oriented Learning at ICML 2020* (July 2020).

[213] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2020), pp. 4–24.

[214] M. Wulfmeier, A. Abdolmaleki, R. Hafner, J. T. Springenberg, M. Neunert, T. Hertweck, T. Lampe, N. Siegel, N. Heess, and M. Riedmiller. "Compositional Transfer in Hierarchical Reinforcement Learning". In: *Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2020.

[215] M. Wulfmeier, A. Byravan, T. Hertweck, I. Higgins, A. Gupta, T. Kulkarni, M. Reynolds, D. Teplyashin, R. Hafner, T. Lampe, et al. "Representation Matters: Improving Perception and Exploration for Robotics". In: *arXiv preprint arXiv:2011.01758* (2020).

[216] M. Wulfmeier, D. Rao, R. Hafner, T. Lampe, A. Abdolmaleki, T. Hertweck, M. Neunert, D. Tirumala, N. Siegel, N. Heess, et al. "Data-efficient Hindsight Off-policy Option Learning". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 11340–11350.

[217] A. Xie, F. Ebert, S. Levine, and C. Finn. "Improvisation through Physical Understanding: Using Novel Objects as Tools with Visual Foresight". In: *arXiv preprint arXiv:1904.05538* (2019).

[218] D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese. "Neural Task Programming: Learning to Generalize Across Hierarchical Tasks". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–8.

[219] T. Xue, J. Wu, K. Bouman, and B. Freeman. "Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 91–99. URL: http://papers.nips.cc/paper/6552-visual-dynamics-probabilistic-future-frame-synthesis-via-cross-convolutional-networks.pdf.

[220] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada. "A brief review of affordance in robotic manipulation research". In: *Advanced Robotics* 31.19-20 (2017), pp. 1086–1101.

[221] T. Ye, X. Wang, J. Davidson, and A. Gupta. "Interpretable Intuitive Physics Model". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 87–102.

[222] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani. "Object-Centric Forward Modeling for Model Predictive Control". In: *Conference on Robot Learning*. 2020, pp. 100–109.

[223] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani. "Compositional Video Prediction". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 10353–10362.

[224] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum. "Clevrer: Collision Events for Video Representation and Reasoning". In: *International Conference on Learning Representations*. 2020.

[225] W. Yifan, N. Aigerman, V. Kim, S. Chaudhuri, and O. Sorkine-Hornung. "Neural Cages for Detail-Preserving 3D Deformations". In: *arXiv preprint arXiv:1912.06395* (2019).

[226] I. Yildirim, T. Gerstenberg, B. Saeed, M. Toussaint, and J. Tenenbaum. "Physical problem solving: Joint planning with symbolic, geometric, and dynamic constraints". In: *CoRR* abs/1707.08212 (2017). arXiv: 1707.08212. URL: http://arxiv.org/abs/1707.08212.

[227] I. Yildirim, M. Siegel, and J. Tenenbaum. "34 Physical Object Representations". In: *The Cognitive Neurosciences* (2020), pp. 399–409.

[228] W.-K. Yoon, T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda, and T. Doi. "Model-based Space Robot Teleoperation of ETS-VII Manipulator". In: *IEEE Transactions on Robotics and Automation* 20.3 (2004), pp. 602–612.

[229] T. Yu, G. Shevchuk, D. Sadigh, and C. Finn. "Unsupervised Visuomotor Control through Distributional Planning Networks". In: *Proceedings of Robotics: Science and Systems*. Freiburg im Breisgau, Germany, June 2019. DOI: 10.15607/RSS.2019.XV.020.

[230] M. D. Zeiler and R. Fergus. "Visualizing and Understanding Convolutional Networks". In: *Proceedings of the IEEE European Conference on Computer Vision*. 2014, pp. 818–833.

[231] R. Zhang, J. Wu, C. Zhang, W. T. Freeman, and J. B. Tenenbaum. "A Comparative Evaluation of Approximate Probabilistic Simulation and Deep Neural Networks as Accounts of Human Physical Scene Understanding". In: *arXiv preprint arXiv:1605.01138* (2016).

[232] B. Zheng, Y. Zhao, J. Yu, K. Ikeuchi, and S. C. Zhu. "Scene Understanding by Reasoning Stability and Safety". In: *International Journal of Computer Vision* 112.2 (2015). ISSN: 15731405. DOI: 10.1007/s11263-014-0795-4.

[233] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, et al. "Reinforcement and Imitation Learning for Diverse Visuomotor Skills". In: *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania, June 2018. DOI: 10.15607/RSS.2018.XIV.009.