Scene Perception for Simulated Intuitive Physics via Bayesian Inverse Graphics

by

Khaled K. Shehada

S.B. in Computer Science and Engineering
Massachusetts Institute of Technology (2022)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2023

© 2023 Khaled K. Shehada. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Khaled K. Shehada
Department of Electrical Engineering and Computer Science
August 11, 2023

Certified by: Nicholas Roy
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Scene Perception for Simulated Intuitive Physics via Bayesian Inverse Graphics

by
Khaled K. Shehada

Submitted to the Department of Electrical Engineering and Computer Science
on August 11, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Humans have a wide range of cognitive capacities that make us adept at interpreting our physical world. Every day, we encounter new environments, yet we can parse those environments with limited visual exposure and make fairly accurate inferences about unfamiliar objects. Emulating scene understanding capacities in computational models has numerous applications ranging from autonomous driving to virtual reality. Despite the proficiency demonstrated by deep neural networks in pattern recognition, recent works have uncovered challenges in their abilities to encode prior physical knowledge, form visual concepts, and perform compositional reasoning, such as inferring inter-object relations like containment. To this end, the thesis introduces the **S**imulated **CO**gnitive **T**asks (**SCOT**) benchmark, a large-scale synthetic dataset and data creation codebase allowing for the procedural generation of videos of simulated cognitive tasks targeting intuitive physics understanding. Those cognitive tasks are adapted from tests in the literature used to comparatively assess the cognitive capacities of non-human primates. Additionally, the thesis presents an analysis of several deep learning models on the benchmark, underlining their limitations in tasks involving object permanence comprehension, quantities, and compositionality and their inability to generalize learned knowledge to complex dynamic scenes. In response to these limitations, we propose a probabilistic generative approach that leverages Bayesian inverse graphics to learn structured scene representations that facilitate learning new objects and tracking objects in dynamic scenes. Our evaluation of this model on **SCOT** revealed near-perfect performance on most tasks with significant data efficiency, suggesting that structured representations and symbolic inference can cooperate with deep learning methods to interpret complex 3D scenes accurately. Overall, this thesis contributes to the field of artificial intelligence (AI) by presenting a new method for improving scene understanding in AI models and providing a benchmark for assessing the visual cognitive capacities of computational models.

Thesis Supervisor: Nicholas Roy
Title: Professor of Aeronautics and Astronautics

Acknowledgments

To my dear parents, Khalil and Amal, who have stood by my side in every chapter of my educational journey and never missed a chance to prioritize my well-being and aspirations. To my beloved siblings, Saja, Lama, and Amir; and to my friends who weathered the storm of my MEng journey. To each and every one of you, my heart is filled with profound gratitude.

Many thanks also go to my supervisor, Professor Nicholas Roy. His inspiring leadership, thoughtful feedback, and gentle nudges toward forward planning were the keystones to my timely MEng completion. I would also like to express my deep and sincere gratitude to Dr. Dhaval Adjodah. His assistance in refining the research problem and his support of my academic and professional aspirations have been pivotal.

Likewise, I am grateful to Nishad Gothoskar for providing insight and technical support on the implementation of the probabilistic generative model. My gratitude extends to the MIT Quest for Intelligence, especially Erik Vogan and Katherine Fairchild, who provided me with all the necessary resources and support to carry out this research.

Lastly, I thank Professor Joshua Tenenbaum and Professor Vikash Mansinghka. The knowledge I acquired from 9.660, along with their published works, have enriched my understanding of the technical subject and inspired me to write this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Imagine walking into a new place. Within mere moments, you begin to parse your surroundings: you identify objects, interpret their context within the environment, and develop an intuition about their relationships. If something drops, it will probably fall *down*. If a container is moved, everything inside it will be moved along with it. This ability to continually and reliably parse unfamiliar physical scenes, often through limited visual exposure [9], marks one of the most distinctive features of human intelligence.

Similarly, like a child constantly discovering the world, we frequently encounter new objects, learn their shapes and attributes, and even with minimal exposure, we can identify those objects in different scenes, poses, and orientations. These abilities stem from a spectrum of cognitive capacities that, through a combination of genetic endowment and experience, allow us to generate mental hypotheses about the observed world, test those hypotheses, and update our mental model of the physical world accordingly [61].

Developing computational models that embody similar cognitive capacities, namely scene understanding, can prove helpful in many applications, ranging from robotic manipulation and autonomous driving to augmented and virtual reality. For instance, robots interacting with objects need to comprehend the objects and their dynamics accurately [84]. Similarly, autonomous vehicles need to parse through significant sensory input and efficiently characterize the structures and locations of objects. There-

fore, these models need not only to understand visual cues but also to construct a compositional and causal representation of the physical world, incorporating prior knowledge [61]. They must also possess a robust mechanism to correct errors and update their knowledge when needed [84]. As numerous research and industry sectors increasingly rely on artificial intelligence (AI) for automated inference, a reliable model that can accurately understand and interpret 3D scenes in real time becomes necessary.

1.2 Problem Statement

Models that aim to learn similar cognitive functions for scene understanding should possess certain ingredients that enable them to interpret 3D scenes through limited and noisy visual input. Those ingredients include:

- An *intuitive physics engine*, as described by Battaglia et al. [9], encodes characteristics about objects and generates physically plausible hypotheses when reasoning about the physical world.
- The capacity to explain observed physical phenomena through a causal (e.g., generative) framework [22], supplementing the intuitive physics model with observation.
- A structured, compositional representation of scenes that encodes object poses and relations and maintains that compositionality over time in dynamic scenes [84].

While many other ingredients are essential in matching human-level cognition, such as an intuitive model of the psychology of agents and theory of mind [7], those three are highlighted because of their relevance to scene understanding and their significance to this project.

Many computational approaches to scene understanding use deep neural networks, which primarily rely on matching the computational model with the statistical patterns of the data. Nonetheless, while neural networks show remarkable performance on vision downstream tasks such as object recognition and image classification, they

lack an inherent mechanism to encode prior physical knowledge, causality, and compositionality [40]. Therefore, it is difficult to investigate whether the representations learned by those models approximate the human core knowledge systems [94] or if they generalize to complex environments through abstraction and concept formation.

The problem this thesis aims to address is twofold. Firstly, despite significant advances, bottom-up deep-learning approaches to scene perception struggle with learning intuitive physics, causality, and compositionality. Thus, there's a need to explore alternate models that encapsulate these aspects and investigate whether they make more accurate and robust predictions.

Secondly, the current benchmarks and methodologies used to evaluate models for scene understanding primarily evaluate the corresponding downstream tasks. Downstream tasks seldom target the specific cognitive functions involved. Thus, it is necessary to develop a benchmark that approaches scene perception as a cognitive task and evaluates core knowledge understanding of scenes.

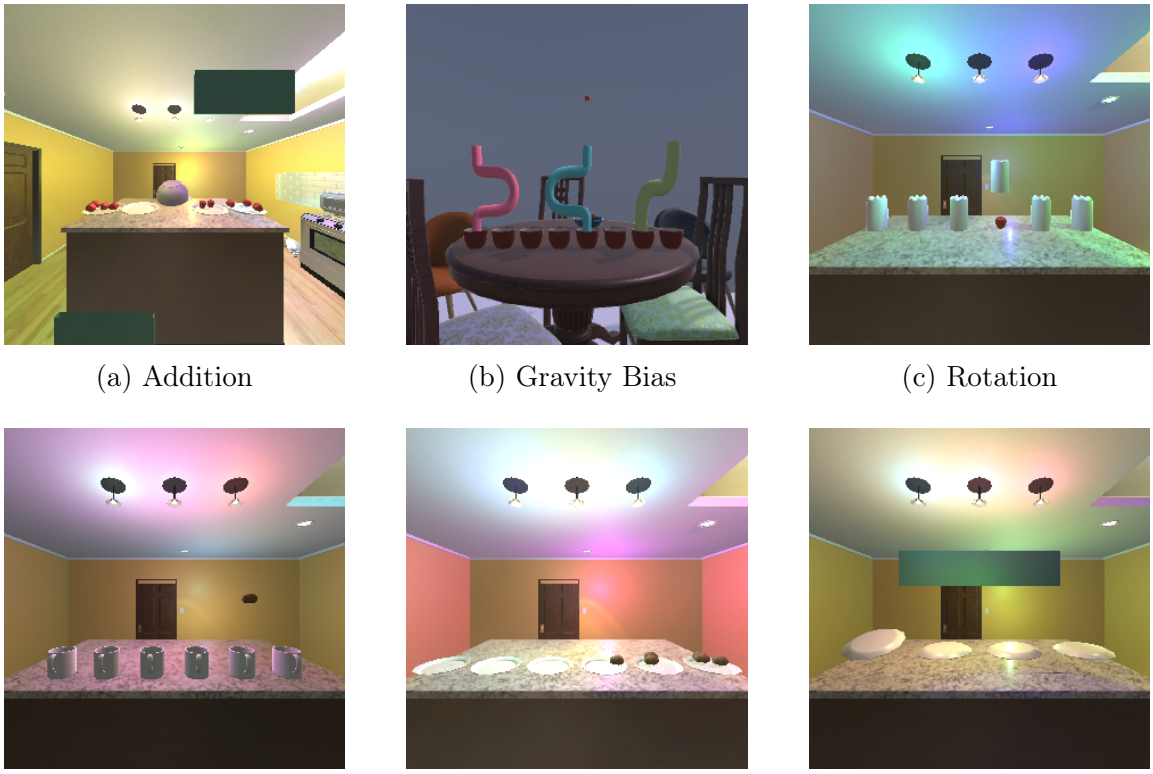
1.3 Approach

We propose to address these issues in two steps: (1) by developing a benchmark to evaluate the high-level core knowledge of objects, space, and numbers in agents using tasks from academic literature on cognition; and (2) by integrating object-level structured inference (using Bayesian modeling) and analysis by synthesis (using inverse graphics) with deep learning models to learn more generalizable scene representations.

1.3.1 SCOT: Experimental Test Suite

We develop a procedural scene generator on top of the Unity [52] game engine and AI2Thor [57] comprised of six cognitive tasks designed to target concepts such as causality, physics biases, and compositionality. Hence, we develop the **S**imulated **CO**gnitive **T**asks (**SCOT**) benchmark: a library of simulated intuitive physics videos inspired by cognitive tests of non-human primates [88], to evaluate the generalizability of learned knowledge.

At its core, each task in the **SCOT** benchmark requires an agent to observe a dynamic scene (video) with rewards and receptacles subject to physical manipulation. The observing agent aims to identify the final location(s) that contain the most rewards at the end of the video. Inspired by reinforcement learning principles, this design philosophy focuses on maximizing cumulative reward through interaction with the environment. However, in our benchmark, the agent only observes a scene and infers the final locations with the highest reward accumulation, necessitating prior knowledge about objects and space that aligns with the core knowledge framework. This emphasis on long-term reward maximization highlights the importance of perception, and it aligns with broader artificial intelligence goals of creating models capable of identifying and tracking objects in complex, dynamic, and noisy environments.



(a) Addition (b) Gravity Bias (c) Rotation

(d) Simple Swap (e) Relative Numbers (f) Shape Causality

Figure 1-1: **Samples from the Simulated Cognitive Tasks** The proposed benchmark comprises six tasks, each assessing a specific cognitive ability. Tasks involve physical manipulation of one or more rewards with scene objects to maximize the agent's reward acquisition. This requires understanding the scene's compositional semantics and consistently tracking object locations over time.

1.3.2 Deep Learning Baselines

Moreover, we use the **SCOT** benchmark to investigate current computational approaches to scene understanding, focusing primarily on deep-learning methods. In particular, we fine-tune the parameters of several pre-trained deep-learning models using gradient descent and backward propagation on a portion of the video dataset and evaluate it on videos with more complex dynamics.

Chapter 4 details the implementation of a PyTorch-based [75] framework that enables the training and evaluation of deep learning models on the benchmark, and it provides experimental results that show significant performance decreases as the physical dynamics in videos grow more complex. Our analysis demonstrates that while neural models trained on tasks perform much better than chance in tasks that require spatial awareness, they perform at or slightly above chance on tasks that require an understanding of object performance, numbers, and compositionality. In addition, we observe a decline in zero-shot performance when we increase the task complexity, indicating a limitation in their capacity to generalize the learned knowledge.

1.3.3 Bayesian Modeling & Inverse Graphics

Inverse graphics, a concept that views 3D scene understanding as the inverse problem to 3D graphics rendering, can serve as an intuitive physics engine in computational models. Specifically, it provides a mechanism for generating physically plausible hypotheses when reasoning about the physical world. Combined with deep learning, it can enhance the focus on primary effects like occlusion and projection, improving tasks such as object pose estimation [76].

Concurrently, Bayesian models offer a structured approach to understanding causality and compositionality in dynamic scenes. In particular, a Bayesian model that operates with a scene graph can handle uncertainties about scene structures by comparing its scene graph state with the observed scenes, and they can incorporate prior knowledge about physics to enrich the intuitive physical model with observations. The careful selection and learning of priors, particularly beneficial in complex systems, can bridge gaps in current deep learning approaches, leading to more comprehensive

computational models for scene understanding [54].

We employ an integrative approach that combines the strengths of Bayesian modeling and inverse graphics to develop a probabilistic generative model capable of internally representing scene states and making inferences about object movements, compositional structures, and causal dynamics in the scenes. A more detailed discussion about the probabilistic model is presented in Chapter 5.

The model operates by first inferring object shapes (meshes) from the depth images by iteratively comparing them to a library of stored object models and identifying the most likely object type under a uniform prior.

Consequently, it tracks object movements by generating pose hypotheses through coarse adjustments to its representation of object poses and renders them using a physics engine. It then uses a voting mechanism to determine the most plausible object poses, all while ensuring no violations of compositionality. In addition, the probabilistic model is based on Bayesian principles and consists of (i) a dynamics prior that encodes physical phenomena such as gravity and velocity and (ii) a likelihood that estimates noise in depth information between the observed depth images and the images generated from the pose hypotheses. It uses a first-order Markov Chain to iteratively update its state of object poses and scene graph.

Finally, it uses a maximum a posteriori estimator to make categorical inferences about each scene type. As shown in Chapter 5, the model exhibits an impressive performance, achieving perfect accuracy on multiple cognitive tasks.

1.4 Contributions

This thesis contributes in three main areas:

(i) the **SCOT** benchmark, a dataset of $10,000$ videos consisting of over 2.3 million RGB images, along with their depth images, segmentation masks, and rich metadata, varying across six intuitive physics tasks, as well as the methodology and the generation codebase for its synthesis and potential extensibility;

(ii) a systematic analysis and experimental results of several deep learning models on

the **SCOT** benchmark, including the methodology and codebase of a framework for automatic data-loading, training, and evaluation in PyTorch; and

(iii) a formulation and implementation of a probabilistic generative model that uses Bayesian inverse graphics with a dynamics prior and a depth noise likelihood for scene perception, as well as experimental results showing perfect performance in multiple tasks, measured on the **SCOT** benchmark.

1.5 Thesis Overview

Following this introduction, Chapter 2 provides a literature review on scene understanding and the different computational approaches used for scene understanding, including deep learning, reinforcement learning, and 3D inverse graphics. Chapter 3 delves deeper into the **SCOT** benchmark, providing a detailed description of each task and the methodology for generating the large-scale synthetic dataset. Subsequently, Chapter 4 examines the performance of deep-learning models on the **SCOT** benchmark and shows their limitations in generalizing knowledge to more complex tasks. Then, Chapter 5 details our proposed model: a probabilistic generative model that combines Bayesian modeling and inverse graphics. Lastly, Chapter 6 provides a direction for future research, suggesting the use of real-world data to increase model robustness, expanding the types and variety of cognitive tasks, testing more recent models and techniques, and implementing Markov Chain Monte Carlo (MCMC) methods for more efficient fine-tuning of the inference pipeline. We conclude the thesis by summarizing the contributions and final remarks in Chapter 7.

Chapter 2

Related Work

2.1 Chapter Overview

This chapter reviews relevant literature on scene perception, both in humans and through computational methods.

First, Section 2.2 presents studies on scene understanding in humans, examining the cognitive and neural mechanisms involved and the methods used for investigating these processes. We then move on to computational approaches, including deep learning techniques and 3D inverse graphics for scene perception in Section 2.3

Additionally, Section 2.4 explores the role of synthetic and simulated data in training vision models, focusing on the potential for generating large amounts of pre-labeled training data. Finally, Section 2.5 reviews specific tasks related to scene perception and their importance to current research, and Section 2.6 briefly reviews concepts from probability theory and Bayesian inference.

2.2 Scene understanding in Humans

Perception of physical scenes in humans plays a central role in how we interact with the world around us, from identifying tools and their potential uses to building a stack of blocks in a game of Jenga. Nonetheless, the neural and cognitive mechanisms underlying scene perception remain an interesting research question.

A line of research investigates the evolution of human cognitive capacities through a comparative assessment with non-human primates [45, 88, 32]. The *Primate Cog-*

nition Test Battery [45] is a collection of behavioral tasks that target specific physical and social cognitive capacities. For physical cognition, tests target various core knowledge domains [94], such as objects, actions, numbers, and space. Examples of such tasks include locating a reward while undergoing physical manipulation (rotation, occlusion, or transposition) [93, 51, 43], discriminating quantities [12], causally inferring added and removed quantities [85], and inferring the location of an object through its causal effects on other objects [13].

To identify the cognitive computations involved, long and parallel lines of research have investigated the neurocognitive mechanisms involved in scene perception. Here, four relevant lines of work are highlighted: hierarchical processing, attention and working memory, the intuitive physical engine framework, and learning from experience.

Recent research has suggested that the brain uses hierarchical processing for scene perception [47, 4, 31]. In particular, Epstein et al. [31] argue the visual system uses goal-specific hierarchical processing and processes information about a scene at multiple levels: low-level features like color and edges, mid-level elements like layout and objects, and high-level semantic and spatial properties like scene category.

In addition, cognitive processes such as attention and short-term memory play a role in scene perception. Visual attention [19] allows us to selectively process vast amounts of visual information. For example, in object recognition, spatial attention in the form of horizontal connections between nearby neurons in the same visual area and feedback connections from higher-level visual areas facilitate identifying and segmenting objects in visual scenes [18].

On an algorithmic cognitive level, scene understanding has been approached through an intuitive physics engine framework [9, 42, 5], whereby the brain uses probabilistic simulations about the physical state of an observed world to make approximate inferences about future unobserved states. In this framework, a physical model of the world is represented in a latent space, and future states are generated by recursively applying learned physical principles over time. Hence, in dynamic scenes, this model evaluates acceptable hypotheses through simulation and updates its state of

the world [80] or problem-solving strategy [5] through limited observations.

Furthermore, experience and learning are crucial in human scene perception. In particular, the brain's ability to extract and process information from a scene is not static but evolves with experience and depends on the context of the visual information. For instance, Bainbridge et al. [6] showed that personal experience with scenes can influence scene recognition, indicating that past experiences can shape how we perceive and understand scenes. In addition, Epstein et al. [31] highlighted the role of learning in the development of scene-selective regions in the brain, suggesting that these regions may be shaped by our experiences and learning throughout life.

2.3 Advances in Computational Scene Perception

2.3.1 Machine Learning Approaches

Many computational approaches to scene understanding involve the use of artificial neural networks. Computational models in this framework often encode images into a learned latent space using a convolutional neural network (CNN) [65] or transformer model [103] and learn classifiers or regressors that minimize an objective function over a set of downstream tasks. Such tasks usually include object segmentation and recognition, 3D scene reconstruction, and depth and pose estimation. With recent advances in deep neural networks and the introduction of multi-million-scale image datasets such as ImageNet [24], COCO [67], and many others [60, 1, 20, 58], state-of-the-art deep learning models have shown great promise on scene understanding tasks, achieving near-human-level accuracies.

Convolutional neural network (CNN)-based models such as ResNet and DenseNet have made significant strides in image classification and object recognition tasks. ResNet introduced an architecture that includes *skip connections* for training deeper networks [44]. DenseNet models connect each layer to every other layer in the network, strengthening feature propagation and reuse [48]. Non-convolution neural networks that employ spatial attention have also been proposed. For example, Dosovitskiy et al. [28] proposed Vision Transformers (ViT) and showed that they outperform es-

tablished CNN models. Furthermore, vision models can be combined with recurrent LSTM networks [46] to create sequential models with a convolutional backbone, allowing them to handle video data with temporal dependencies. This combination leverages the strength of CNNs in extracting spatial features from images and the ability of LSTMs to model temporal dependencies, making it a powerful tool for video-based tasks [27, 104, 86].

Recently, transformer-based models have also been applied to video scene understanding. For instance, TimeSformer applies self-attention across space and time in videos, capturing long-range spatiotemporal dependencies efficiently [10]. Similarly, XClip leverages both transformer and convolutional architectures to learn representations from a large amount of weakly-supervised data, using a two-branch architecture to process image frames and text data, thereby capturing both visual and textual information [70]. Furthermore, VideoMAE uses transformer-based masked auto-encoding to learn video representations, predicting masked-out patches in the video to understand the context in both spatial and temporal dimensions [98].

One limitation of the previous approaches is that they do not enforce particular structure learning; hence they tend to fail on tasks where the data involves complex relationships and interactions. Graph neural network (GNN) models that learn structured graph representations have been used for scene understanding. For instance, Chen et al. [17] use GNNs to predict scene graphs by propagating information between objects in the scene. Similarly, Huang et al. [49] propose a GNN for instance segmentation by using a textual query to guide information aggregation in neighboring nodes in graph representations.

In reinforcement learning, a common approach to scene perception is to use convolutional networks to extract visual features from the environment. These features can then inform the agent about the current state of the environment [66, 33, 3]. For example, Moses et al. [74] use a convolutional network to encode visual cues in the environment and learn a policy that correlates observed features with actions, leading to learning more generalizable policies. Another example is Gupta et al. [41], where a model uses a combination of convolutional processing backbones and recurrent units

to learn a policy that guides the agent through solving a maze. In particular, the model learns to construct a spatial map of the environment from visual input and uses this map to plan a path to the goal. Likewise, Adjodah et al. [2] introduce relational architectures, which employ neural network sub-modules, in reinforcement learning and showed improved generalization by transforming input states into relational object representations.

2.3.2 3D Inverse Graphics

Another common approach to computational scene understanding involves 3D inverse graphics. The core idea behind inverse graphics is to reverse the graphics rendering process, going from 2D images to 3D scene representations. Hence, the inverse graphics method is inherently probabilistic since the same 2D image can correspond to multiple 3D scenes due to occlusions, lighting conditions, and other factors.

Analysis-by-synthesis involves generating hypotheses about the scene's structure and comparing it with the observed data [107]. This approach resembles a Bayesian process as it involves computing a posterior distribution over possible scenes given the observed data. In addition, models following this approach tend to be explicitly stateful and can generalize to various dynamic compositions, especially for modeling states under limited observability conditions such as occlusion [39, 111]

However, traditional analysis-by-synthesis approaches often involve hand-crafted models and are computationally expensive as they require repeated rendering and likelihood evaluation with the observed data. To overcome these limitations, recent work has focused on learning-based approaches that accelerate inference [23, 71, 38]. For instance, Picture [71] allows for the specification of generative models that include graphics rendering operations and uses a combination of MCMC and variational inference methods to infer the scene structure.

2.4 Learning from Simulated Data

Using synthetic and simulated data in training vision models has gained increased interest in artificial intelligence research. In particular, the ability to generate large

amounts of labeled data without manual annotation effort is a significant advantage, especially for tasks where real-world data collection and annotation can be challenging or expensive. Hence, these datasets provide diverse environments and objects that aid in tasks such as semantic segmentation, object detection, and pose estimation.

One of the key advantages of synthetic data is the ability to relatively-cheaply generate a large amount of pre-labeled training data with little effort. The mass generation procedures often rely on physical simulators such as Unity [52], Unreal [30], and Drake [97] to provide realistic object poses, accurate physics, and plausible scenery. Accordingly, platforms such as ThreeDWorld [34], AI2Thor [57], and Habitat [87, 96] provide a layer of abstraction specifically designed for machine learning research by providing pre-built environments, objects, and interactions that are commonly used in machine learning tasks. For instance, ThreeDWorld [34] provides tools for generating complex 3D scenes with realistic lighting and textures and allows for control over camera parameters. Meanwhile, AI2Thor [57] provides an interactive 3D environment for visual AI. It includes a variety of pre-built objects and scenes and provides high-quality visual rendering and physics simulation. This makes it suitable for tasks that require high-level scene understanding.

To facilitate the mass generation of synthetic data, dataset creators often develop custom simulators that generate task-specific scenes sampled from a distribution over the possible scene generation parameters (e.g. colors, lighting, object poses, etc..) [63, 72, 108, 15]. For instance, Synthia [83] provides a virtual city simulator with pixel-level annotations and sequences of images with temporal consistency. Similarly, datasets such as SyVic [14] and BlenderProc [25] provide modular pipelines that generate realistic images and metadata for various computer vision tasks.

In robotics, synthetic environments have been used to generate simulated datasets for embodied agent learning. For instance, Tremblay et al. [100] developed the Falling Things dataset, a synthetic resource for 3D object detection and pose estimation, which proved beneficial for training models for robotic manipulation tasks. Similarly, the GeneSIS-Rt framework [95], RobotriX dataset [36], and others [101, 29, 72], were introduced to show that learning from large-scale and photorealistic synthetic im-

ages can improve the performance of robotic vision systems for tasks such as robot localization, mapping, and object interaction.

While synthetic and simulated data offer advantages, they also present challenges, especially the domain gap between synthetic and real-world data. This gap can lead to models that perform well in simulation but poorly in real-world tasks, primarily due to the significant differences between the characteristics of real versus simulated visual information [100].

Multiple approaches have been proposed to bridge the domain gap. One approach is domain randomization, where the irrelevant attributes of the synthetic data are randomized to force the model to learn the essential features of the object of interest [99]. In addition, multiple domain adaptation techniques have been investigated where the synthetic data is enhanced with real data features through stylization [112, 50] to make the models robust to diversity in styles, or Adversarial training [102, 91, 35] to align the latent features of simulated data with those of real data. Nonetheless, Baradad et al. [8] have demonstrated that augmenting synthetic images with structured noise improves representation learning as long as the generated noise embodies certain structural properties of real data and offers sufficient diversity.

2.5 Review of Scene Perception Tasks

We refer to three main tasks that pertain to scene perception throughout the manuscript.

Object Identification aims to assign a predefined class label to an object given its representation. In the case of 2D images, given an image I of $H \times W$ pixels, each of $\in \mathbb{Z}^3$ between 0 and 255, and a set of object classes $O = \{o_1, o_2, ..., o_n\}$, the task can be formulated as a function $f : [\![0, 255]\!]^{H \times W \times 3} \rightarrow O$, as per [59, 92, 44]. When dealing with depth point clouds, the task remains the same but the input changes. A depth point cloud is a set of points in a 3D space, each with its own spatial coordinates. We denote a depth point cloud as $\mathbf{c} \in \mathbb{R}^{H \times W}$. The task can be formulated as a function $g : \mathbb{R}^{H \times W} \rightarrow O$, as per [78, 79].

Semantic Segmentation aims to assign a class label to each pixel in an image,

hence partitioning the image into regions corresponding to different objects or parts of objects. Given an input image I, semantic segmentation can be formulated as a function $f : [\![0, 255]\!]^{H \times W \times 3} \to O^N$, where O is the set of object classes and N is the number of pixels in the image [69, 16, 62, 73].

Object Pose Estimation aims to determine an object's 3D translation and orientation in a scene. Given an image I and an object class o, the task can be mathematically formulated as a function $f : O \times [\![0, 255]\!]^{H \times W \times 3} \to \mathbb{R}^3 \times SO(3)$. That is, f outputs a 6D pose $p = (t, R)$ for the object o in the image I, where $t \in \mathbb{R}^3$ is the translation vector representing the object's position in 3D space, and $R \in SO(3)$ is the rotation matrix representing the object's orientation. Notable works in this area include using deep learning for 6D pose estimation in images [109, 53, 105].

2.6 Preliminaries: Bayesian Inference

2.6.1 Events and Sample Space

Sample space, often represented by S or Ω, refers to the entire set of all possible outcomes that can be derived from an experiment. For example, when flipping a fair coin, the sample space encompasses two possibilities: $\Omega = \{heads, tails\}$.

On the other hand, an event is a subset (one or multiple outcomes) of this sample space. Applying this to the coin toss scenario, one could define an event A such that $A = \{heads\}$. When the result is *heads*, event A has occurred.

2.6.2 Bayesian Networks

Bayesian networks serve as graphical models that encapsulate the probabilistic relationships among variables. Each node in a Bayesian network symbolizes a random variable, whereas the edges represent the conditional dependencies between these variables. If two variables lack a connecting edge, they are conditionally independent.

For example, consider three binary variables: Rain (R), Sprinkler (S), and Wet Grass (W). One could construct a Bayesian network where the occurrence of rain (R)

influences whether the sprinkler is activated (S), and both rain and sprinkler whether the grass is wet (W). Using this network, one can easily calculate probabilities such as $P(W = \text{True}|R = \text{True}, S = \text{False})$, indicating the probability of the grass being wet given it rained and the sprinkler was not activated.

2.6.3 Markov Models and Stochastic Processes

Stochastic processes are used for predicting a sequence of potential events wherein the future state could rely on the current state and a certain level of randomness. Markov models are examples of stochastic processes that describe systems changing over time under the influence of random events. The distinguishing factor of Markov models is that the probability of each event relies solely on the state of the preceding events. This attribute is often referred to as the Markov property.

A first-order Markov model is a Markov model where the likelihood of transitioning to the subsequent state depends only on the current state and not the sequence of previous events. Formally, given a sequence of random variables $X_1, X_2, ..., X_n$, for any i the likelihood $P(X_{i+1} = x|X_1 = x_1, ..., X_i = x_i)$ equals $P(X_{i+1} = x|X_i = x_i)$. Hence, the future state depends solely on the current state, with the past states having no influence, making it *memoryless*. For example, if we are modeling the weather, we might assume that tomorrow's weather depends only on the weather today and not on the weather of all previous days.

2.6.4 Probability Distributions

Uniform Distribution defined over a range $[a, b]$, with all outcomes being equally likely.

$$U(x; a, b) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

Gaussian Distribution a common probability distribution for a real-valued random variable.

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Chapter 3

Proposed Benchmark

3.1 Chapter Overview

Inspired by the Primate Cognition Test Battery [88], we leverage the advantages of synthetic data, including its cost-effectiveness, flexibility, and rapid generation, to evaluate computational models on intuitive physics cognitive tests. In particular, we construct the **S**imulated **CO**gnitive **T**asks (**SCOT**) benchmark, which comprises six intuitive physics tasks adapted from literature in cognitive tests for human infants and non-human primates [93, 51, 43, 12, 85, 13].

This chapter introduces the **SCOT** benchmark, detailing the six tasks, their objectives, and the task-specific complexity definitions (Section 3.2). It also explains the process of creating the synthetic dataset and discusses potential biases (Section 3.3). The chapter concludes by formalizing the dataset into a benchmark, outlining its metrics, use cases, and how it can be utilized for training and evaluating computational models (Section 3.4).

3.2 Simulated Cognitive Tasks

3.2.1 Examination of the Cognitive Tasks

We implement a system that facilitates simulations of six cognitive tasks, each designed to evaluate a certain cognitive capability of agents. Below, we list the intuitive physics scenes and tasks associated with each of them. Figure 1-1 shows examples of the described scenes.

Addition Multiple rewards are moved from a central plate to one of several other occluded plates. The agent's goal is to identify which plate holds the most rewards after this transfer. This task challenges the agent's ability to keep track of multiple occluded objects and perform basic numerical computations.

Gravity Bias Multiple rewards are dropped into an opaque tube with a complex geometric design, leading to one of several receptacles. The agent's understanding of how gravity influences the movement of objects is tested. In particular, by observing the rewards' initial movement and the tube's shape, the agent is tasked with predicting in which receptacle the rewards will end up. This task tests the agent's grasp of fundamental physics principles in a dynamic setting.

Rotation A reward is initially hidden under one of several receptacles. Random pairs of receptacles are then rotated, and the agent's task is to predict under which receptacle the reward ends up. This task tests the agent's ability to understand and track spatial movements.

Simple Swap Inspired by the classic shell game, a reward is initially hidden into or under one of several receptacles, which are then shuffled around randomly. The agent's challenge is to keep track of the reward's location and correctly identify the bowl hiding the reward at the end of the shuffling sequence. This task tests the agent's ability to persistently track an object's location and its ability to track moving objects.

Relative Numbers The agent is presented with several plates, each containing a certain number of rewards. The agent's task is to identify which plate holds the most rewards. This task assesses the agent's basic numeracy skills, specifically its ability to compare and evaluate quantities. Unlike other tasks, this task does not involve a dynamic component (i.e., a static scene).

Shape Causality The agent is tasked with finding a reward hidden beneath an object. The object subtly changes shape to accommodate the hidden reward. The agent's challenge is recognizing the correlation between the object's altered shape and the hidden reward's location. This task assesses the agent's ability to

understand causality, specifically how changes in an object's shape can indicate the presence of a hidden item.

As the thesis aims to evaluate the fundamental cognitive abilities of computational models and assess how well they generalize their learned knowledge, each task in the **SCOT** benchmark features varying levels of complexity. However, it's important to note that complexity is defined differently for each task; hence the ability to generalize may not increase linearly with complexity within a single task, nor is it directly comparable between different tasks. Table 3.1 explains the complexity variable for each task. For instance, a scene in the Gravity Bias task may contain anywhere from 1 to 8 tubes, increasing the task's difficulty. Figure 3-1 provides examples from the Gravity Bias and Relative Numbers tasks, illustrating how complexity varies within each task.

Task	**Complexity Variable**	**Difficulty Levels**
Addition	number of receptacles involved in the transfer	[1-4]
Gravity Bias	number of tubes in the scene	[1-8]
Rotation	number of receptacle rotations in the video	[1-8]
Simple Swap	number of receptacle swaps in the video	[1-8]
Relative Numbers	number of non-empty receptacles in the scene	[1-6]
Shape Causality	number of deformed receptacles	[1-3]

Table 3.1: **Task-Specific Complexity Definitions** Illustrates the variables that determine the complexity levels for each task in the **SCOT** benchmark. For every task, increasing the complexity variable makes the task more difficult, either by requiring more advanced reasoning capacities or extended memory.

3.3 Simulated Dataset Curation and Analysis

3.3.1 Dataset Generation Procedure

This section outlines the components and the pipeline of our approach used to generate the synthetic dataset for the proposed **SCOT** benchmark.

(a) Gravity Bias with 1, 3, 5 tubes respectively

(b) Relative Numbers with 1, 3, 5 engaged receptacles respectively

Figure 3-1: **Examples of Task-Specific Complexity Levels** Complexity for the gravity bias dataset is defined by the number of tubes in the scene. On the other hand, complexity is defined as the number of engaged (non-empty) receptacles for the Relative Numbers task.

We develop a procedural video generation pipeline using the Unity game engine to simulate the cognitive tasks. This pipeline can record videos of the simulated scenes and provide dense ground-truth information, including object poses, instance segmentations, and depth data. It also offers extensive customization options for each experiment, including the ability to randomize settings like colors, background objects, poses, and lighting and experimental design elements such as the number of rewards or receptacles. These options are task-dependent and can be exported as metadata.

For most tasks, custom environments were designed within the AI2Thor platform, which facilitates rendering by providing an API with a Unity renderer to communicate scene information and object movements. For the Gravity Bias task, we created a custom environment directly in Unity to facilitate the on-the-spot generation of tube

geometries. In both scenarios, we ensured strong supervision for each video by having the simulation platform provide comprehensive metadata. Researchers can utilize this metadata to train or evaluate computational models on various downstream tasks.

By obtaining strong supervision over the simulation environment, users can obtain detailed descriptions of the generated scenes for each frame. In particular, alongside RGB frames, the simulation platform is able to generate rich metadata, including (i) the world coordinates of each object in the scene, including the camera position and viewing direction; (ii) the physical attributes of each object in the scene, including color, size, and material; (iii) rendered depth images, instance segmentation masks, and category segmentation masks. The metadata is rich by design to support a variety of downstream tasks such as semantic segmentation, object identification, and pose estimation. For example, as detailed in Section 3.4, this benchmark expects models to make a single categorical prediction about the index or indices of the receptacles containing the most rewards. Hence, ground-truth labels can easily be derived from the metadata by comparing object positions. However, the pipeline accommodates other use cases as the metadata can be used to obtain ground-truth labels for other downstream tasks, such as object localization, prediction of compositional structures or scene graphs, or captioning for V&L training as in [14].

Furthermore, the pipeline was implemented in Python and offers a convenient interface to the simulation platforms, allowing users to control high-level scene attributes via a YAML configuration file. The configuration file can be used to adjust a wide range of parameters, including simulation parameters, such as camera intrinsic, lighting, colors, and textures; experimental variables, such as the count and types of receptacles and rewards; and task-specific variables, such as the count of occluders or tubes and the task complexity level. Figure 3-2 presents a high-level overview of our dataset generation procedure, highlighting the process from user-specified parameters to producing comprehensive scene data and annotations.

Additionally, our pipeline is designed to be accessible, requiring no prior knowledge of rendering or Unity, and is compatible with both MacOS and Linux platforms. The simulation platform source code is publicly available for building on unsupported

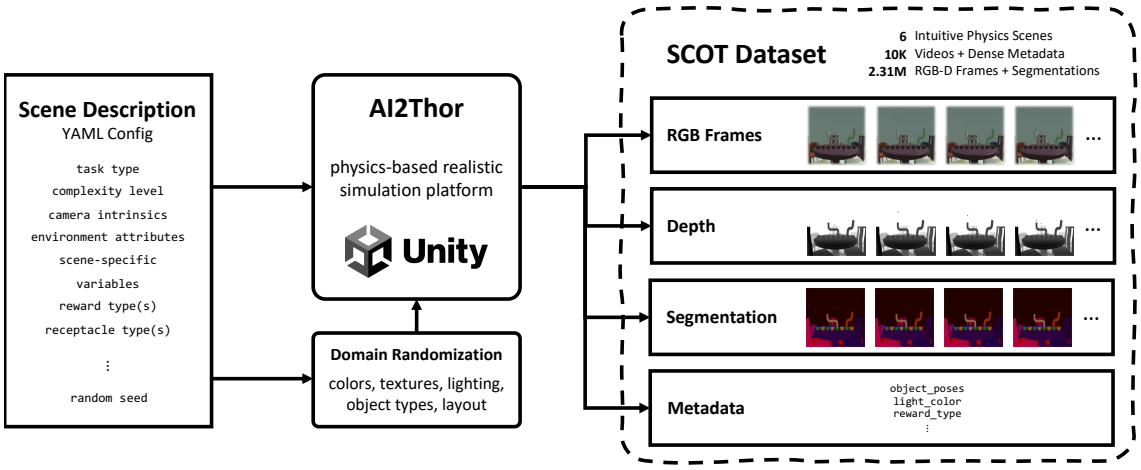


Figure 3-2: **Overview of the Dataset Generation Procedure** Users may specify the experimental and rendering parameters, along with a random seed, in a YAML config file. If any parameters are left unspecified, they are randomly selected from a predefined range using the provided seed, resulting in a comprehensive scene configuration. Accordingly, the simulator performs the task using this configuration, generating a sequence of RGB frames, depth images, instance segmentations, and dense annotations of the scene state.

operating systems. Moreover, the simulator's design supports headless operation, facilitating large-scale scene generation on high-performance computing clusters.

3.3.2 Dataset Bias Analysis

Using the proposed dataset generation pipeline, we employed Satori, a high-performance computing cluster at MIT, to generate a large-scale dataset of intuitive physics scenes. The contributed dataset encompasses a total of 10,000 videos, which collectively contain over 2.3 million RGB-D frames, along with their depth images, segmentation masks, and dense metadata. These videos span the six cognitive tasks, offering a diverse array of videos with highly randomized configurations for each task.

Leveraging domain randomization in the dataset generation pipeline ensures a robust and comprehensive representation of intuitive physics scenes. This technique randomizes the parameters of each video, such as the physical properties of objects, lighting, and camera field of view, thereby sampling from a uniform distribution across the entire parameter space. As a result, no specific condition is over-represented,

Task	# Videos	# Frames	# Conditions	# Videos per Condition
Addition	2,000	639K	4	500
Gravity Bias	400	89K	8	50
Rotation	2,000	476K	8	250
Simple Swap	2,000	1,065K	8	250
Relative Numbers	2,100	2.1K	6	350
Shape Causality	1,500	49.5K	3	500

Table 3.2: **Quantitative Overview of the SCOT Dataset** Presents the overall number of videos and frames generated for each simulated task.

mitigating potential biases. The value of domain randomization lies in its ability to generate a diverse array of scenarios and control for variables irrelevant to the simulated task, hence eliminating any favoritism towards learning scene attributes instead of physical properties on a higher level. This ensures a fair evaluation by exposing models to a wide variety of conditions, thereby reducing overfitting and improving their ability to handle unfamiliar situations.

Additionally, we use label distribution analysis, as visualized Figure 3-3, as a critical tool for assessing the balance and fairness of our dataset across all classes. In particular, label distribution analysis allows us to ensure a fair evaluation and robust training on the dataset by punishing models that favor any specific class due to overrepresentation or underrepresentation in the training data. The plot reveals that our dataset adheres closely to an ideal uniform distribution for each task, with a standard deviation $\leq 1.2\%$. This demonstrates the effectiveness of our data generation and sampling methods in maintaining class balance. A minor exception is observed in the Gravity Bias dataset, which exhibits a slightly higher variation of 1.75%. This deviation is likely due to the smaller size of this dataset of only 400 videos, and thus, minor fluctuations in label distribution can lead to a larger percentage deviation.

3.4 Benchmark and Metrics

Following the dataset generated in the previous section, we propose a benchmark that assesses the performance of computational models on the Simulated Cognitive

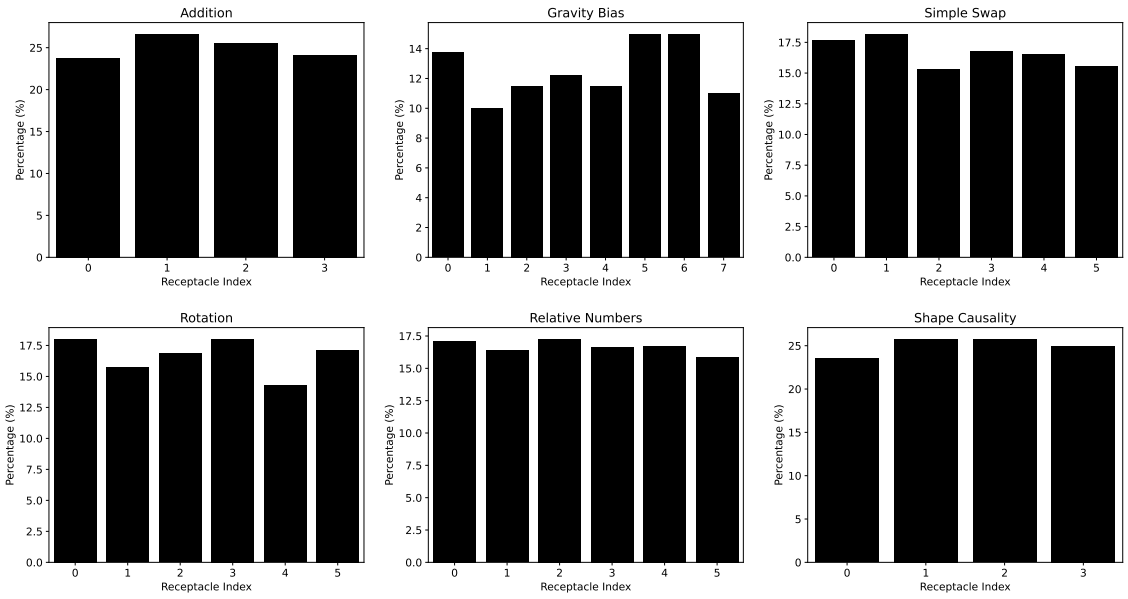


Figure 3-3: **Label Distribution Bias Analysis** Our analysis of label distribution within the dataset reveals a well-balanced representation across all classes. Each label exhibits a roughly equal likelihood of occurrence, indicating that no single class is significantly over-represented or underrepresented. This balanced distribution mitigates the risk of model bias towards any particular class.

Tasks (**SCOT**) dataset. In particular, we pose scene perception as a classification problem where an agent observing the scene makes a categorical prediction about the index or indices of the receptacle(s) that contain the highest number of rewards. In particular, given a sequence of frames $\{I_1, I_2, ..., I_T\}$, each containing $H \times W$ pixels, from a scene containing k receptacles, we model the agent as a function M that estimates a posterior PMF over the k receptacles, where each probability corresponds to the likelihood that a given receptacle contains the highest number of rewards.

$$M : \mathbb{Z}^{(H \times W \times \times 3) \times T} \rightarrow \left\{ P : [1-k] \in \mathbb{Z} \rightarrow [0,1] \;\middle|\; \sum_{i=1}^{k} P(i) = 1 \right\}$$

We use task complexity as a proxy to evaluate the generalizability of an agent's intuitive physics core knowledge. However, as complexity parameters differ among tasks, we neither expect a linear relationship within a specific task nor direct comparability between tasks. Thus, we view complexity as a qualitative indicator, focusing our comparisons on individual task levels rather than across tasks. We evaluate the

performance of artificial agents using a range of metrics, including categorical classification accuracy, mean absolute error (MAE), precision, recall, and F1 score. Each metric offers a unique insight into the agent's performance.

Below, we describe and formulate each metric. Let n be the total number of videos and P_i be the PMF produced by the agent after observing the i-th video. Accordingly, the most-likely class for that video is $\hat{y}_i = \underset{j \in [1,k]}{\operatorname{argmax}} P_i(j)$, and the true class is y_i.

Classification Accuracy is the ratio of correct predictions to the total number of predictions. This metric provides a fundamental assessment of the predictive performance.

$$\text{Classification Accuracy} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(\hat{y}_i = y_i)$$

Mean Absolute Error (MAE) computes the absolute difference between the predicted and the actual receptacle index. This metric is particularly useful as it indicates the model's predictive accuracy in numerical terms, helping us understand the magnitude of the error.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$

Precision and Recall deliver a more detailed view of the model's performance for each class. Precision calculates the proportion of true positive predictions among all positive predictions for each class, whereas Recall measures the proportion of true positive predictions out of all actual positives for each class. Since tasks involve multiple classes, we compute the average (macro) precision and recall.

$$\text{Average Precision} = \frac{1}{k} \sum_{j=1}^{k} \left(\frac{\sum_{i=1}^{n} \mathbb{1}(y_i = j \text{ and } \hat{y}_i = j)}{\sum_{i=1}^{n} \mathbb{1}(\hat{y}_i = j)} \right)$$

$$\text{Average Recall} = \frac{1}{k} \sum_{j=1}^{k} \left(\frac{\sum_{i=1}^{n} \mathbb{1}(y_i = 1 \text{ and } \hat{y}_i = 1)}{\sum_{i=1}^{n} \mathbb{1}(y_i = 1)} \right)$$

F1 Score is the harmonic mean of precision and recall, providing a comprehensive

measure for imbalanced datasets. Given that some tasks might result in a slightly-uneven distribution of labels, the F1 Score ensures that the model's performance is thoroughly evaluated. Denoting p_j and r_j to refer to the precision and recall, respectively, of class j,

$$F1_{\text{macro}} = \frac{1}{k}\left(\sum_{j=1}^{k} 2 \cdot \frac{p_j \cdot r_j}{p_j + r_j}\right)$$

Integrating the metrics above allows for a holistic evaluation of the model's performance, thereby providing a comprehensive understanding of its strengths and areas for improvement. We believe that our benchmark should serve not only as a measure of the agent's performance but also help identify specific strengths and weaknesses of the different computational models. The combination of these metrics addresses both the overall accuracy and the quality of the predictions for each class, enabling a thorough evaluation of the model's ability to generalize from training to unseen scenarios.

Hence, interpretability plays a crucial role in the evaluation. An agent that performs well quantitatively but whose internal workings are a "black box" may prove less valuable than a less performant but interpretable model. Therefore, the benchmark also considers the interpretability of the model's reasoning, adding another dimension to our evaluation. We do not specify a particular qualitative metric that can be used for qualitative evaluation beyond complexity levels as model designs can vary significantly. Nonetheless, we recommend developing models whose decisions can be efficiently explainable as it facilitates the iterative development of those models. In addition, we take a step to ensure that no overfitting is achieved by strongly controlling non-relevant scene attributes through domain randomization.

Chapter 4

Deep Learning Baselines

4.1 Chapter Overview

To facilitate the testing of computational models on the **SCOT** benchmark, we implement a general system through which artificial neural models can interact with the **SCOT** benchmark. This system is further described in Section 4.2 and is used to train and evaluate several baseline deep learning models on the **SCOT** benchmark. We present evaluation results and briefly discuss the limitations of those neural models on our intuitive physics dataset (Section 4.3).

4.2 Experimental Setup

Inspired by recent work in model and benchmark separation [89], the system, at its core, develops an abstraction that separates model specification from the datasets and inference procedure.

In particular, we develop a "Training Job" abstraction that streamlines the process of feeding data into the models, training the models if necessary, carrying out inference, and evaluating the results based on the defined metrics. This abstraction, therefore, ensures a clear and coherent workflow, reducing the overhead often associated with linking deep learning models to efficient data loaders and inference procedures. Figure 4-1 illustrates the different components of the system and how they interact with each other.

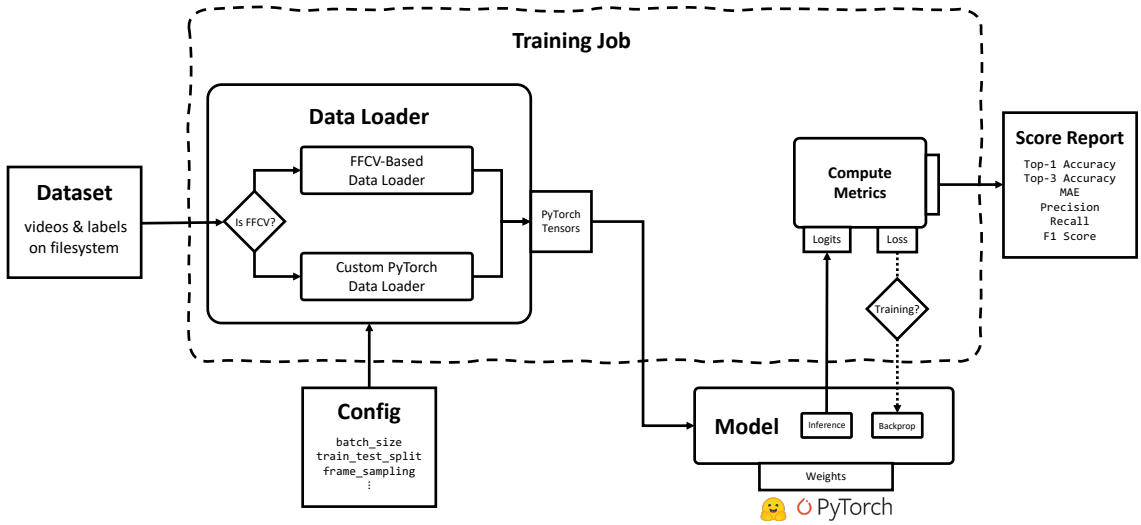


Figure 4-1: **Baseline Evaluation System Diagram** The intuitive physics dataset, consisting of videos and labels, is loaded from the file system and transformed into PyTorch tensors via the appropriate data loader. The model, either implemented by the user or chosen from a pre-implemented list, performs inference. Following that, a procedure for computing evaluation metrics is carried out. If training is enabled, the cross-entropy loss is computed and used for backward propagation. Lastly, an evaluation report is produced with scores for each metric. Each component of the system may be customized through configuration files.

4.2.1 Model Descriptions

Using this system, we have implemented two classes of models that can be tested on the **SCOT** benchmark: (i) LSTM models with visual encoder backbones and (ii) specialized video understanding models from HuggingFace [106]. We conduct a systematic evaluation of example models from both classes. Here, we describe each class and list the models we used for evaluation.

Visual Encoder-LSTM Models are constructed from three primary components: a visual encoder, an LSTM network, and a multilayer perceptron for classification. The visual encoder backbone, which can be a convolutional neural network (CNN) such as ResNet or DenseNet, or a transformer model like Vision Transformer (ViT-B16), processes the visual aspects of the scene to extract frame-level features. This includes recognizing spatial patterns, objects, and scene composition based on individual pixels, resulting in a vector representation for

each frame. Subsequently, a two-layer LSTM model is employed to encode temporal information from frame-level features using attention mechanisms. The last component, a multilayer perceptron, takes the hidden temporal features from the last LSTM layer and maps them into a distribution over the indices related to the cognitive task. We source the CNN and ViT models and obtain their weights using Torchvision.

Video Understanding Models are specifically designed for and pre-trained on video data. These models include TimeSformer, VideoMAE, and XClip, all of which have demonstrated a unique capacity to extract meaningful features for video classification. We remark that XClip is inherently multimodal, designed to work with both visual and textual data; hence we only use the embeddings from the visual encoder following the cross-frame attention mechanism. Like the CNN-LSTM models, we employ a multilayer perceptron to map the processed visual embeddings into a probability distribution across possible receptacle indices. In our system, we source video understanding models and obtain their weights using Huggingface.

4.2.2 Implementation Details

The developed system operates on top of PyTorch [75], a versatile and widely-adopted open-source machine-learning library in Python. Accordingly, any model implemented in Python or with a Python interface can be integrated into the system. All model constructions and weights described in 4.2.1 are loaded using PyTorch, and both the training and inference pipelines adhere to standard PyTorch implementations. Similarly, the system allows the integration of various pre-trained models from the Huggingface repository, including VideoMAE, TimeSformer, and XClip.

Moreover, an efficient data-loading pipeline was implemented that uses FFCV [64] to mitigate memory bottlenecks during data loading. In particular, we (optionally) provide a version of the dataset in FFCV format, which allows users to use efficient file storage format, caching, pre-loading, and other techniques to speed up data loading, ensuring full memory utilization. As a result, model training becomes much more

efficient, a valuable attribute considering the data-intensive nature of video datasets used in the system. We remark that using FFCV accelerates training on V-100 GPUs twofold.

Furthermore, by abstraction, the system emphasizes configuration flexibility and extensibility. Specifically, users can adjust the attributes of the data loading pipeline and model specifications. For example, for the pre-implemented CNN-LSTM baseline model, users can change the CNN architecture, number of LSTM layers, and hidden size of the LSTM block. Additionally, users can develop their configuration files for new models, expanding the system's adaptability.

4.3 Baseline Evaluations

We evaluate the models from 4.2.1 on the **SCOT** benchmark using Supercloud [82], an MIT high-performance compute cluster. We allocated one V100 GPU for each job involved in the training and evaluation of the models.

The datasets employed in this study were sourced from the **SCOT** video library presented in Chapter 3. We divided the video dataset for each task into two distinct subsets - a training/validation subset and a testing subset. The complexity level definitions guided the partitioning process, allocating the smaller $\lfloor {}^{2}/_{3}\, n \rfloor$ of the n complexity levels for training/validation and the remaining for testing. For example, the gravity bias task has eight complexity levels, so we used the first five (1-5 tubes per scene) for training and the last three (6-8 tubes per scene) for testing.

Within the training/validation subset, we randomly sample 80% of the dataset to fine-tune the neural models and use the remaining 20% for validation.

Accordingly, we fine-tuned pre-trained versions of the neural models, sourced from Torchvision and Huggingface, for 100 epochs or until the validation loss monotonically increased for five consecutive epochs. We define an "epoch" as a full pass through the fine-tuning dataset. Subsequently, we select the model weights (checkpoint) at the epoch with the lowest categorical cross-entropy loss on the validation dataset to perform inference on the testing dataset.

The training jobs utilized a linear learning rate schedule and the Adam optimizer,

as per [55]. The linear learning rate schedule gradually decreased the learning rate throughout the training. Meanwhile, the Adam optimizer, known for its adaptive learning rates and efficient memory usage, facilitated the optimization process.

To identify the most effective parameters for the optimizer, we conducted a comprehensive hyperparameter tuning process using the Weights & Biases Sweeps [11]. This sweeping tool allowed us to explore various combinations of hyperparameters and identify the configuration that yielded the best performance for each model.

4.3.1 Experimental Results

Following fine-tuning, we evaluate the models on the held-out testing dataset using top-1 accuracy, macro-average F1 score, and mean absolute error. Table 4.1 compares the six deep learning models' performance on each task of the **SCOT** benchmark. For each of the six models assessed, two values are presented: one for the validation dataset and one for the testing dataset. The validation dataset is the randomly selected 20% subset of the videos with low complexity, unseen during the training phase but used to select the best model weights for inference. In contrast, the testing dataset comprises videos of higher complexity, also unseen during the training phase.

We observe that XClip and TimeSformer consistently outperform the other models in top-1 accuracy and average F1 scores across most tasks. In contrast, despite its memory-augmented mechanism, VideoMAE did not reach the performance levels of TimeSformer and XClip on most tasks. This result suggests that VideoMAE's model architecture may face challenges encapsulating all the necessary information from multiple video clips.

On the other hand, mean absolute errors offer an alternative perspective on the models' performance. In this metric, lower values are desirable as they signify smaller errors. In particular, it measures *how far* the correct receptacle is from the predicted receptacle, on average, assigning lower errors to spatially closer predictions and aligning more with the intuitive definition of the benchmark's objective. Once again, TimeSformer and XClip perform better than other models, suggesting their predictions are generally proximate to the ground truth.

	Addition		Gravity		Rotation		Swap		Relative		Shape	
Model	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test
ResNet-18/LSTM	26.40	25.91	38.00	16.00	22.16	19.07	16.96	16.40	59.86	28.00	98.02	71.20
Densenet/LSTM	27.93	25.39	44.40	18.67	22.16	19.07	16.96	16.40	17.64	16.00	97.40	66.20
ViT-B16/LSTM	25.89	27.85	15.20	12.00	22.16	19.07	17.52	17.87	47.14	21.71	99.70	80.95
VideoMAE	31.19	27.14	18.00	14.67	24.24	18.93	20.40	13.20	96.43	54.57	100.0	89.70
TimeSformer	74.21	49.42	97.20	**52.67**	68.72	27.73	52.96	**17.87**	97.36	**72.71**	100.0	**100.0**
XClip	67.18	**52.35**	94.80	43.33	76.24	**28.00**	41.60	14.93	96.86	65.71	100.0	**100.0**

(a) Top-1 Accuracy (%)

	Addition		Gravity		Rotation		Swap		Relative		Shape	
Model	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test
ResNet-18/LSTM	18.27	16.62	36.43	14.90	6.04	5.34	4.83	4.70	59.71	27.86	97.81	69.84
Densenet/LSTM	10.92	10.12	42.01	15.96	6.04	5.34	4.83	4.70	5.00	4.59	97.15	65.29
ViT-B16/LSTM	16.72	17.40	5.50	5.50	6.04	5.34	4.97	5.05	42.92	19.56	99.66	78.81
VideoMAE	23.17	18.18	6.91	6.54	12.34	9.83	14.59	9.46	96.40	54.84	100.0	87.77
TimeSformer	73.92	49.26	96.48	**46.82**	61.84	21.98	51.51	**17.23**	97.33	**71.59**	100.0	**100.0**
XClip	66.76	**51.92**	93.86	39.51	72.81	**22.52**	41.01	13.30	96.78	64.85	100.0	**100.0**

(b) Average F1 Score (%)

	Addition		Gravity		Rotation		Swap		Relative		Shape	
Model	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test	Val.	Test
ResNet-18/LSTM	1.14	1.06	1.62	2.31	1.44	1.49	1.52	1.52	0.90	1.71	0.02	0.29
Densenet/LSTM	0.96	0.99	1.62	2.37	1.44	1.49	1.52	1.52	2.46	2.48	0.03	0.34
ViT-B16/LSTM	0.96	0.97	2.20	2.27	1.44	1.49	2.47	2.37	1.37	1.94	0.00	0.19
VideoMAE	0.98	1.01	2.47	2.37	1.45	1.61	2.24	2.41	0.07	1.00	0.00	0.10
TimeSformer	0.40	0.77	0.07	**0.67**	0.46	1.36	1.08	**1.97**	0.07	**0.71**	0.00	**0.00**
XClip	0.55	**0.75**	0.12	1.26	0.38	**1.30**	1.52	2.09	0.07	0.80	0.00	**0.00**

(c) Mean Absolute Error

Table 4.1: **Performance of Deep Learning Models on the SCOT Benchmark** Illustrates the top-1 accuracy, macro F1 score, and mean absolute error achieved by each baseline neural model on both the validation and test subsets of each task within the benchmark. The validation subset is a random selection comprising 20% of the videos with low complexity not seen during training, while the test subset contains the videos with higher complexity levels. The best test performance on each task is highlighted in bold text.

Additionally, Figure 4-2 provides a more comprehensive summary of each model's performance on the datasets as a function of the dataset's complexity levels. We observe a consistent pattern of decreasing model performance, as indicated by lower accuracy, lower F1 score, and higher MAE scores as the complexity level increases. The results suggest that the ability of the models to generalize is reduced when they are exposed to scenes with more complex dynamics.

While we remark that this pattern may result from overfitting on the complexity

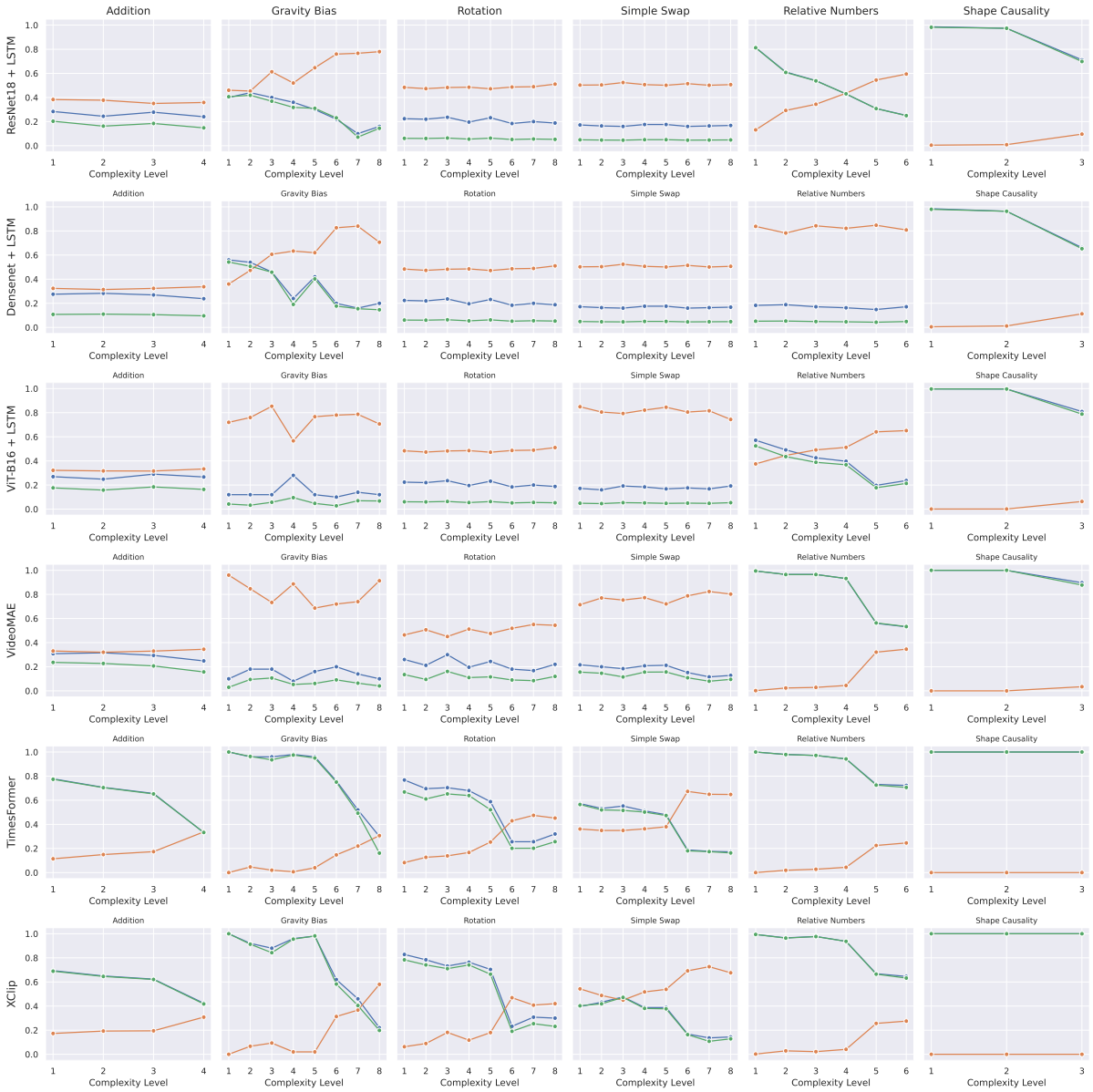


Figure 4-2: **Performance Comparison of Models Across Datasets Complexity Levels** Presents a comparative performance evaluation of various machine learning models across multiple datasets, depicted in terms of three key metrics: Accuracy(blue), Mean Absolute Error (orange) (scaled by $1/3$ for better visualization), and the F1 Score (green). The metrics are plotted against increasing complexity levels. Each row corresponds to a distinct dataset, and each column represents a different machine-learning model.

levels of the training data, we still observe a performance decrease among the held-out testing complexity levels for most models. However, this pattern does not hold uniformly across all models and datasets. Some models demonstrate a stable or

slightly improved performance with higher complexity levels, potentially due to their mainly incorrect and noisy predictions.

Finally, we use Grad-CAM [90, 37] visualizations to provide qualitative insights into the interpretability of the neural baseline models. Specifically, we interpret explanations of correct classifications by visualizing the feature map of the last convolutional block in the ResNet-18 model and the spatial attention maps following the last self-attention layer in TimeSformer. Those visualizations are presented in Figure 4-3.

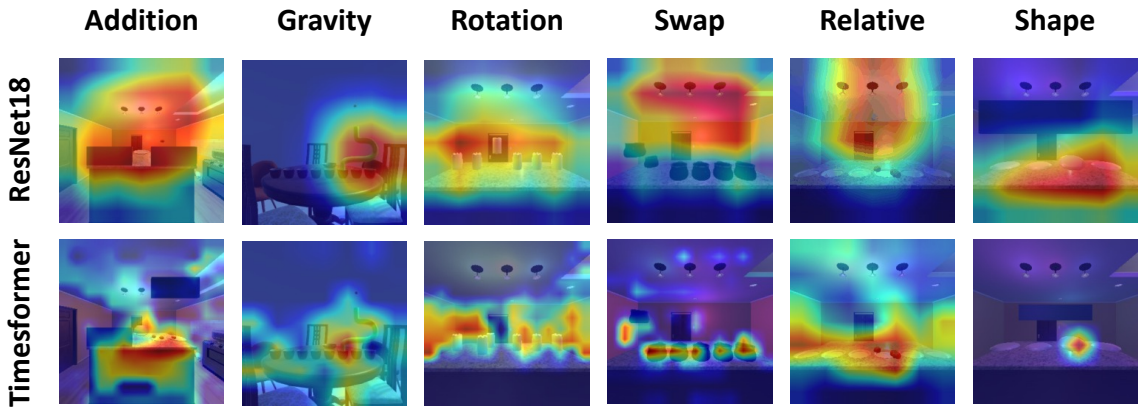


Figure 4-3: **Qualitative Evaluation of the Feature Maps of Neural Baselines** Shows Grad-CAM visualizations of the ResNet-18/LSTM and TimeSformer models on sample frames from each task in the **SCOT** dataset.

4.3.2 Discussion

The results provide observations about the capabilities and limitations of deep learning models in generalizing intuitive physics knowledge.

The first insight is the contrast between the average performance of deep learning models when recognizing spatial attributes of scenes versus understanding objects and numbers in the visual world. In particular, in tasks requiring spatial awareness, such as Gravity Bias and Shape Causality, we observe that models displayed high competence in identifying the correct receptacles. For instance, TimeSformer achieves 52.67% accuracy on the high-complexity scenes of the Gravity Bias test set, significantly higher than the expected accuracy of a random classifier of 12.5%. Similarly, TimeSformer and XClip achieve near-perfect classification accuracies on the entire Shape Causality video dataset.

However, models perform at or slightly above chance levels on tasks that require developing *concepts* of objects and keeping track of them under occlusion, such as the Rotation and Swap tasks. For example, in the Simple Swap and Rotation datasets, the highest top-1 classification accuracies on the test set are 17.23% and 22.52%, respectively, slightly higher than the expected accuracy of a random classifier of 16.7%.

This finding suggests that the emphasis of current deep learning algorithms on pattern recognition and feature extraction enables the models to identify objects, their shapes, and their spatial proximity fairly well. This competence may be attributed to the large amounts of data that deep learning models were pre-trained on, which require understanding visual attributes like shape and orientation as they are common across various scenes in nature. Hence, those models are successful in tasks such as image classification and object detection.

Nonetheless, deep learning models are not required to form abstract concepts of objects. Therefore, we observe a performance drop if a task involves concepts and processes that do not have clear statistical patterns or that require abstraction and high-level cognitive processing. The difficulty with quantity understanding and object conceptualization represents an example of this bottleneck in deep learning. It suggests a limitation in the ability of models to generalize their learning to more abstract or dynamic concepts, reflecting a shortcoming in dealing with tasks that require more than just pattern recognition.

Furthermore, the Grad-CAM visualizations in Figure 4-3 reinforce our conclusions about the capabilities and limitations of deep learning models. These visualizations show that the models can effectively identify the relevant scene components in tasks like Gravity Bias and Shape Causality, demonstrating a firm grasp of low-level geometric features. However, in tasks such as Addition and Simple Swap, which demand a deeper understanding of scene composition, the models exhibit less-focused attention across all objects and surroundings, implying difficulties in comprehending complex scene dynamics. Hence, these visualizations corroborate the performance metrics, providing qualitative evidence of the models' abilities and challenges in handling different intuitive physics tasks.

Another insight from the experimental results is that as the complexity variable increases, we observe a drop in performance in most tasks. This generalization gap implies that models, while adept at making predictions under lower complexity levels, fail to effectively extrapolate the physical and compositional principles when exposed to increased complexity.

For example, the accuracy of TimeSformer and XClip, the highest-performing models on the Gravity Bias dataset, monotonically drops as the number of tubes in the videos increase. This failure suggests another limitation in the models' capacity to internalize task-specific rules or logic at the level of objects, resorting instead to reliance on the learned patterns. This also highlights the need for advancements in computational modeling to facilitate pattern abstraction and derivation of generalized insights under high-complexity conditions. This might necessitate the integration of robust understanding mechanisms, such as symbolic reasoning or cognitive architectures, that can supplement the pattern recognition capabilities of these models.

In addition, the experiments suggest that the difficulty in complex tasks is not merely a matter of insufficient training data or computing power. The models struggled with the more complex Simple Swap and Rotation tasks even when fine-tuned on a sizeable amount of data (e.g., 1M+ frames for Simple Swap, 476K frames for Rotation). This points to a deeper issue that cannot be resolved by simply training models on more data. In particular, it suggests that scaling up current approaches should take the form of new methods that equip models with a more structured understanding of the world beyond just pattern recognition. The goal should be to develop models capable of achieving high performance under data-sparse conditions and effectively generalize from limited, low-complexity scenarios to unfamiliar, high-complexity ones.

Integrating Bayesian inverse graphics with deep learning could address some limitations in understanding complex physics and generalizing at scale. This approach provides a structured understanding of scene dynamics by inferring scene properties from observed images. Deep learning can help guide the inference process by, for instance, enabling the extraction of low-level features or helping identify and segment

objects from their basic observed properties. This combination could offer a scalable approach to intuitive physics tasks, enabling explicit modeling of physical processes and improving model generalizability.

Finally, while the range of models examined here is not exhaustive, we believe additional video-understanding models should be assessed. We design the system used for training and inference to facilitate examining additional models. It is intended to enable the iterative development of techniques that increase the generalizability of learned knowledge in AI agents.

Chapter 5

Bayesian Generative Model

5.1 Chapter Overview

Inspired by the intuitive physics engine framework [9], we develop a Bayesian probabilistic generative model that builds upon previous work in generative inverse graphics and Bayesian inference, specifically 3DP3 [39]. Our model encodes hierarchical priors about object dynamics and relations, leverages enumerative tracking to generate object pose hypotheses, and employs a noise model on depth data to estimate the likelihood of the generated hypotheses following observed depth images.

This chapter describes the model and its components (Section 5.2), presents an evaluation of the model on the **SCOT** benchmark (Section 5.3), and briefly discusses its advantages and limitations.

5.2 Model Description

We implement a probabilistic generative model that 1) infers the physical meshes and initial poses of each object in the scene (5.2.2); 2) tracks object poses and relations over time (5.2.3); and 3) draws task-specific classification inferences based on the final inferred scene graph and object poses (5.2.4). Figure 5-1 presents a high-level overview of the components of the model and the overall inference procedure.

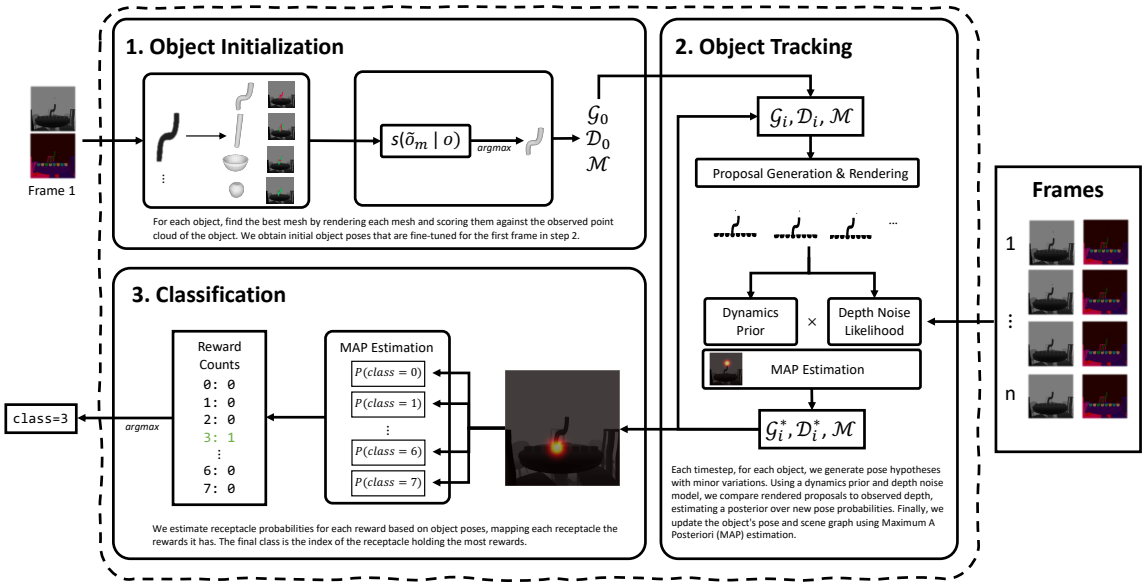


Figure 5-1: **Overview of the Bayesian Generative Model** The model begins by identifying object meshes and estimating their initial poses. It then tracks these objects, updating their positions and relationships in a dynamic scene graph. Finally, it estimates a distribution of the final class over the possible receptacle indices. This figure provides a step-by-step visualization of the process, from initial object identification to final classification.

5.2.1 Preliminaries

We formulate the classification task posed by the **SCOT** benchmark as a Bayesian inverse graphics problem searching for the scene representation that, when rendered through a simulator, explains the observed depth images. Concretely, we define a scene (video) as a series of T frames (images) $\mathbf{I}_1, ..., \mathbf{I}_T$, where $\mathbf{I}_i \in [\![0, 255]\!]^{H \times W \times 3}$. Each frame I_i contains a collection of N_i objects with their corresponding mesh classes $\mathcal{M}_i = [m_1, ..., m_{N_i}]$, where $m_1, ..., m_{N_i} \in \{1, ..., M\}$ and 6DoF poses $\mathcal{D} = [P_1, ..., P_{N_i}]$, where $P_1, ..., P_{N_i} \in \mathbb{SE}(3)$, represented as 4×4 spatial transformation matrices.

In addition to the RGB image $\mathbf{I}_i$, each frame i has an associated depth image $\mathbf{D}_i \in \mathbb{R}^{H \times W}$ and instance segmentation map $\mathbf{S}_i \in \{1, ..., N_i\}^{H \times W}$. A segmentation map, $\mathbf{S}_i$, maps each pixel in $\mathbf{I}_i$ to an object from $\{1, ..., N_i\}$.

At time i, the model maintains a state (scene representation) $\mathcal{S}_i$ of object poses, $\mathcal{D}_i, ..., \mathcal{D}_{i-w}$, where w is the number of past poses to use for physics inference (such

as velocity estimation) and scene graph, $\mathcal{G}_i$.

The scene graph encodes objects as nodes and semantic relations between those objects, specifically occlusion and containment, as directed edges. On a high level, containment enforces positional constraints on object movements: if an object o_i contains another object o_j, then any changes to the world coordinates of o_i apply to o_j. Similarly, o_j may not move beyond the bounds of o_i unless containment is broken. Additionally, an occluded object can assume any pose behind its occluding object; thus, the proposed posterior of the occluded object's translations is primarily guided by the physical priors.

5.2.2 Inferring Object Meshes and Poses from Depth Data

For a given scene, we initialize $\mathcal{D}_0$ using the ground-truth segmentation image $\mathbf{S}_1$ and the depth map $\mathbf{D}_1$ of the first frame, produced by the physical simulator (Unity). For each object $o \in \{1, ..., N_1\}$, we first construct an object point cloud $\mathbf{o}$ by collecting its depth points $\mathbf{D}_{1[\mathbf{S}_1=o]}$ (i.e. depth points in $\mathbf{D}_1$ corresponding to o's pixels in $\mathbf{S}_1$). Then, we compute a matching score s between the observed object point cloud $\mathbf{o}$ and each mesh model $m \in \{1, ..., M\}$ by rendering the mesh at the center of the object point cloud to get a hypothesis $\tilde{\mathbf{o}}_\mathbf{m}$ and counting the number of mutually-unexplained points in $\mathbf{o}$ and $\tilde{\mathbf{o}}_\mathbf{m}$.

$$s(\tilde{\mathbf{o}}_\mathbf{m}, \mathbf{o}) = P(\tilde{\mathbf{o}}_\mathbf{m}|\mathbf{o}) = \frac{\sum_{(i,j)} \mathbb{1}[\tilde{\mathbf{o}}_{\mathbf{m}(i,j)} \cdot \mathbf{o}_{(i,j)} > 0]}{\sum_{(i,j)} \mathbb{1}[\mathbf{o}_{(i,j)} > 0]}$$

Figure 5-2 visualizes how different mesh models compare to the observed point cloud of the tube.

Thus, we infer that the best model m for object o is $\underset{m}{\operatorname{argmax}}\, s(\tilde{\mathbf{o}}_\mathbf{m}, \mathbf{o})$. Accordingly, we approximate the spatial transformation matrix of object o as the translation matrix where the rotation component is an identity matrix and the translation vector corresponds to the center of the rendered point cloud $\tilde{\mathbf{o}}_\mathbf{m}$.

Following the initial approximation of $\mathcal{D}_0$, we run one tracking step (Section 5.2.3) to correct for minor translation and rotation discrepancies between the approximate poses and the initial point clouds, obtaining the object poses in the first frame $\mathcal{D}_1$.

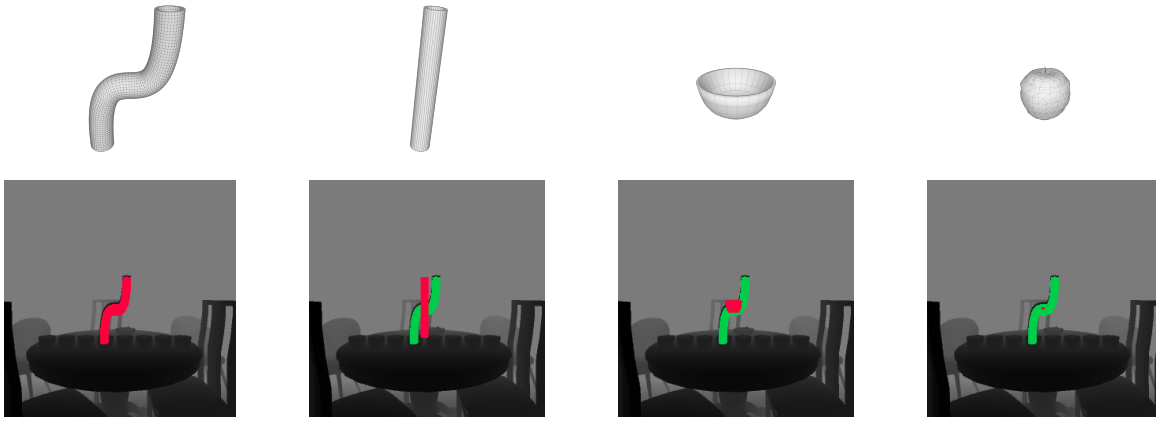


Figure 5-2: **Example: Inferring Correct Object Mesh from Observed Point Cloud** The observed point cloud of the tube is highlighted in green. We illustrate how when example meshes are rendered at the center of the point cloud, as shown in red, they provide varying explanations of the point cloud. We infer that the best object mesh is the one that maximizes $s(\tilde{\mathbf{o}}_\mathbf{m}, \mathbf{o})$.

Upon the initialization of the scene, we infer containment and occlusion relations through an exhaustive pairwise comparison of the bounding boxes of objects. Concretely, o_i is *contained by* o_j if the bounding box of o_i lies entirely within that of o_j. Similarly, o_i is *occluded by* o_j if, when rendering o_j as $\tilde{\mathbf{o}}_\mathbf{j}$ and both o_j and o_i as $\tilde{\mathbf{o}}_\mathbf{ij}$, we find that $\big(\sum_{(i',j')} \mathbb{1}[\tilde{\mathbf{o}}_{\mathbf{ij}(i',j')} \neq \tilde{\mathbf{o}}_{\mathbf{j}(i',j')}] \big) \leq \epsilon_o$, where ϵ_o is an occlusion threshold of the number of non-occluded pixels. Thus, we obtain the state of the first frame as $\mathcal{D}_1$ and $\mathcal{G}_1$. For our experiments, we use scenes of 300×300 pixels and an occlusion threshold $\epsilon_o = 10$.

5.2.3 Object Tracking: Enumerative Proposal Generation

Tracking object movements over time accurately and efficiently proves challenging in continuous 3D space. Therefore, following [111], we convert the object pose space into discrete voxel units of size (d_x, d_y, d_z) following the camera parameters. Given an object o's pose P_o, we create an array of pose hypotheses with minor variations in translation and rotation. For translation, we grid the $7 \times 7 \times 7$ voxel space surrounding the object's center and enumerate proposals where the object's center is located at each voxel in the grid. Similarly, we generate rotation proposals by sampling 64 orientations from a tightly-clustered von Mises–Fisher spherical distribution centered

around the object's rotation. Thus, we consider $k = 343 \times 64$ pose hypotheses, the Cartesian product of the translation and rotation proposals.

Consequently, we identify the proposal that best describes the observed depth data of a frame i following a Bayesian procedure that uses 1) physical priors about free object movements and hierarchical priors about pose changes under containment and occlusion; and 2) a likelihood model based on 3DP3 that encodes noise in the depth information. Specifically, for each object, we render the pose hypotheses (while fixing the poses of all other objects in the scene) as depth images $\tilde{\mathbf{c}}_1, ..., \tilde{\mathbf{c}}_k$ and approximate a posterior distribution $P(\tilde{\mathbf{c}}_j|\mathbf{c})$ over rendered hypotheses c_j as the product of the dynamics prior and the depth likelihood models:

$$P(\tilde{\mathbf{c}}_j|\mathbf{c}; \mathcal{S}_i, p, V, \sigma) \sim P(\tilde{\mathbf{c}}_j; \mathcal{S}_i) P(\mathbf{c}|\tilde{\mathbf{c}}_j; p, V, \sigma)$$

where p, V, σ are likelihood parameters and $\mathcal{S}_i$ is the scene representation at time i.

This approximation is valid for estimating the Maximum A Posteriori (MAP) as the marginal likelihood $P(\mathbf{c})$ does not depend on the state of the model or the likelihood parameters and hence does not affect the MAP.

Prior We select a prior that favors intuitive physical dynamics in $\mathcal{D}_i$, such as gravity and velocity, as well as consistent semantic relations in $\mathcal{G}_i$. That is, for a set of rendered proposals $\tilde{\mathbf{c}}_1, ..., \tilde{\mathbf{c}}_\mathbf{k}$ targeting the pose of object o, the prior distribution resembles a 3-dimensional Gaussian distribution with identity covariance centered at the position vector $P^*_{(o)_i}$, which is the position of object o following a dynamics model $D : \mathbb{R}^{w \times 3} \to \mathbb{R}^3$. The dynamics model heuristically predicts the new pose of an object given the object's current pose and its w past poses by applying a gravity translation and approximating the object's translational velocity as the exponentially-weighted moving average of pose deltas over the preceding w poses.

$$P^*_{(o)_i} = D(P_{(o)_i}, ..., P_{(o)_{i-w}}) = P_{(o)_i} + \underbrace{[0\ g\ 0]^T}_{\text{gravity shift}} + \underbrace{\sum_{j=0}^{w-1} \alpha(1-\alpha)^j (P_{(i-j)} - P_{(i-j-1)})}_{\text{velocity shift}}$$

where g is the translation shift caused by gravity, α is the recency weight pa-

rameter of the moving average, and w is the number of past poses to consider for the velocity shift. Empirically, we determine $g = 0.2$, $\alpha = 0.15$, $w = 5$.

Accordingly, the prior distribution over the pose proposals targeting object o, whose proposed new pose in $\tilde{\mathbf{c}}_j$ is $\tilde{P}_{(o)_j}$, is

$$P(\tilde{\mathbf{c}}_j; \mathcal{S}_i) = P(\tilde{P}_{(o)_j}; \mathcal{S}_i) = \mathcal{N}(\tilde{P}_{(o)_j}; P^*_{(o)_i}, I_3)$$

where the covariance of the Gaussian distribution is I_3, the 3×3 identity matrix.

Likelihood We use a likelihood function acting as a noise model on observations, estimating the likelihood of the observed depth image given the rendered images (from object pose hypotheses). It incorporates a Gaussian noise component around non-outlier points and a uniform outlier component. Outliers refer to data points in the observation that significantly deviate from the expected renderings. This is particularly relevant in 3D scenes where factors like noise, occlusions, and environmental conditions can cause depth points to differ from the object models.

The Gaussian noise component accommodates small deviations between observed and rendered points, accounting for uncertainties in rendering. This noise is assumed to follow a Gaussian distribution centered around each rendered point. Additionally, the outlier component accommodates observed points that do not align with the expected data using a uniform distribution over a 3D bounding volume V. Thus, the likelihood is expressed as follows:

$$P(\mathbf{c}|\tilde{\mathbf{c}}; p, V, \sigma) = \prod_{k}^{K} \left(p \cdot \frac{1}{V} + (1-p) \cdot \frac{1}{\tilde{K}} \sum_{\tilde{k}}^{\tilde{K}} \mathcal{N}(\mathbf{c}_k; \tilde{\mathbf{c}}_{\tilde{k}}, \sigma^2 \cdot I_3) \right)$$

Here, $\mathbf{c}$ and $\tilde{\mathbf{c}}$ refer to the latent observed and rendered depth points, respectively. p is the a priori probability of a point being an outlier. V is the volume of the 3D bounding space where an outlier could be found. σ characterizes the Gaussian noise we anticipate around each rendered point, and I_3 is the 3×3 identity matrix. K and $\tilde{K}$ correspond to the number of latent points in the

observed and rendered point clouds, respectively. We empirically determine the likelihood parameters to be $p = 0.05$, $\sigma = 0.1$, and $V = 10^3$ to provide effective estimation for our specific tasks.

Due to the computational cost of calculating this likelihood for all combinations of observed and rendered points, the algorithm approximates this calculation by only considering a subset of terms. We assume that faraway points in the depth map do not significantly affect the likelihood estimation for the current point. Therefore, we select this subset based on proximity, including only those points within a certain filter size around the current point.

To ensure that the posterior maps to a proper probability distribution, we remark that the prior is defined as a Gaussian distribution that integrates to 1 and is always non-negative. Similarly, the likelihood model is a weighted Gaussian that integrates to 1 and is always non-negative. In addition, we remark that a finite set of possible observed and rendered depth images exists, as depth is encoded via a 16-bit integer for each pixel, meaning that there are $2^{16\times H\times W}$ possible images. Therefore, the sum of the likelihood over all possible observed and depth images is finite and well-defined. Hence, there exists a normalization constant $P(\mathbf{c})$, making the approximated posterior map to a proper distribution.

Finally, we score each pose proposal j for the object using the posterior $P(\tilde{\mathbf{c}}_j|\mathbf{c}$ and update the pose of the object in $\mathcal{D}_i$. We repeat this process for every object in the frame and update the scene graph by re-checking occlusion and containment between objects. We repeat this process at every frame to track the poses of objects and maintain a scene graph.

5.2.4 Classification: From Tracking to Categorical Inference

Following object tracking over time, we assign final classes for inference as per the **SCOT** benchmark using a task-generic inference procedure. This procedure translates object poses into a distribution over potential receptacles.

Firstly, we identify the objects that serve as receptacles, i.e., those that match

the rendered receptacle mesh. They are sorted based on their x-coordinates. We also identify reward objects corresponding to the reward mesh.

For each receptacle object i with pose P_i and each reward object j with pose P_j, we estimate the probability that the reward j belongs to receptacle i. This probability is equivalent to drawing P_i from a 3-dimensional Gaussian distribution centered around P_j, with an identity covariance matrix. Consequently, we determine the receptacle r to which reward i belongs by finding the maximum value of this Gaussian:

$$r = \underset{i}{\text{argmax}}\ \mathcal{N}(P_i; P_j, I_3)$$

where I_3 is the three-dimensional identity matrix.

We repeat this process for all reward objects and count the rewards in each receptacle. Following the philosophy of the benchmark that agents aim to maximize the number of obtained rewards, the inferred class is defined as the index of the receptacle that holds the most rewards. The distribution over potential receptacles is calculated as follows:

$$P(r = i) = \frac{1}{R} \sum_{j=1}^{R} \mathcal{N}(P_i; P_j, I_3)$$

where R is the total number of reward objects in the scene.

It's important to note that different tasks may have varying types of receptacles, such as flat plates, which don't necessarily contain or occlude the reward. Therefore, relying solely on the scene graph for inference may not yield accurate results. Hence, we use the categorical inference procedure described above and only use the scene graph to facilitate the tracking procedure.

5.2.5 Additional Implementation Details

Our model operates primarily on the depth images and segmentation images generated by the physical simulator. For real-life data with no ground-truth depth information, substantial work has been proposed on using deep learning for frame-by-frame robust and accurate depth estimation [68, 81, 110] and object instance segmentation [69,

16, 62, 73] on real images. Hence, depth images and segmentation maps can be obtained through a deep learning component. Similarly, we assume the availability of the relevant object meshes for each scene type. This is feasible for many tasks, but even if the assumption is relaxed, meshes can be constructed from the initial observed object point cloud by triangulation.

The proposed model is implemented in Python using JAX, a powerful library for efficient numerical computations. The model extensively uses JAX's accelerated linear algebra (XLA) compilation for improved performance, particularly on GPUs. JAX's just-in-time (JIT) compilation, coupled with its automatic vectorization feature, is used to accelerate depth image rendering from pose hypotheses and to compute the prior and likelihood for observed images in parallel across batches of proposals. For rendering, we use the OpenGL-based parallel renderer from [111].

5.3 Evaluation

We evaluate the proposed model on the **SCOT** benchmark on Supercloud [82], dedicating one V100 GPU for inference on each task. Unlike the machine learning baselines, we perform inference on the entire dataset and report the results in the following subsections.

5.3.1 Experimental Results

Table 5.1 reports the performance of the probabilistic model on the different tasks of the **SCOT** benchmark. In particular, for each task, we present the top-1 accuracy, macro-average F1 score, and mean absolute error results for each complexity level.

Additionally, for comparison, we report the averaged accuracy over the complexity levels of each task of both the probabilistic model and the highest-performing deep learning baselines, XClip and TimeSformer, in Table 5.2.

As we can observe, the probabilistic model demonstrates significant improvements over the deep learning baselines in all tasks. In particular, for the Rotation, Relative Numbers, and Shape Causality tasks, the model correctly identified the receptacle(s) containing the highest rewards in every video in the **SCOT** dataset. In addition,

Level	Addition			Gravity Bias			Rotation		
	Acc.	F1	MAE	Acc.	F1	MAE	Acc.	F1	MAE
1	90.40	86.60	0.21	92.00	90.59	0.08	100.0	100.0	0.00
2	86.40	84.93	0.46	91.00	90.85	0.13	100.0	100.0	0.00
3	86.20	84.35	0.71	93.00	90.91	0.19	100.0	100.0	0.00
4	84.20	83.52	0.84	94.00	93.44	0.12	100.0	100.0	0.00
5	-	-	-	91.00	90.31	0.12	100.0	100.0	0.00
6	-	-	-	94.00	93.49	0.13	100.0	100.0	0.00
7	-	-	-	95.00	92.29	0.18	100.0	100.0	0.00
8	-	-	-	90.00	88.03	0.20	100.0	100.0	0.00
Level	Simple Swap			Relative Numbers			Shape Causality		
	Acc.	F1	MAE	Acc.	F1	MAE	Acc.	F1	MAE
1	92.40	88.42	0.11	100.0	100.0	0.00	100.0	100.0	0.00
2	92.00	88.71	0.11	100.0	100.0	0.00	100.0	100.0	0.00
3	95.20	92.62	0.07	100.0	100.0	0.00	100.0	100.0	0.00
4	94.40	93.89	0.09	100.0	100.0	0.00	-	-	-
5	94.80	94.26	0.08	100.0	100.0	0.00	-	-	-
6	94.00	92.21	0.09	-	-	-	-	-	-
7	92.80	90.61	0.10	-	-	-	-	-	-
8	91.60	91.92	0.12	-	-	-	-	-	-

Table 5.1: **Performance of Bayesian Model on the SCOT Benchmark** Illustrates the top-1 accuracy (Acc. %), macro F1 score (F1 %), and mean absolute error (MAE) achieved by the probabilistic model on each task within the benchmark by complexity level. Dashes (-) indicate that the experiment (column) does not contain the corresponding complexity level (row).

the model considerably outperformed the neural baselines, obtaining 24.98%, 12%, and 53.6%, average accuracy improvement on the Addition, Gravity Bias, and Simple Swap tasks, respectively, over the best neural baseline results for each task.

Model	Addition	Gravity	Rotation	Swap	Relative	Shape
XClip	59.77	75.50	58.15	31.60	86.48	100.0
TimeSformer	61.82	80.50	53.35	39.80	89.14	100.0
Probabilistic Model	86.80	92.50	100.00	93.40	100.00	100.0

Table 5.2: **Performance Comparison of Models on the SCOT Benchmark** Presents the averaged accuracy of the probabilistic model and the highest-performing deep learning baselines (XClip and TimeSformer) on each task.

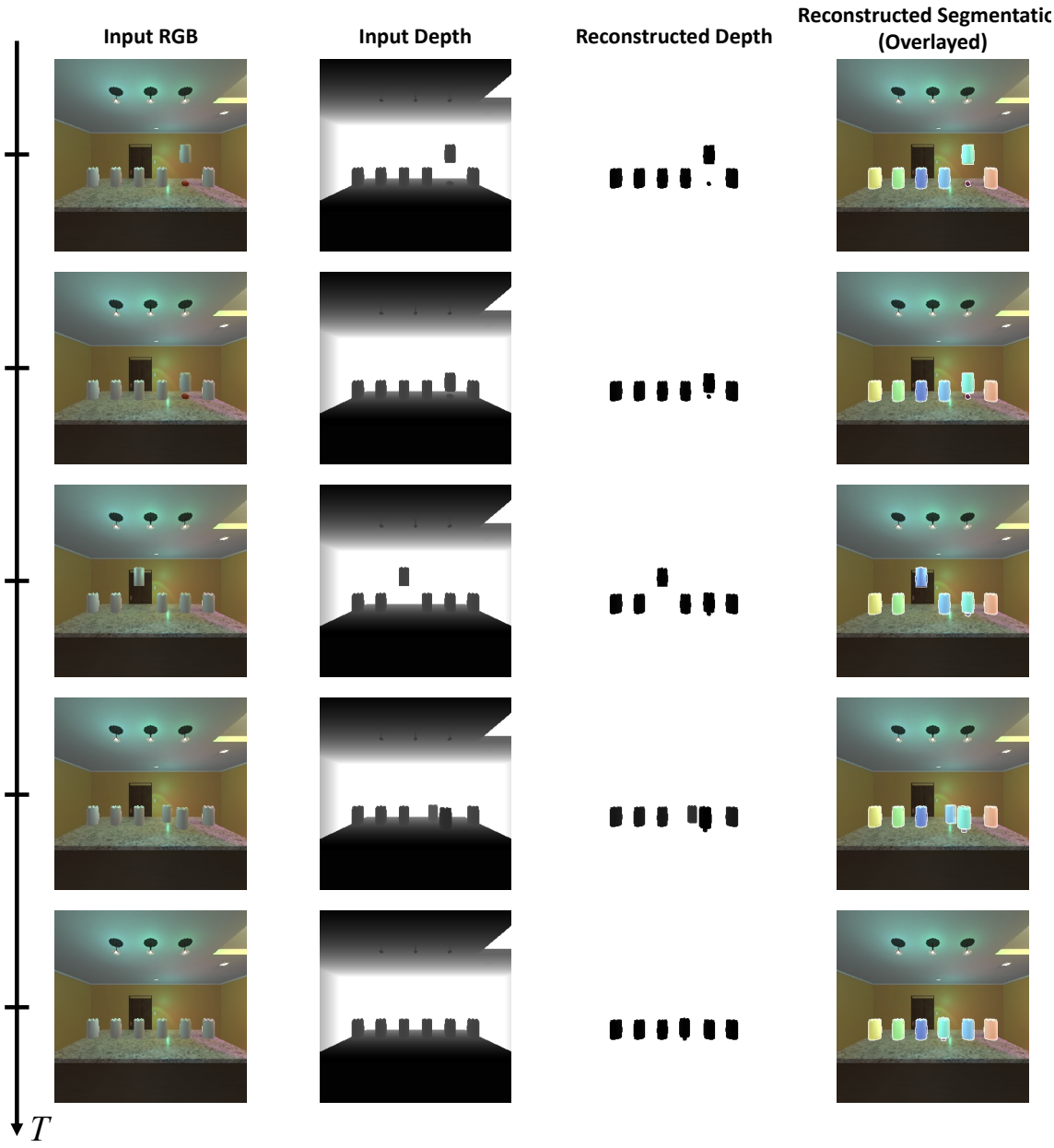


Figure 5-3: **Qualitative Demonstration of the Input and Rendered Representations of the Probabilistic Model** Shows a series of input RGB and depth images from a video in the Rotation dataset, as well as the reconstructed depth and segmentation information of the model's representation of object poses. Reconstructed images were generated by rendering the model's estimate of object poses at each frame following the tracking step.

5.3.2 Discussion

The evaluation results demonstrate the advantage of using Bayesian inverse graphics across all tasks in the **SCOT** benchmark. In particular, the proposed probabilistic

model consistently outperformed neural baseline models, showcasing the advantages of integrating prior knowledge and probabilistic reasoning into object tracking and scene understanding.

In tasks such as Rotation, Relative Numbers, and Shape Causality, the proposed model achieved perfect accuracy, suggesting that it can accurately capture and predict the dynamics of these environments. In particular, it shows that representing objects symbolically and tracking them through iterative hypothesis testing significantly aids intuitive physics understanding.

For the Addition and Gravity Bias tasks, while there was a slight decrease in performance as the complexity level increased, the model still showed superior results compared to the best-performing neural baselines. Even as the number of objects or interactions increases, both variables that contribute uncertainty to the estimation process, the model maintained a high degree of accuracy, demonstrating higher robustness and resilience to increasing task difficulty.

Similarly, in the Simple Swap task, the model maintained a high level of performance, even as the complexity level increased, highlighting its strong ability to track multiple objects simultaneously and accurately reason about their dynamics. Additionally, we do not observe a consistent drop in performance as task complexity increases, suggesting that accurate predictions are not necessarily a function of task complexity and indicating the model's capability to generalize to longer times of object manipulation.

The observed performance increase can be traced back to the different components of the inference mechanism. First, using a physical renderer allows the model to infer the effect of causality when objects cause pose variations in other objects, as in the Shape Causality task. Moreover, representing objects symbolically and using hierarchical priors on object relations, especially containment, allows the model to predict and maintain object compositions over time in the Addition, Simple Swap, and Rotation tasks. In addition, incorporating physical priors such as gravity and velocity significantly aids the Gravity Bias task, enabling the model to predict object movements without observation. The enumerative tracking is also crucial in generat-

ing and updating object pose hypotheses, particularly useful in the Simple Swap and Rotation tasks. Finally, the noise model on depth data enables the model to estimate the likelihood of these hypotheses and hence object poses over time.

Moreover, Figure 5-3 qualitatively shows the robustness of the Bayesian probabilistic model. We see a strong qualitative match between the reconstructed depth and segmentation maps; and the ground truth depth images. This demonstrates the model's ability to track object poses accurately, even within dynamic scenes like the Rotation task. This validates integrating probabilistic reasoning and generative modeling in object tracking and scene understanding tasks.

Lastly, we remark that the current implementation of the model has several limitations. The model's sequential nature can lead to significant errors, as a failure in one component can disrupt the entire inference pipeline. This issue could be mitigated by introducing a mechanism for retroactive error correction, such as an error detection and correction module, which could backtrack and revise incorrect inferences when later stages encounter high levels of uncertainty. Another challenge lies in manually fine-tuning several gridding and likelihood parameters, which can lead to sub-optimal performance. Using automated parameter tuning methods, such as Markov Chain Monte Carlo techniques, could alleviate this problem and provide means of recovering from significant inference errors. Lastly, the model's reliance on ground-truth segmentation and depth maps could introduce further uncertainty when these inputs are estimated from real-world RGB images. Future iterations could incorporate uncertainty handling mechanisms, such as Bayesian deep learning models, to incorporate estimation uncertainties of deep learning models during tracking. Addressing these limitations can enhance the model's robustness and applicability, providing interesting directions for future research.

Chapter 6

Future Works

This chapter outlines potential directions for future research based on the findings of this thesis. We achieved promising results on the **SCOT** benchmark using the generative probabilistic model described in Chapter 5, but numerous opportunities exist for further exploration and improvement. We discuss four potential research directions: 1) the use of real-world data; 2) the expansion of benchmark task types and dataset generation procedure; 3) the evaluation of additional neural or neuro-symbolic models; and 4) employing MCMC methods to fine-tune different components of the inference pipeline.

Use of real-world data Our work heavily relied on simulated data, allowing us to work under controlled conditions, generate a large amount of video data, and obtain strong supervision via dense annotations. However, simulated data may only partially capture the complexities and unpredictability of the real world. For example, our graphics simulator has a simplified model of lighting, sensor noise, and rendering functions compared to the real world, which may impact robustness to color constancy. Additionally, real-world data introduces challenges like noise, variability, and the need for manual annotation, but it also provides opportunities for learning more robust models.

Therefore, an area of future work could involve applying our techniques to real-world data. Incorporating real-life data would require identifying suitable real-world datasets or creating new datasets for similar cognitive tasks and applying our model to make inferences on those datasets. Such efforts could provide

insights into the performance of our models under real-world conditions and reveal areas needing improvement.

Moreover, the generation of synthetic data could be improved by using advanced domain adaptation methods and rendering or by potentially using techniques such as Generative Adversarial Networks (GANs) [21] or Variational Autoencoders [56], and exploring techniques to incorporate real-world variability into synthetic data or combining real and synthetic data during training.

Expansion of task types & dataset generation procedure Although the cognitive tasks in the dataset cover numerous core cognitive capacities, future research could enhance the **SCOT** dataset generation pipeline with additional cognitive tasks targeting other core knowledge systems. Specifically, tasks for intuitive psychology, where videos simulate agents attempting to achieve a particular goal within the environment, could be implemented. The observing model would then have to infer the goal. Another line of research may involve integrating additional modalities, such as audio or text, into tasks, thereby requiring models to make inferences based on multimodal perception.

In addition, future work could enhance the scene generation procedure in Unity by incorporating insights from recent research in synthetic scene generation. For instance, future work can leverage diffusion-based approaches such as Pronovost et al. [77] to generate scene descriptions aligned with real-life scenarios. Additionally, similar to Meta-Sim2 [26], reinforcement learning can be employed to learn scene structure and automatically adjust parameters for more accurate synthetic scene generation. Other approaches include using textual prompts for scene generation and learning a distribution of synthetic scenes over the language domain. By combining these approaches, we can generate more realistic and diverse scenes in Unity, leading to more effective training of deep learning models. Future work could investigate how combinations of those strategies and the dataset generation procedure described in Chapter 3 can create more realistic and varied intuitive physics scenes.

Evaluation of additional models Over the past two years, numerous vision models and techniques have been proposed, offering ample opportunities to expand upon our work and potentially mitigate the failure modes of both the neural baselines and the probabilistic model. For instance, diffusion models, which model data generation as a stochastic differential equation, have demonstrated promising generative capacities and could be used for scene perception. Additionally, reinforcement learning algorithms for scene perception and decision-making offer an exciting avenue of research. Future work could explore their performance on the **SCOT** benchmark.

Ideally, future work will focus on combining probabilistic models with strong physical inductive biases and neural models with robust generative and feature extraction capacities. The probabilistic model can provide a formal framework for modeling uncertainty and complex dependencies, while the neural model can learn features that facilitate probabilistic inference. Combining the two types of models could potentially lead to more robust and accurate models for scene perception and cognitive tasks.

Enabling MCMC methods for proposal evaluation Markov Chain Monte Carlo (MCMC) methods offer a promising direction for fine-tuning the different components of the inference pipeline in our research. In the context of our work, MCMC methods could be used to fine-tune the mesh-initialization and tracking components of our probabilistic model. Specifically, Metropolis-Hastings MCMC could be used to iteratively propose small changes to object poses in each frame and accept them based on the MH acceptance probability, thus enabling the model to explore the state space more efficiently. The current proposal generation pipeline can be easily adapted into a first-order Markov chain that is irreducible, aperiodic, positive-recurrent, and thus converges to a stationary distribution.

Chapter 7

Conclusions

This thesis explores the potential of computational models for scene perception, an essential aspect of human cognition. The thesis discussed the complexities and nuances in achieving a generalizable understanding of physical scenes, including the importance of prior knowledge of physical dynamics, compositional understanding of scene structures, and causal mapping of events.

Deep learning models have dictated the status quo in scene perception tasks, achieving state-of-the-art results in several challenging benchmarks that evaluate performance on downstream tasks such as object identification, semantic segmentation, and pose estimation. Nonetheless, existing models often struggle to achieve scene understanding at a high cognitive level with limited visual exposure, especially under noisy conditions.

This work first proposed a data generation pipeline, which was used to create a million-scale dataset of synthetic images and metadata. This dataset was used to design the **SCOT** benchmark for evaluating core knowledge understanding in agents, particularly objects, space, and quantities. Agents are expected to observe dynamic intuitive physics scenes and make inferences that maximize their obtained rewards.

This research has also studied the performance of several recent deep neural models on the proposed benchmark. We have developed a framework to facilitate the training and evaluation of neural models implemented in PyTorch on the **SCOT** benchmark. We experimented with training and evaluating several deep learning baselines on the benchmark to identify the strengths and limitations of deep learning algorithms in

performing cognitive visual tasks.

Following training and evaluation, we found that these models excel in tasks that primarily require spatial awareness and the identification of visual patterns. However, their performance significantly diminishes when the tasks involve higher levels of cognitive processing, abstraction, or dynamic understanding. Additionally, while large models like TimeSformer and XClip perform well on videos with simple dynamics, they perform poorly on the same tasks but with more complex dynamics.

Finally, inspired by the *intuitive physics engine* framework of human perception, we integrated Bayesian modeling and inverse graphics to construct a probabilistic generative model. The proposed model explicitly encodes physical priors and compositional structures, using a Markov model to make causal inferences about object movements and changes in scene structure. We demonstrated that the proposed model performed significantly better than the neural baselines, achieving perfect accuracy on a few.

While the model implementation used ground-truth depth and segmentation maps from the renderer, this thesis shows that a probabilistic generative model with a simple physics prior can accurately perform inference over complex scene dynamics. We believe that better systems can emerge by combining the powerful predictive capacity of neural models with the robust representations of the probabilistic model. For instance, future iterations of this work could incorporate deep learning components that estimate the depth images and segmentation maps from RGB images and use the model to perform high-level tracking and inference over structured scene representations.

Looking ahead, this research has identified several promising directions for future work. These include the incorporation of real-world data, the expansion of the types and complexity of tasks in the benchmark, the evaluation of additional models, and the potential use of Markov Chain Monte Carlo methods to fine-tune inference processes. These directions highlight the scope for further advancements in computational models for scene understanding.

Bibliography

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Dhaval Adjodah, Tim Klinger, and Joshua Joseph. Symbolic relation networks for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS) Workshop on Representation Learning*, 2018.

[3] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[4] Merav Ahissar and Shaul Hochstein. The reverse hierarchy theory of visual perceptual learning. *Trends in Cognitive Sciences*, 8(10):457–464, 2004.

[5] Kelsey R. Allen, Kevin A. Smith, and Joshua B. Tenenbaum. Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *Proceedings of the National Academy of Sciences*, 117(47):29302–29310, 2020.

[6] Wilma A Bainbridge. The resiliency of image memorability: A predictor of memory separate from attention and priming. *Neuropsychologia*, 141:107408, February 2020.

[7] Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009. Reinforcement learning and higher cognition.

[8] Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[9] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[10] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[11] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[12] Sarah T Boysen and Gary G Berntson. Responses to quantity: perceptual versus cognitive mechanisms in chimpanzees (pan troglodytes). *Journal of Experimental Psychology: Animal Behavior Processes*, 21(1):82, 1995.

[13] Juliane Bräuer, Juliane Kaminski, Julia Riedel, Josep Call, and Michael Tomasello. Making inferences about the location of hidden food: social dog, causal ape. *Journal of Comparative Psychology*, 120(1):38, 2006.

[14] Paola Cascante-Bonilla, Khaled Shehada, James Seale Smith, Sivan Doveh, Donghyun Kim, Rameswar Panda, Gül Varol, Aude Oliva, Vicente Ordonez, Rogerio Feris, and Leonid Karlinsky. Going beyond nouns with vision language models using synthetic data. *arXiv preprint arXiv:2303.17590*, 2023.

[15] Paola Cascante-Bonilla, Hui Wu, Letao Wang, Rogerio S Feris, and Vicente Ordonez. Simvqa: Exploring simulated environments for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5056–5066, 2022.

[16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[17] Vincent S. Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2580–2590, 2019.

[18] Zhe Chen. Object-based attention: A tutorial review. *Attention, Perception, & Psychophysics*, 74(5):784–802, 2012.

[19] Marvin M. Chun and Jeremy M. Wolfe. Visual attention. *Blackwell Handbook of Perception*, page 272–310, 2001.

[20] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[21] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[22] Marco F. Cusumano-Towner, Feras A. Saad, Alexander K. Lew, and Vikash K. Mansinghka. Gen: A general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, pages 221–236, New York, NY, USA, 2019. ACM.

[23] Marco F Cusumano-Towner, Feras A Saad, Alexander K Lew, and Vikash K Mansinghka. Gen: a general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th acm sigplan conference on programming language design and implementation*, pages 221–236, 2019.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[25] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023.

[26] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. Meta-sim2: Unsupervised learning of scene structure for synthetic data generation. In *Proceedings of the European conference on Computer Vision (ECCV)*, pages 715–733. Springer, 2020.

[27] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015.

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[29] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

[30] Epic Games. Unreal engine.

[31] Russell A. Epstein and Chris I. Baker. Scene perception in the human brain. *Annual Review of Vision Science*, 5(1):373–397, 2019. PMID: 31226012.

[32] Claudia Fichtel, Klara Dinter, and Peter M Kappeler. The lemur baseline: how lemurs compare to monkeys and apes in the primate cognition test battery. *PeerJ*, 8:e10025, 2020.

[33] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. In *International Conference on Learning Representations (ICLR)*, 2015.

[34] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

[35] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[36] Alberto Garcia-Garcia, Pablo Martinez-Gonzalez, Sergiu Oprea, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Alvaro Jover-Alvarez. The robotrix: An extremely photorealistic and very-large-scale indoor dataset of sequences with robot trajectories and interactions. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6790–6797. IEEE, 2018.

[37] Jacob Gildenblat and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

[38] Noah Goodman, Vikash Mansinghka, Daniel M Roy, Keith Bonawitz, and Joshua B Tenenbaum. Church: a language for generative models. *arXiv preprint arXiv:1206.3255*, 2012.

[39] Nishad Gothoskar, Marco Cusumano-Towner, Ben Zinberg, Matin Ghavamizadeh, Falk Pollok, Austin Garrett, Joshua B Tenenbaum, Dan Gutfreund, and Vikash Mansinghka. 3dp3: 3d scene perception via probabilistic programming. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[40] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition, 2020.

[41] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2616–2625, 2017.

[42] Jessica B Hamrick, Peter W Battaglia, Thomas L Griffiths, and Joshua B Tenenbaum. Inferring mass in complex scenes by mental simulation. *Cognition*, 157:61–76, 2016.

[43] Daniela Hartmann, Marina Davila-Ross, Siew Te Wong, Josep Call, and Marina Scheumann. Spatial transposition tasks in indian sloth bears (melursus

ursinus) and bornean sun bears (helarctos malayanus euryspilus). *Journal of Comparative Psychology*, 131(4):290, 2017.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[45] Esther Herrmann, Josep Call, María Victoria Hernàndez-Lloreda, Brian Hare, and Michael Tomasello. Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis. *Science*, 317(5843):1360–1366, 2007.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[47] Shaul Hochstein and Merav Ahissar. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804, 2002.

[48] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[49] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1610–1618, 2021.

[50] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017.

[51] Amy S Joh, Vikram K Jaswal, and Rachel Keen. Imagining a way out of the gravity bias: Preschoolers can visualize the solution to a spatial problem. *Child Development*, 82(3):744–750, 2011.

[52] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

[53] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1521–1529, 2017.

[54] Minhee Kim and Kaibo Liu. A bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics. *IISE Transactions*, 53(3):326–340, 2020.

[55] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[56] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[57] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[58] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 123:32–73, 2017.

[59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, 2012.

[60] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision (IJCV)*, 128(7):1956–1981, 2020.

[61] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

[62] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, 2018.

[63] Hoang-An Le, Thomas Mensink, Partha Das, Sezer Karaoglu, and Theo Gevers. Eden: Multimodal synthetic dataset of enclosed garden scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1579–1589, 2021.

[64] Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Mądry. Ffcv: Accelerating training by removing data bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12011–12020, 2023.

[65] Yann LeCun and Yoshua Bengio. *Convolutional Networks for Images, Speech, and Time Series*, page 255–258. MIT Press, Cambridge, MA, USA, 1998.

[66] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[67] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[68] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015.

[69] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[70] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 638–647, 2022.

[71] Vikash K Mansinghka, Tejas D Kulkarni, Yura N Perov, and Joshua B Tenenbaum. Approximate bayesian image interpretation using generative probabilistic graphics programs. *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013.

[72] Enrico Meloni, Luca Pasqualini, Matteo Tiezzi, Marco Gori, and Stefano Melacci. Sailenv: Learning in virtual visual environments made simple. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8906–8913. IEEE, 2021.

[73] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

[74] Caris Moses, Michael Noseworthy, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Nicholas Roy. Visual prediction of priors for articulated object interaction. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10480–10486, 2020.

[75] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance

deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2019.

[76] Arul Selvam Periyasamy, Max Schwarz, and Sven Behnke. Refining 6d object pose predictions using abstract render-and-compare. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 739–746. IEEE, 2019.

[77] Ethan Pronovost, Kai Wang, and Nicholas Roy. Generating driving scenes with diffusion. *arXiv preprint arXiv:2305.18452*, 2023.

[78] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[79] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[80] Rishi Rajalingham, Aida Piccato, and Mehrdad Jazayeri. The role of mental simulation in primate physical inference abilities. *bioRxiv*, 2021.

[81] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

[82] Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Lauren Milechin, Julia Mullen, Andrew Prout, Antonio Rosa, Charles Yee, and Peter Michaleas. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2018.

[83] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[84] Nicholas Roy, Ingmar Posner, Tim Barfoot, Philippe Beaudoin, Yoshua Bengio, Jeannette Bohg, Oliver Brock, Isabelle Depatie, Dieter Fox, Dan Koditschek, Tomas Lozano-Perez, Vikash Mansinghka, Christopher Pal, Blake Richards, Dorsa Sadigh, Stefan Schaal, Gaurav Sukhatme, Denis Therien, Marc Toussaint, and Michiel Panne. From machine learning to robotics: Challenges and opportunities for embodied intelligence. 2021.

[85] Duane M Rumbaugh, Sue Savage-Rumbaugh, and Mark T Hegel. Summation in the chimpanzee (pan troglodytes). *Journal of Experimental Psychology: Animal Behavior Processes*, 13(2):107, 1987.

[86] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. Ieee, 2015.

[87] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9339–9347, 2019.

[88] Vanessa Schmitt, Birte Pankau, and Julia Fischer. Old world monkeys compare to apes in the primate cognition test battery. *PLOS ONE*, 7(4):1–10, 04 2012.

[89] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007, 2018.

[90] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[91] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2107–2116, 2017.

[92] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[93] Catherine Sophian. Spatial transpositions and the early development of search. *Developmental Psychology*, 20(1):21, 1984.

[94] Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[95] Gregory J Stein and Nicholas Roy. Genesis-rt: Generating synthetic images for training secondary real-world tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7151–7158. IEEE, 2018.

[96] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot,

Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[97] Russ Tedrake. Drake: Model-based design and verification for robotics, 2019.

[98] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.

[99] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 969–977, 2018.

[100] Jonathan Tremblay, Thang To, and Stan Birchfield. Falling things: A synthetic dataset for 3d object detection and pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2038–2041, 2018.

[101] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *Conference on Robot Learning*, pages 306–316. PMLR, 2018.

[102] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, 2017.

[103] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017.

[104] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[105] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16611–16621, 2021.

[106] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[107] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.

[108] Riccardo Zanella, Alessio Caporali, Kalyan Tadaka, Daniele De Gregorio, and Gianluca Palli. Auto-generated wires dataset for semantic segmentation with domain-independence. In *International Conference on Computer, Control and Robotics (ICCCR)*, pages 292–298. IEEE, 2021.

[109] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1386–1383. IEEE, 2017.

[110] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian. Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences*, 63(9):1612–1627, 2020.

[111] Guangyao Zhou, Nishad Gothoskar, Lirui Wang, Joshua B Tenenbaum, Dan Gutfreund, Miguel Lázaro-Gredilla, Dileep George, and Vikash K Mansinghka. 3d neural embedding likelihood for robust sim-to-real transfer in inverse graphics. *arXiv preprint arXiv:2302.03744*, 2023.

[112] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.