UNIVERSITY OF CALIFORNIA

Los Angeles

Value Learning for Interactive Games, Embodied Artificial Intelligence, and Robotics

A dissertation submitted in partial satisfaction

of the requirements for the degree Doctor of Philosophy

in Statistics

by

Yizhou Zhao

2023

ABSTRACT OF THE DISSERTATION

Value Learning for Interactive Games, Embodied Artificial Intelligence, and Robotics

by

Yizhou Zhao
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2023
Professor Song-Chun Zhu, Co-Chair
Professor Yingnian Wu, Co-Chair

Simulation plays a crucial role in modern academic study, particularly in the field of artificial intelligence (AI). The simulation environment can mimic real-world scenarios, allowing the AI agent to learn, adapt, and make decisions in a controlled and safe setting. This thesis tackles two important problems in building the next generation of artificial general intelligence (AGI): how to efficiently train an AI agent with values and how to overcome the simulation to reality gap to bring the training results to real-world applications. The current studies of AI mainly consider learning about the potential or energy function (U), referring to understanding the impact of the outside environment. The U function helps the agent apprehend the physical world laws, natural potentials, and social norms. However, taking into account the value learning, usually representing modeling one's inside thinking, benefits the agent to derive its goals, intents, and social values.

Our research shows that both U and V learning are equally important to the pathway to AGI. The learning of U is usually data-driven. It enables the agent to imitate and complete the task through statistical learning. By incorporating the value function, the agent can spontaneously specify a task plan and its behavior is more in line with human cognition and value.

This thesis consists of three parts: (1) Potential function learning, which explores the

process of acquiring knowledge or skills that are useful and practical for a particular purpose. (2) Value learning when learning the potential (U) function can not satisfy all the learning goals, which investigates situations where utility-based learning approaches might be limited or ineffective. (3) Combining U and V learning, which focuses on the integration of simulation-based learning and data-driven learning methods.

We primarily focus on assessing the effectiveness of U learning within a simulated environment. Our investigation commences with agents operating in a controlled simulated setting, where the action space is intentionally kept small. Through rigorous testing and iterative refinement, we gradually expand the scope of our analysis to encompass agents dealing with increasingly complex and continuous action spaces. Upon achieving compelling results in the simulated realm, we proceed to the crucial next step: transferring the knowledge and expertise gained from the well-trained agents in the simulation space to real-world scenarios. This process entails adapting the learned policies, strategies, and decision-making capabilities of the agents to navigate the intricacies and uncertainties of genuine environments.

The dissertation of Yizhou Zhao is approved.

Kai-Wei Chang

Jingyi Li

Song-Chun Zhu, Committee Co-Chair

Yingnian Wu, Committee Co-Chair

University of California, Los Angeles

2023

Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

First, I would like to thank my advisor at the Computer Science and Statistics Department, Song-Chun Zhu. Professor Zhu is a pioneer and an excellent explorer, who leads me to explore my interest in the field of AI. He is always supportive and he encourages me to tackle the innovative, valuable, and challenging topics. I am grateful to him, and I wish him well in Beijing. I believe he can lead the lab to another great success.

I would also like to express my sincere gratitude to co-chair, Professor Ying Nian Wu. Dr. Wu is a great mathematician, a statistician, and a computer scientist. He is also extremely helpful and kind person. During my Ph.D. study at UCLA, I always enjoyed discussing with him about my ideas, experiments, and papers. I have been deeply impressed by his passion for finding simple and genuine solutions for scientific problems.

It is also my great honor to have Professor Kai-Wei Chang on my thesis committee. A part of my research focus on the dialogue systems, which is very relevant to his work. I also feel lucky to have Professor Jessica Jingyi Li. I took her classes about regression models and the methodology I learned still throws light on my study for simulation to reality today.

During my Ph.D., I have also done two memorable internships at Amazon. I would like to thank my mentors and managers Qiaozi Gao, Kaixiang Lin, Govind Thattai, and Gaurav Sukhatme. My internship Alexa helped me dive deeper into the synthetic data generation and led me to the study of Embodied AI. The work about indoor scene synthesis and robotics inspired me to build my own start-up (Digital Bot Lab, Inc) to apply the synthetics methods to solve realistic problems.

I also had a great time as a part-time researcher at Cara.ai. I thank my colleagues, friends, and teachers. I thank all co-authors of my papers, including Liang Qiu, Pan Lu, Xiaofeng Gao, Steven Gong, Yaofang (Victor) Zhang, Ziheng Zhou, Feng Shi, and Tian Han. Especially Liang and Pan, my dissertation would not be possible without their great contributions.

Finally, I thank my family for their unconditional love and support to guide me through the hardest pandemic period during my study at UCLA.

VITA

Education

M.S., Statistics University of California, Berkeley	August 2016 - December 2017
B.S., Mathematics Peking University	September 2012 - June 2016

Publications

Y. Zhao, Y. Zeng, Q. Long, Y.N. Wu, S.-C. Zhu. Sim2Plan: Robot Motion Planning via Message Passing between Simulation and Reality. Future Technology Conference (FTC), 2023.

R. Gong, J. Huang, Y. Zhao, H. Geng, X. Gao, Q. Wu, W. Ai, Z. Zhou, D. Terzopoulos, S.-C. Zhu, B. Jia, S. Huang. ARNOLD: A Benchmark for Language-Grounded Task Learning with Continuous States in Realistic Scenes. International Conference on Computer Vision (ICCV), 2023.

L. Qiu, Y. Zhao, J. Li, P. Lu, B. Peng, J. Gao, S.-C. Zhu. ValueNet: A New Dataset for Human Value Driven Dialogue System. Association for the Advancement of Artificial Intelligence (AAAI), 2022.

Y. Zhao, K. Lin, Z. Jia, Q. Gao, G. Thattai, J. Thomason, G. S. Sukhatme, Luminous: Indoor scene generation for embodied ai challenges, CtrlGen Workshop at Neural Information Processing Systems (NeurIPS), 2021.

Y. Zhao, L. Qiu, P. Lu, F. Shi, T. Han, S.-C. Zhu. Learning from the Tangram to Solve Mini Visual Tasks. Association for the Advancement of Artificial Intelligence (AAAI), 2022.

P. Lu, L. Qiu, J. Chen, T. Xia, Y. Zhao, W. Zhang, Z. Yu, X. Liang, S.-C. Zhu. IconQA: A New Benchmark for Abstract Diagram Understanding and Visual Language Reasoning. Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, 2021.

L. Qiu, Y. Liang, Y. Zhao, P. Lu, B. Peng, Z. Yu, Y.N. Wu, S.-C. Zhu. SocAoG: Incremental Graph Parsing for Social Relation Inference in Dialogues. Association for Computational Linguistics (ACL), 2021.

L. Qiu, Y. Zhao, W. Shi, Y. Liang, F. Shi, T. Yuan, Z. Yu, S.-C. Zhu. Structured Attention for Unsupervised Dialogue Structure Induction. Empirical Methods in Natural Language Processing (EMNLP), 2020.

X. Gao, R. Gong, Y. Zhao, S. Wang, T. Shu, and S.-C. Zhu. Joint Mind Modeling for Expla-nation Generation in Complex Human-Robot Collaborative Tasks. IEEE International Conferenceon Robot and Human Interactive Communication (RO-MAN), 2020.

R. Gong, J. Huang, Y. Zhao, H. Geng, X. Gao, Q. Wu, W. Ai, Z. Zhou, D. Terzopoulos, S.-C. Zhu, B. Jia, S. Huang. ARNOLD: A Benchmark for Language-Grounded Task Learning with Continuous States in Realistic Scenes. CoRL 2022 Workshop on Language and Robot Learning.

Y. Zhao, S. Gong, X. Gao, W. Ai, S.-C. Zhu. (2022). VRKitchen2.0-IndoorKit: A Tutorial for Augmented Indoor Scene Building in Omniverse. *arXiv*.
https://arxiv.org/abs/2206.11887

Y. Zhao, W. Ai, L. Qiu, P. Lu, F. Shi, T. Han, S.-C. Zhu. (2021). GenMotion: Data-driven Motion Generators for Real-time Animation Synthesis. *arXiv*.
https://arxiv.org/abs/2112.06060

Y. Zhao, L. Qiu, W. Ai, P. Lu, S.-C. Zhu. (2022). Triangular Character Animation Sampling with Motion, Emotion, and Relation. *arXiv*. https://arxiv.org/pdf/2203.04930.pdf

Y. Zhao, S.-C. Zhu. (2020). Weighted Entropy Modification for Soft Actor-Critic. Weighted Entropy Modification for Soft Actor-Critic. *arXiv*.
https://arxiv.org/abs/2011.09083

Y. Zhao, L. Qiu, W. Ai, F. Shi, S.-C. Zhu. (2020). Vertical-Horizontal Structured Attention for Generating Music with Chords. *arXiv*. https://arxiv.org/abs/2011.09078

Chapter 1

Introduction

In statistical learning, both Utility (U) and Value (V) are important components of the probabilistic framework used to model scenes over time and guide agent behavior. *U* captures the likelihood of different scene configurations based on observed frequencies, physical laws, and social norms, while *V* represents the desirability and preferences that drive agent actions to optimize their behavior and achieve certain goals.

U, which is usually well-studied well-documented in machine learning, refers to the potential function in a probability distribution represented as:

$$p(pg;\Lambda) = \frac{1}{Z}\exp\{-U(pg;\Lambda)\} \tag{1.1}$$

Here, *pg* represents the parse graph, a realized node of a Spatial, Temporal, and Causal And-Or Graph (STC-AOG), and Λ is the parameter. The potential function *U* captures the occurring frequency and social and physical norms. It is learned from statistical observations and is associated with inductive learning. Essentially, *U* helps model the likelihood of different configurations, considering various attributes and relations between nodes.

The Value function *V* accounts for preferences, motivations, social values, morality, and other factors that guide the behavior of agents.

$$V(pg;\Omega) = f_\Omega(pg) \tag{1.2}$$

pg represents the parse graph and Ω is the parameter specifying a value system. *U* is applied to drive and optimize the actions of agents, aiming to improve the world and environments to meet human needs. Unlike the potential function U, which focuses on frequency and social norms, the utility function V is concerned with the desirability of different configurations of the scene.

The current machine learning studies often focus on U, and are typically conducted by amassing a substantial volume of data labeled by human annotators. This approach seeks to capture the underlying patterns of occurrences and social norms within scenes over time, allowing models to learn from statistical observations and reproduce observed frequencies.

In contrast, the study of V, which encapsulates preferences and guiding principles that drive agent behavior, remains relatively under-explored. The challenge lies in the intricate nature of individual preferences, making the collection of value-related data a daunting task.

As a result, investigations into V often resort to simulation methodologies, such as the formulation of value functions in certain reinforcement learning frameworks like Actor-Critic. These simulations aim to approximate and encapsulate the complex interplay of preferences and desires that motivate agent actions.

It becomes a common practice to employ U during the initial stages of model training to instill a fundamental understanding of the environment within agents. In this pre-training phase, U equips agents with a foundational grasp of scene configurations and norms. Subsequently, V takes center stage, undergoing fine-tuning, transfer learning, and even few-shot learning. This process allows agents to refine their behavior, adapt to new contexts, and develop their unique set of preferences that govern their interactions with the environment.

Chapter 2

Learning Energy Function for Indoor Scene Synthesize

Learning-based methods for training embodied agents typically require a large number of high-quality scenes that contain realistic layouts and support meaningful interactions. However, current simulators for Embodied AI (EAI) challenges only provide simulated indoor scenes with a limited number of layouts. This paper presents LUMINOUS, the first research framework that employs state-of-the-art indoor scene synthesis algorithms to generate large-scale simulated scenes for Embodied AI challenges. Further, we automatically and quantitatively evaluate the quality of generated indoor scenes via their ability to support complex household tasks. LUMINOUS incorporates a novel scene generation algorithm (Constrained Stochastic Scene Generation (CSSG)), which achieves competitive performance with human-designed scenes. Within LUMINOUS, the EAI task executor, task instruction generation module, and video rendering toolkit can collectively generate a massive multimodal dataset of new scenes for the training and evaluation of Embodied AI agents. Extensive experimental results demonstrate the effectiveness of the data generated by LUMINOUS, enabling the comprehensive assessment of embodied agents on generalization and robustness.

2.1 Introduction

Embodied artificial intelligence (EAI) has attracted significant attention, both in advanced deep learning models and algorithms [VSP17, LBP19, SGT21, ZC21] and the rapid development of simulated platforms [PRB18, KMH17, GSA20, LXM21, SKM19]. Many open challenges [STG20, SCU21, WDK21, SXL20] have been proposed to facilitate EAI research. A critical bottleneck in existing simulated platforms [STG20, WDK21, LXM21, PRB18, YMB18] is the limited number of indoor scenes that support vision-and-language navigation, object interaction, and complex household tasks. This limitation makes it difficult to verify whether state-of-the-art methods generalize well to unseen scenarios or whether they are specialized to a small number of room structures. Low cost, automatic creation of large numbers of high-quality simulated environments is essential to resolve this question.

Here, we leverage advances in indoor scene synthesis to achieve the large-scale automatic creation of simulated environments. Indoor scene synthesis has been a long-standing challenge for both computer graphics and machine learning communities resulting in considerable recent progress [YYT11, FRS12, FSL15, QZH18, WLW19, LZW20, ZWK19, WYN20, ZYM20]. To effectively utilize indoor scene synthesis for EAI, three key challenges remain. First, for synthesized scenes to be useful in EAI, they must directly support household tasks requiring object pick and place, state changes, and articulation. Second, the generated scenes with randomized layouts must be *natural*—layouts that *make sense* according to human judgement—and *functional*—layouts that match human use given the room type, such as *Bedroom* or *Living Room*. Finally, any scene generation method must provide efficient access to massive, multimodal embodied agent trajectory data, including low-level action sequences for task completion, egocentric image frames during action execution, and language instructions.

We present LUMINOUS, a scalable, indoor scene generation framework to facilitate EAI tasks such as vision-and-language navigation and language-guided task completion (Figure 2.1). We introduce the Challenge Definition Format (CDF), which provides a user-friendly task specification of the required objects, their relative spatial relationships, and high-level descriptions of downstream EAI tasks to facilitate. We introduce Constrained Stochastic Scene Generation (CSSG) to generate an arbitrary number of indoor scenes from the CDF specification. LUMINOUS produces scenes that are well-aligned with human common sense and satisfy the CDF conditions, thereby ensuring that the generated scenes are readily applicable to EAI tasks. In addition, we develop a task solver to plan sequences of low-level actions for corresponding task

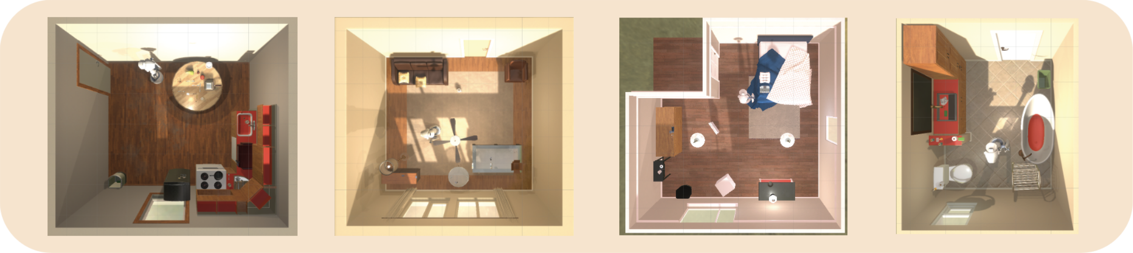


Figure 2.1: Generated Indoor scenes. LUMINOUS scenes are evaluated quantitatively via EAI task success rates and qualitatively via human judgements.

completion. We also implement a task instruction generation module to annotate trajectories with language instructions. LUMINOUS generates large-scale multimodal trajectories for the training and evaluation of embodied agents.

LUMINOUS also contributes to indoor scene synthesis. Generally, scene generation lacks ground truth for quantitative evaluation. Metrics like bounding box and angle prediction [LZW20] and synthetic classification [WLW19] are not always correlated with the quality of a generated scene. By connecting indoor scene synthesis to EAI, we propose measuring planner-based task success rate as an automatic evaluation metric of the synthesized scene quality. Besides CSSG, LUMINOUS is compatible with state-of-the-art learning-based indoor scene synthesis algorithms [WSC18, LZW20]. We demonstrate that CSSG with LUMINOUS qualitatively outperforms other learning-based synthesis methods (Section 2.4.1).

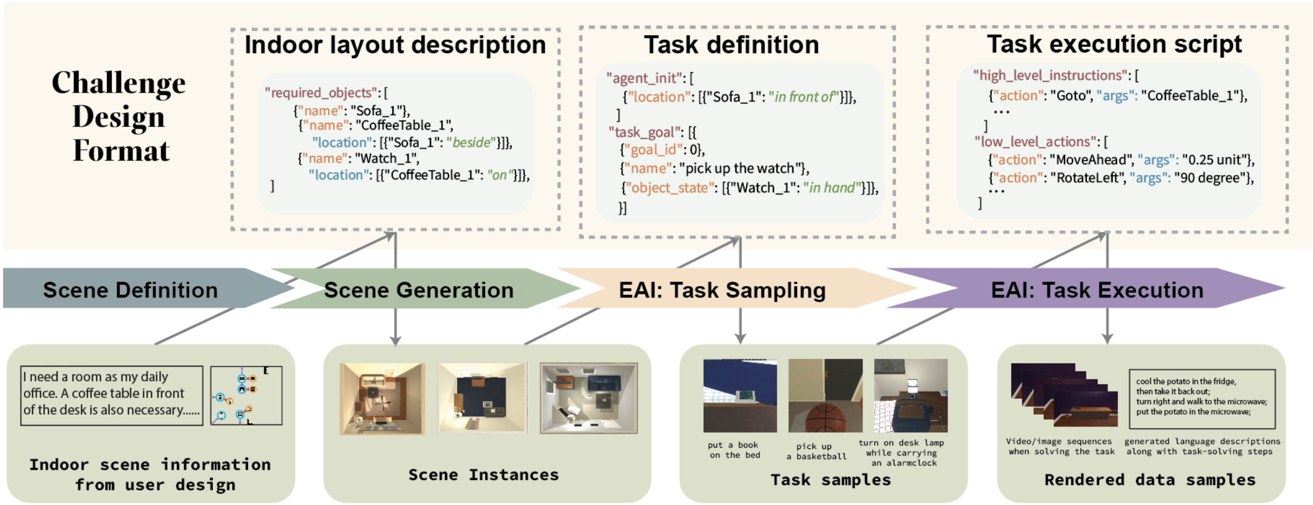
The main contributions of our work are threefold. First, we introduce a framework (LUMINOUS) which serves as a standard and unified benchmark for indoor scene synthesis algorithms. Second, LUMINOUS generates a large number of randomized scenes that achieve competitive quality compared to human-designed scenes in AI2Thor [KMH17]. Third, the rendered scenes, along with the multimodal trajectories, directly support typical EAI task completion to facilitate generalization research. Extensive evaluation on ALFRED [STG20], a language-guided task completion challenge, demonstrate the effectiveness and scalability of LUMINOUS. Further, our evaluation with LUMINOUS scenes suggests that existing, state of the art models for ALFRED may overfit to the hand-created scenes in AI2Thor.

2.2 Related Work

LUMINOUS builds on and extends research in indoor scene synthesis, simulation environments in EAI, and language-guided task completion.

Indoor Scene Synthesis. In computer graphics, extensive research exists in 3D indoor scene synthesis. Early work either used explicit rule-based constraints [XSF02] or incorporated stochastic priors into the generative procedure [YYT11, FRS12, FSL15, QZH18]. Recent advances [WLW19, LZW20, WYN20] utilize deep neural networks to extract patterns from large-scale datasets [SYZ17]. While these data-driven approaches significantly enhance the automation of the scene generation process, the resulting synthesized scenes are still relatively simple in terms of object quantity and inter-object spatial relationships. Many works generate scenes based on the natural representation of the scene graph [ZWK19, WLW19, LZW20]. Other lines of research condition on the image [WSC18, RWL19] or text [MPF18, CSM14] representation of indoor scenes. The discrepancies in the input representation of scene generation models and the diverse sources of data make it difficult to compare and contrast the performance of different methods. To facilitate research in learning-based approaches, LUMINOUS is designed to support end-to-end scene generation evaluation and a unified rendering tool to accommodate the outputs of various approaches simultaneously.

Embodied AI Simulators. In the past few years, researchers have developed many simulation environments [KMH17, GSA20, SXL20, PRB18, SKM19] to serve as training and evaluation platforms for embodied agents. These simulation environments propel research progress in a wide range of embodied tasks, including vision-and-language task completion [STG20, SBK20], rearrangement [WDK21, GSA20], navigation [SKM19, SXL20], manipulation [XQM20, JMA20] and human-robot collaboration [PRB18]. Recently, AllenAct [WSK20] integrates a set of embodied environments (such as iThor, RoboThor, Habitat [SKM19], etc.), tasks, and algorithms thereby facilitating the evaluation of the same model or algorithm across multiple EAI platforms. Many EAI platforms are designed with sophisticated indoor scenes to perform embodied tasks. Platforms such as iGibson [SXL20], AI2Thor [KMH17] can randomize materials, color, and small objects in the scene, while the basic room layouts



```
"required_objects": [
    {"name": "Sofa_1"},
     {"name": "CoffeeTable_1",
       "location": [{"Sofa_1": "beside"}]},
     {"name": "Watch_1",
       "location": [{"CoffeeTable_1": "on"}]},
 ]
```

```
"agent_init": [
  {"location": [{"Sofa_1": "in front of"}]},
 ]
"task_goal": [{
  {"goal_id": 0},
  {"name": "pick up the watch"},
  {"object_state": [{"Watch_1": "in hand"}]},
 }]
```

```
"high_level_instructions": [
  {"action": "Goto", "args": "CoffeeTable_1"},
  ...
 ]
"low_level_actions": [
  {"action": "MoveAhead", "args": "0.25 unit"},
  {"action": "RotateLeft", "args": "90 degree"},
  ...
 ]
```

I need a room as my daily office. A coffee table in front of the desk is also necessary......

cool the potato in the fridge, then take it back out; turn right and walk to the microwave; put the potato in the microwave;

Figure 2.2: The Luminous Framework. Scene definitions constrain generated scenes, which are pragmatically evaluated via household task sampling and execution to ensure generated scene quality.

remain unchanged. To facilitate more robust and thorough evaluation of embodied agents, LUMINOUS automatically generates indoor scenes with randomized layouts at a large scale that readily support vision-and-language navigation and high-level object interactions.

Language-Guided Task Completion. Among existing EAI challenges, we use ALFRED [STG20] as our downstream exemplar task to evaluate the scene generation quality of LUMINOUS. ALFRED enables agents to follow natural language descriptions to complete complex household tasks. ALFRED tasks involve resolving vision-and-language grounding, affordance-aware navigation, and high-level object interactions. Roughly speaking, there are two categories of approaches to tackling ALFRED. Initial approaches learned end-to-end models that mapped language instructions into low-level actions directly [SBK20, SGT21, PSS21]. Subsequently, hierarchical approaches [ZC21, BPF21] were proposed that enabled better generalization and interpretation. However, those approaches are only tested in four indoor scenes unseen during training time. Towards a more convincing evaluation, LUMINOUS generates an order of magnitude larger number of scenes for better assessment of generalization and robustness.

2.3 Luminous: a utility-driven scene generation framework

LUMINOUS bridges the fields of indoor scene generation and EAI task completion. A well-designed indoor scene needs to support different daily tasks. Accordingly, LUMINOUS generates an unlimited number of randomized layouts for EAI training and evaluation, while using the task success rate of an oracle planner as an automatic metric to evaluate the quality of the generated scenes.

2.3.1 Framework Overview

The scene generation pipeline of LUMINOUS consists of four stages, as shown in Figure 2.2. First, in the SCENE DEFINITION stage, users specify the required objects and, optionally, objects' relative spatial relationships. In the SCENE GENERATION stage, we propose a Constrained Stochastic Scene Generation (CSSG) algorithm to synthesize scenes whose layouts are randomized while satisfying user requirements and incorporating common sense knowledge to encourage scenes to be natural and functional. Next, the TASK SAMPLING stage programmatically samples household tasks that are executable in the current scene. Finally, the TASK EXECUTION stage plans a sequence of low-level actions for the agent to execute to complete the task, and generates a series of natural language instructions to describe the agent's behavior.

2.3.2 Challenge Definition Format

We introduce the Challenge Definition Format (CDF) to concurrently support the description of indoor layouts and the execution of household tasks (Figure 2.2). Learning-based indoor scene synthesis approaches are restrictive for generating EAI simulated environments [RCV21]. For example, these predict absolute locations for meshes, voxels, or point clouds for objects. By contrast, humans naturally understand the layout of an indoor scene in terms of the relative relationships among objects, such as a coffee cup on a table, a bed against a wall, and a chair in front of a desk. Recent scene synthesis algorithms such as Planit [WLW19] and 3D-SLN [LZW20] have demonstrated the effectiveness of using a directed graph to store the relative positions of furniture. Based on this insight, we argue that relative object relationships are more important than the absolute locations of objects for understanding the functional and intrinsic utility of the room. Anecdotally, we feel specifying scene layouts through relative object relationships is more flexible and user-friendly than absolute coordinates. In the indoor layout description section of the CDF, we define the required objects that must exist in the scene, including furniture, household items, and decorations, along with the relationship among those objects, for example that a book is on a table. Figure 2.2 shows an example of the indoor layout description. Each entry holds the name, type, or class of an item and may optionally have its spatial relation relative to another object. In addition, similar to 3D-SLN [LZW20], attributes such as color, material, and size can also be attached to an entry to further describe the object.

The CDF also contains of a task definition section and a task execution script. Instead of being specified by users, these sections can be automatically generated via the task sampling stage and the task execution stage. The task definition section specifies the task to be completed within the scene. The execution script lists out the action sequences for completing the task. Within the task definition section, inspired by Planning Domain Definition Language (PDDL) [HLM19, STG20], the CDF defines the initial state of the scene, comprising the position of the agent and the states of objects, and the conditions for task completion, for example that a desk lamp is toggled on. Figure 2.2 shows an example of an EAI task definition. The CDF can contain the execution script for the task in the form of human-understandable (high-level) instructions and atomic (low-level) actions.

2.3.3 Constrained Stochastic Scene Generation

To stochastically generate high-quality indoor scenes satisfying the layout constraints defined in the CDF, we propose a novel method: Constrained Stochastic Scene Generation (CSSG). Inspired by the energy-based indoor scene synthesis method [QZH18], CSSG generates scenes in a hierarchical manner, which enables great flexibility to enforce constraints and to incorporate prior knowledge. First, CSSG samples the room structure, such as walls, floors, and windows, from a set of pre-defined candidates. Next, CSSG samples types, positions, and rotations of large furniture defined in the CDF. During sampling, unlike human-centric indoor scene synthesis which learns the distribution of furniture from data, CSSG generates the distribution of the position and orientation of furniture according to *relationships* among furniture and room structure. Next, CSSG places objects in or on specific furniture, for example placing a coffee machine on a dining table. Finally, CSSG optionally generates decorations such as wall paintings

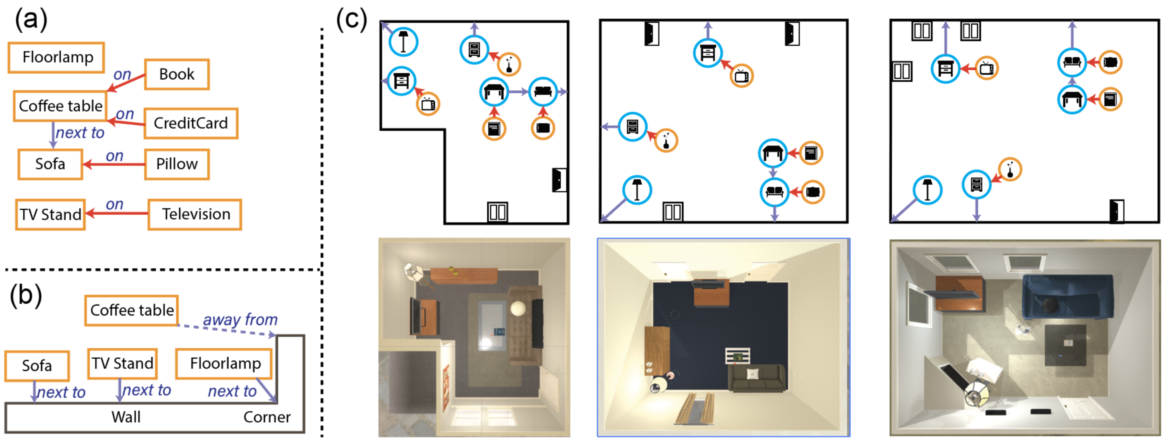


Figure 2.3: Constraint Stochastic Scene Generation. (a) explicit relationships defined in the CDF; (b) implicit relationships added by LUMINOUS; (c) sampled scenes satisfying relationships defined in (a) and (b) with different room structures.

and carpets.

Apart from the relationships defined explicitly in the CDF file, CSSG also integrates implicit relationships based on common sense. For example, if the CDF specifies "a bed is beside a reading desk", CSSG adds an implicit rule "the bed is against the wall" when sampling the position of the bed. When multiple relationships influence the position of an object, we use a set of predefined weights for different types of relationships. Experimental results (Section 2.4.1) show that the *rule-based* CSSG with predefined weights can reasonably balance human prior knowledge with the constraints specified in the CDF thus generating meaningful and functional indoor scenes. Therefore, LUMINOUS adopts CSSG as the default scene generation algorithm for EAI evaluation. We refer readers to Section 2.5 in the Appendix for details on implicit relationships, types of relationships, and predefined weights. Figure 2.3 illustrates the scene generation pipeline of CSSG and shows several sample scenes generated by CSSG, with more in Appendix Section 2.5.

2.3.4 Automatic EAI Task Sampling and Task Execution

Another challenge of using traditional indoor scene synthesis for EAI tasks is the lack of logic inherent to object interaction, state changes, and agent actions. It is unclear how to enable complex interaction capabilities within the framework of prior scene generation algorithms. To enable consideration of object interaction constraints, LUMINOUS is implemented on top of the interactive 3D platform AI2Thor [KMH17], which possesses 102 interactive object types, more than 2000 3D meshes, and most importantly: physical interaction mechanisms. We seamlessly connect the high-quality indoor scenes generated by CSSG and the sophisticated physical interaction logic provided by AI2Thor. LUMINOUS can thus directly support many complicated EAI challenges, including but not limited to ALFRED [STG20], Rearrangement [BCC20], and RoboTHOR [DHH20].

Given generated scenes, LUMINOUS can utilize the planner proposed in ALFRED [STG20] to sample solutions to simulation tasks. Additionally, given the tasks, LUMINOUS can resolve and generate appropriate scenes to support those EAI tasks. For details on task generation with ALFRED, see Section 2.3.6. Note that the task generation in LUMINOUS does not rely on ALFRED challenges. With the CDF used in LUMINOUS, we can easily sample an arbitrary number of simple tasks.

The task execution stage in LUMINOUS decomposes a household task into *navigation* and *interaction* tasks. *Navigation* requires the agent to find an optimal route from one place to another while avoiding collisions, which is achieved by a planner inside of LUMINOUS. *Interaction* often requires the agent to trigger the state change of certain object. For example, "taking a book on the coffee table" can be decomposed into the navigation part "go to a coffee table" and the interaction part "pick up the book". LUMINOUS applies Dijkstra's algorithm to get the shortest path for navigation, and AI2Thor's interaction mechanism to perform the agent-object interaction.

LUMINOUS provides two methods to generate natural language descriptions for household tasks involving navigation and object interactions. The first method relies on a rule-based language template to generate language instructions for different tasks (See Appendix Section 2.5). The second method uses the *Speaker* model proposed in Episodic Transformer [PSS21] that maps the low-level actions and corresponding egocentric images into generated language task instructions.

2.3.5 Accommodating Learning-based Indoor Scene Synthesis

Apart from the energy-based approach (CSSG), LUMINOUS incorporates two learning-based indoor scene synthesis methods, 3D-SLN [LZW20] and Deep-synth [WSC18], by training indoor-scene generators from the 3D-FRONT dataset [FCG20]. An obstacle that hinders the application of most learning-based methods to EAI tasks are object model discrepancies between the indoor-scene dataset and EAI simulators. LUMINOUS accommodates indoor scenes generated by 3D-SLN and Deep-synth by matching model names, furniture sizes, and room shapes between 3D-FRONT and AI2Thor, thereby providing a unified interface for learning-based approaches to train on the 3D-Front dataset and generated scenes with AI2Thor assets. For details, see Appendix Section 2.5.

2.3.6 LUMINOUS for ALFRED: A Comprehensive Example

We apply LUMINOUS to ALFRED, a benchmark for learning a mapping from natural language instructions and egocentric vision to sequences of actions for household tasks. The goal is to automatically generate additional data by LUMINOUS that shares exactly the same format as ALFRED training and evaluation data.

Given a trajectory T_i from the ALFRED training dataset, we employ a task parser to deduce objects and their relationships and save the scene conditions into the indoor-scene description part I_i of CDF. Since each training scene in ALFRED supports dozens of trajectories $\{T_i\}_{i=1,2,...}$, there may be some conflicting parts in their scene description $\{I_i\}_{i=1,2,...}$. For example, one task requires $\{Apple\_1\}$ to be on the countertop; another says $\{Apple\_1\}$ should be in the fridge. We propose a *merge* operator $merge(I_1, I_2, ...) \to \hat{I}$, where $\hat{I}$ denotes the merged links in indoor-scene description file, that tries to maximize common parts in the scene descriptions to tackle this problem. We use this merge operation for sampling indoor scene layouts S by CSSG. Since ALFRED does not change the positions of large pieces of furniture, such as fridges, sofas, and beds, the *merge* operator records the requirements for large pieces of furniture and extracts the most common criteria for small objects (e.g., apple, cup, and book). Figure 2.4 shows the comparison between AI2Thor original scenes and LUMINOUS scenes generated to augment the ALFRED challenge.

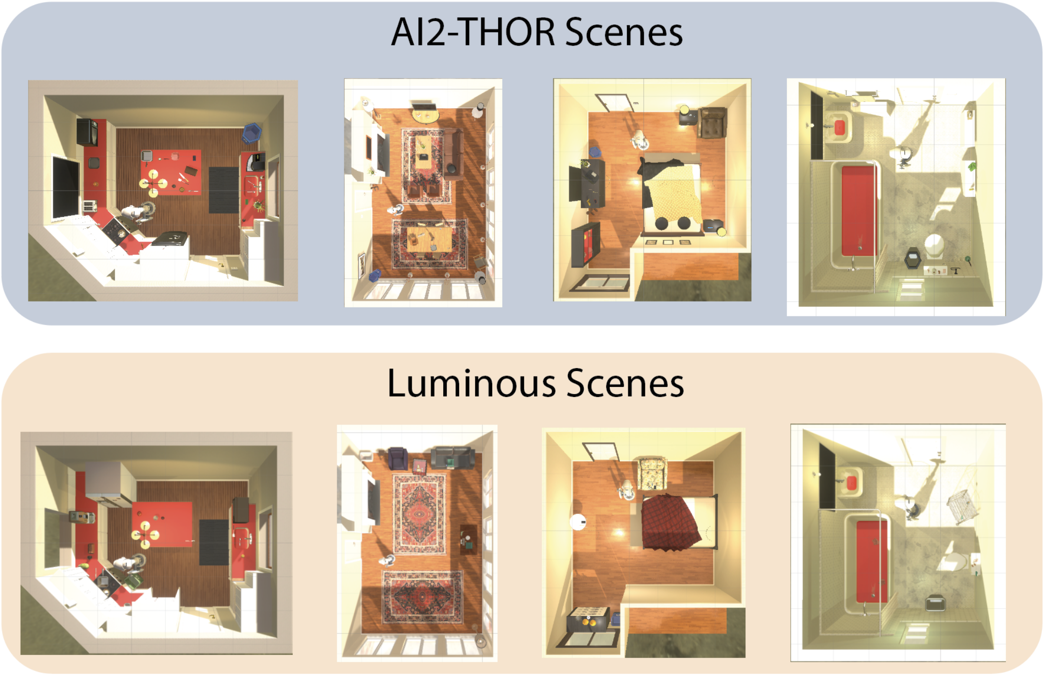


Figure 2.4: Sample AI2Thor and LUMINOUS scenes for EAI challenges. For kitchens and bathrooms, LUMINOUS keeps more parts of the room structures. See Appendix 2.5 for more details.

After obtaining an indoor scene S, we apply two techniques to sample tasks and trajectories. The first follows the Fast-Forward Planner (FF-Planner) [STG20] and samples tasks and trajectories by sequentially setting initial conditions, sampling task goals, and executing trajectories. The second follows the original task design D_i and directly applies the *task execution* component to generate the trajectory T_i'. Locations of small objects defined by I_i must be resampled for each task before execution.

The FF-Planner is slower at sampling tasks because it experiences trial and error in different sampling stages. We compare the efficiency of this method between sampling from AI2Thor original scenes and from LUMINOUS-generated scenes in Section 2.4.1. The sampling efficiency indicates the quality of the indoor scene. The second method samples trajectories much faster since it directly applies the task design D_i from original ALFRED training data which can be quickly solved by the TASK EXECUTION stage in LUMINOUS. We apply this method to generate a large number of scenes for the evaluation performance of different models in Section 2.4.2.

2.4 Experiments

We evaluate LUMINOUS both quantitatively and qualitatively. Our experiments focus on answering the following questions: 1) LUMINOUS *for indoor scene synthesis*: Does LUMINOUS generate high-quality scenes that are aligned with human common sense? 2) LUMINOUS *for EAI*: How well do the generated scenes support downstream EAI tasks? 3) *EAI task evaluation with* LUMINOUS: Can LUMINOUS generate indoor scenes that serve as reliable evaluation environments for EAI tasks? In addition, we discuss the insights obtained from the evaluation of state-of-the-art language-guided task completion models with larger set of unseen environments generated via LUMINOUS.

2.4.1 The Quality of LUMINOUS-generated Scenes

To answer the first two questions on evaluating the quality of LUMINOUS generated scenes from the perspective of both human common sense and the capability of supporting EAI tasks, we conduct user studies and oracle task success rate. We further demonstrate the great variety of tasks supported on scenes generated by LUMINOUS.

User Studies: Following the evaluation protocol proposed in [QZH18], we conducted user studies on Amazon Mechanical Turk comparing the quality of *Bedroom* scenes generated by LUMINOUS with two state-of-the-art learning-based approaches: Deep Priors [ZYM20] and 3D-SLN [LZW20]. Generated scenes are shown to users without any post-processing such as removing bad samples. Additionally, we compared LUMINOUS scenes against human-designed scenes in AI2Thor [KMH17]. Users were asked to evaluate scene quality, with scenes given as top-view images (Figure 2.4), based on two criteria: functionality and naturalness. Functionality describes how the room layout satisfies a human's needs for daily life. Naturalness indicates whether the room layout is realistic. Scales of responses range from 1 to 5, with 5 indicating perfect functionality or naturalness. For every scene, we collect three ratings per metric. The mean ratings and standard deviations are summarized in Table 2.1. LUMINOUS achieves competitive performance with the human-designed scenes in AI2Thor [KMH17]. We ran six Welch's unpaired, two-tailed t-tests to compare LUMINOUS scores with those of AI2Thor

	Method	Scenes, Ratings	Functionality (1-5)	*p*-value vs. LUMINOUS	Naturalness (1-5)	*p*-value vs. LUMINOUS
Generated	Deep Priors	50, 150	2.40 ± 1.40	$\sim$ **.0**	1.78 ± 1.06	$\sim$ **.0**
	3D-SLN	50, 150	2.45 ± 1.43	$\sim$ **.0**	2.03 ± 1.35	$\sim$ **.0**
	LUMINOUS	50, 150	**4.13** ± 1.00		**3.83** ± 1.11	
Human	AI2Thor	30, 90	4.23 ± 0.97	.416	3.68 ± 1.07	.308

Table 2.1: Human subjects' ratings of the functionality and naturalness of *Bedroom* scenes. LUMINOUS is rated statistically significantly better than existing, state-of-the-art generation methods.

and the learning-based approaches on both metrics. After a Bonferroni multiple-comparison correction, we find that LUMINOUS scenes are rated statistically significantly more functional and natural than scenes from both Deep Priors and 3D-SLN, the learning-based approaches, and not significantly differently from human-designed AI2Thor scenes.

Task Success Rate: Our proposed framework for indoor scene generation aims to promote better training and evaluation of the Embodied AI tasks. We show that, powered by the Constrained Stochastic Scene Generation strategy, LUMINOUS procedurally generates indoor scenes that can produce high-quality trajectories for downstream navigation and object manipulation tasks in a comparable level of efficiency even to the manually-designed scenes provided by the ALFRED [STG20] dataset. We adopt the same task sampling strategy as in the ALFRED dataset, which roughly samples 200 tasks for each of the 7 task types (Pick & Place, Stack & Place, Examine in Light, etc.) The tasks designed in the ALFRED dataset involve long-horizon navigation and object manipulations in indoor scenes and are very challenging such that even those sampled in the hand-designed scenes fail to be solved most of the time by a carefully-tuned Planning Domain Definition Language (PDDL) rule-based [AHK98] motion planner. Here we present the task success rate for a given set of scenes, defined as the percentage of tasks randomly sampled in the scenes that can be successfully solved by a rule-based, oracle planner. To make a fair comparison, we use the same sampling strategy and motion planner provided by the ALFRED dataset. As similar to the training fold in ALFRED, we construct 108 scenes by using LUMINOUS (26 scenes for each of the 4 room types). We compare the task success rate of these scenes with the rate of the manually designed scenes from AI2Thor [KMH17]. Our scene generation algorithm is automatic, and does not leverage knowledge of the motion planner in ALFRED that is tailored towards AI2Thor scenes.

	Task Success Rate	
	AI2Thor (Human)	Luminous (Generated)
Pick & Place	.33	.13 (Δ-.20)
Pick Two & Place	.10	.06 (Δ-.04)
Examine in Light	.55	.59 (Δ .04)
Clean & Place	.18	.17 (Δ-.01)
Heat & Place	.19	.09 (Δ-.10)
Cool & Place	.07	.07 (Δ .00)
Stack & Place	.05	.09 (Δ .04)
Overall	.21	.17 (Δ-.04)

	Subgoal Success Rate	
	AI2Thor (Human)	Luminous (Generated)
Heat Object	.19	.09 (Δ-.10)
Cool Object	.07	.07 (Δ .00)
Clean Object	.18	.17 (Δ-.01)
Slice Object	.11	.09 (Δ-.02)
Put Object	.15	.10 (Δ-.05)
Toggle Object	.55	.59 (Δ .04)
Pickup Object	.22	.18 (Δ-.04)
Goto Location	.21	.17 (Δ-.04)

Table 2.2: Task success rate and subgoal success rate.

			Trajectories per Task Type							
Split	**Scene**		Pick	Pick Two	Examine	Clean	Heat	Cool	Stack	**Overall**
Seen	AI2Thor	(S)	46	33	29	27	34	38	34	**251**
	Luminous	(S+)	226	167	236	210	163	202	201	**1405**
Unseen	AI2Thor	(U)	30	24	54	36	42	36	33	**255**
	Luminous	(U+)	27	18	178	56	21	56	79	**435**

Table 2.3: Validation Trajectory Counts by Task Type. ALFRED trajectories were sampled from both human-created AI2Thor scenes and generated Luminous scenes to evaluate EAI agents.

Subgoal Statistics: Scenes generated by Luminous support a large variety of (sub-)tasks introduced as "subgoals" in the ALFRED dataset. Each task in ALFRED consists of several subgoals ranging from navigation to object manipulations such as "SliceObject" and "ToggleObject". In total there are 8 types of subgoals and we calculate the statistics of these subgoals in tasks sampled from scenes as described above. See Table 2.2 (Right) for the comparison between Luminous and AI2Thor. This subgoal level evaluation further reveals appealing properties of Luminous. For example, Luminous achieves 17% task success rate in the GotoLocation subgoal, which indicates the generated scene has a comparable connectivity with human-created scenes in AI2Thor and the robot can move freely across a large portion of scene using a simple planner that does not account for held-object collisions.

2.4.2 LUMINOUS as an EAI Evaluation Platform

We use Luminous to provide two different settings to evaluate state-of-the-art inference models for the ALFRED challenge. All simulated scenes, trajectories, and task instructions are generated by Luminous. In the first setting, we use the room structures (the shape of floor, wall, and ceiling) in the *unseen* validation set of ALFRED, and then apply Luminous to randomize the scene layouts and sample the tasks and trajectories under the same room structures. For each of the four rooms' structures in the validation *unseen* set, we sample four room layouts and dozens of tasks. For each task, we sample one trajectory to solve the task. In total, we generate 16 indoor scenes and 435 trajectories. In the second setting, we randomly take 10 room structures in the *training* set of ALFRED for each room type (*Kitchen*, *Living Room*, *Bedroom*, and *Bathroom*). Then, with the 40 room structures, we randomize one layout and dozens of tasks for each. The second setting produces 1405 trajectories for evaluating EAI models, which is an order of magnitude larger than ALFRED *unseen* in terms of both task numbers and scene numbers. Table 2.3 summarizes the number of trajectories for each task type in ALFRED validation *seen*, *unseen*, and the two evaluation settings empowered by LUMINOUS.

With the aforementioned four test settings, we evaluate three top-ranked models for ALFRED challenge: MOCA [SBK20], Episodic Transformer (ET) [PSS21], and HiTUT [ZC21] on LUMINOUS validation settings. We denote the first validation setting as Unseen Plus (U+) and the second as Seen Plus (S+). For the validation performance of MOCA and HiTUT on ALFRED *seen* and *unseen*, we directly report their performance described in the paper. For the experimental results of ET, we evaluate its performance based on the checkpoints provided by the authors of ET.

In Table 2.4, we show the overall performance and per-task type's for MOCA, ET, and HiTUT. First, we found that the relative performance of the three models in our setting is generally consistent with ALFRED's overall generalization performance, where HiTUT achieves the best performance among the three models, and ET outperforms MOCA. It indicates that the models that perform well in the ALFRED challenge adapt to our randomized scenarios and tasks. However, comparing the evaluation results in unseen environments (U vs U+), there is a notable drop in generalization performance when we increase the number of test scenes from 4 to 16. This confirms that the current evaluation in ALFRED might not provide "true" generalization evaluation and highlights the significance of LUMINOUS for the embodied AI research. Second, we notice that the performance under S+ is similar to ALFRED *unseen* (U) in terms of large performance drop compared to ALFRED *seen* (S), even though the scenes and tasks generated by LUMINOUS share the same room structure (including walls, windows, doors,

	ALFRED Inference Model											
	MOCA				ET				HiTUT			
Task	S	S+	U	U+	S	S+	U	U+	S	S+	U	U+
Pick	.295	.131	.005	.429	.500	.227	.040	.381	.359	.314	.260	.259
Cool	.261	.000	.070	.000	.532	.035	.010	.018	.190	.035	.046	.034
Stack	.052	.000	.018	.000	.296	.025	.028	.000	.122	.065	.073	.038
Heat	.158	.000	.027	.000	.458	.000	.074	.000	.140	.061	.119	.000
Clean	.223	.000	.024	.000	.482	.129	.170	.109	.500	.229	.212	.232
Examine	.202	.000	.132	.000	.426	.072	.070	.034	.266	.173	.081	.067
Pick Two	.112	.011	.011	.000	.419	.034	.051	.000	.177	.096	.124	.111
Average	.186	.022	.038	.021	.448	.078	.066	.048	.252	.147	.124	.090

Table 2.4: Success rate on ALFRED tasks across validation splits. S: ALFRED *seen*; U: ALFRED *unseen*; U+ *Unseen Plus via* LUMINOUS; S+ *Seen Plus via* LUMINOUS. Note that all ALFRED models, in both *seen*- and *unseen*-based layouts, suffer loss of performance when generalizing to generated LUMINOUS scenes for nearly every task.

etc.) with scenes in ALFRED's training. The randomized layouts from LUMINOUS that produce different locations of objects introduce extra difficulties for the models to accomplish tasks. It is worth noting that the high success rate of Pick tasks is due to LUMINOUS place the object in the edge of receptacles (e.g., table, shelf, sofa, etc.). This provides a broader range of areas for the robot to pick up the objects and thus leads to a much higher success rate than other task types.

2.5 Appendix

Details on incorporating Learning-based Indoor Scene Synthesis

As we shown in Figure 2.2, the overall structure of LUMINOUSmainly consists of three components. First, we propose a unified representation of indoor scene processing, providing various interfaces for data processing, making the original data in different formats required by different models: e.g. RGB images, bounding boxes with object types, etc. After that, different data formats are used as inputs to different models for training indoor scene generation models. It is worth noting that we unify the model-generated scene formats again, allowing us to use the same scene rendering tools to automatically visualize the scenes. Finally, we provide different testing interfaces to uniformly evaluate the quality of various algorithm-generated scenarios.

Data processing: Since our ultimate task is to provide indoor scenes as experimental environments for Embodied AI, the data we target should provide a full set of information about the indoor scenes: e.g., house structure, furniture models, and object placement information. Luminous selects three data sources for data processing: mesh information from 3D-FRONT [FCG20], and game designs from AI2Thor [KMH17]. In the data processing, we first unify the names of items in different datasets (e.g. *picture* $=$ *painting*, *bedside cabinet* $=$ *nightstand*). The full list of unified furniture and object names are attached in the appendix. Then we normalize the coordinated w.r.t. locations and rotations. We also normalize room scales. Finally, according to different formats of the training data for different methods, we generally provides three different data formats: RGB-D images, semantic segmentation, and bounding boxes together with object types and rotations.

Scene Synthesis: Luminous provides some state-of-the-art algorithms for indoor scene synthesis. We chose Python as programming language ,and Pytorch for deep learning. We have carefully referred to the source code of these these methods. However, for the reason such as missing public training dataset, and the compromise we have made for unifying data formats (e.g. *double bed* $\rightarrow$ *bed*), the re-implemented performance in Luminous for those methods may differ from the original one.

Constrained Stochastic Scene Generation

We consider the problem of indoor scene generation under certain constraints represented by text descriptions [MPF18] or scene graphs [LZW20]. In our baseline, each constraint not only defines the type of an object, but also optionally describes the object's relationship with others in the scene. In detail, a constraint c_i provides the information for placing object i by defining its type o_i (e.g. *bed*), and a set of relationship with others $R_i = \{rel(i, j_k)\}_{k=1,2,...}$, where j_k stands for another object in the scene and $rel(\cdot,\cdot)$ specifies the relationship between two objects (e.g. *bed* ***beside*** *window*).

Given a set of constraints $\{c_i\}_{i=1,2,...}$ and the room structure (the shape of floor, wall and ceiling), an indoor scene is sampled from a sequential process of three layers. The first layer samples pieces of **furniture** that represent the overall function of the room and can be placed directly on the floor, such as *bed, dinning table*, and *refrigerator*. The second layer samples **objects** that are usually supported by another piece furniture such as *book, pen*, and *coffee* *machine*. Finally, the third layer samples **decorations** in the scene such as *painting* and *carpet*.

In each layer, we empirically defined the priority value $q(i)$ as the order for placing furniture according to object types. For example, we prefer to place *desk* before placing *chair*: $q(desk) > q(chair)$. Besides, we limit the constraints that can be represented by a direct acyclic graph (DAG) and resolve the relationship between objects to ensure that when calculating $rel(i, j_k)$, we have $q(i) > q(j_k)$. For example, if the text description says *a desk is in front of a chair*, it is resolved as *a chair faces a desk*.

When placing each object, we samples the position and rotation of the object by its explicit relationship with others $\{rel(i, j_k))\}_{k=1,2,...}$ defined previously, and implicit relationship with others $\{\widetilde{rel}(i, j_k))\}_{k=1,2,...}$ predefined heuristically from our prior knowledge. For example, humans are in favor of pushing the *bed* up against the *wall* of a *Bedroom* $(bed, (wall, against))$.

Each relationship $rel(i, j_k)$ generates a vector field in space: each position p is characterized by $(s_{p,k}, r_{p,k})$, where $s_{p,k}$ is the score of the point. s_i depends on the distance d_i between p and the target object j_k. Figure 2.5(a) shows different types of relationship and the scores deduced by the relative distance. $r_{p,k}$, suggesting the relative rotation of placing the object, depends on the direction vector from p to its target j_k the type of relationship. Combining $s_{p,k}$ with parameter $w_{type(rel(i,j_k))}$ related only the type of relationship, we sample the position to place object i according to weighed sum of scores among all relationship, and the rotation of the object at position p is defined by the type of relationship which has the largest weight.

$$s_p = \sum_k w_{type(rel(i,j_k))} \tag{2.1}$$

$$P(p|R_i) \propto \exp(-s_p) \tag{2.2}$$

$$r_p = r_{p,k'} \quad k' = \arg\max\{w_{type(rel(i,j_k))}\} \tag{2.3}$$

Comparison between CSSG and advanced indoor scene generation algorithms

In Table 2.5, we summarize the properties of CSSG and other indoor scene algorithms. As the table shown, the state-of-the-art scene generation algorithms use SUNCG dataset [SYZ17] as

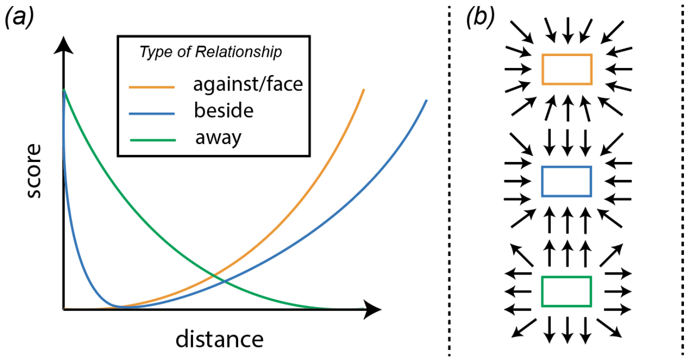


Figure 2.5: Illustration of how to sample the position of an object according to the type of relationship. (a) Score functions for different types of relationship, depended on the distance between the sampling position p and the target object j_k. (b) Direction vectors suggesting the rotation $r_{p,k}$ of the object being placed on the position.

training , is not currently not available. It is hard to reproduce the results from those approaches. In LUMINOUS, we reproduce the learning based approaches such as 3D-SLN [LZW20] using publicly available dataset (3D-FRONT [FCG20]) for training. We believe this could serve as first step to provide a unified benchmark for comparing indoor scene generation algorithms.

Algorithm	**Scene graph Inference?**	**Constrained?**	**RGBD rendering?**	**Dataset?**
PlanIT (2019)	✓	✓	✓	**unavailable**
Grains (2018)	**N/A**	**N/A**	✓	**unavailable**
3D-SLN (2020)	**N/A**	✓	✓	**unavailable**
Human-centric (2019)	**N/A**	**N/A**	✓	**unavailable**
Luminous CSSG	✓	✓	✓	**N/A**

Table 2.5: Comparison of CSSG and state-of-the-art indoor scene generation algorithms. Scene graph inference refers to the algorithm's ability to infer the latent scene graph of the indoor scene. Some of the algorithms support taking scene graphs as constraints. The dataset for training the indoor scene synthesis model is missing due to legal issues.

g

t5bdImplicit relationships between furniture We list the implicit relationships when sampling the position of the furniture. Basically, the relationships can categorizes into two types: furniture v.s. room structure, and furniture v.s. furniture.

- furniture v.s. room structure: (CounterTop, against, wall border), (TVStand, against, wall border), (Sofa, against, wall border), (border, against, wall border), (Bed, against, wall

High-level action	Instruction candidates
GotoLocation	go to, find, walk to
PickupObject	pick up, take, carry
PutObject	put, place
SliceObject	slice, cut
CoolObject	chill, cool
HeatObject	heat, cook
CleanObject	clean, wash, rinse
ToggleObject	turn on

Table 2.6: Language template: mapping high-level actions to language instructions

border), (Dresser, against, wall border),(Desk, against, wall border),(SideTable, against, wall border),(FloorLamp, against, wall corner), (DiningTable, away from, wall border)

- furniture v.s. furniture: (Chair, face, Desk), (Stool, face, DiningTable), (CoffeeTable, beside, Sofa), (DiningTable, away from, Sofa)

If multiple relationships influence the distribution of the sampling position of an object, we give the weight coefficient as 2.0 if the relationship is from *furniture v.s. room structure*, and as 1.0 if the relationship is from *furniture v.s. furniture*.

Task Instructions Generation

Unlike ALFRED, LUMINOUS obtains the natural language as high-level instructions from an automatic pipeline instead of human annotations.

We design a language template to generate natural language instructions corresponding to the high-level instructions in ALFRED. Table 2.6 shows mappings from high-level action to language instructions. The natural language instruction is generated as:

$$[instruction\ candidate] + [object\ name] + [attribute]$$

Where the attribute specifies the receptacle for *PickupObject* (e.g., pick up an apple *in the fridge*), or the target location for *PutObject* (e.g., put a book *on the table*).

However, the language instruction for navigation can be too simple and vague if we just say *go to* some place. We apply the *Speaker* provided by ET to generate task instructions, especially for the navigation part. The training data come from the ALFRED dataset. The input of the *Speaker* is the low -level action sequence (e.g. *MoveAhead*, *MoveAhead*,*RotateLeft*) and images from the egocentric view the agent, and the output is a piece of natural language instruction.

$$(low\ level\ actions, images) \xrightarrow[\text{Speaker}]{} (language\ instructions)$$

We refer readers to ET [PSS21] for model details and put the generated examples in Appendix 2.5

Illustration of ALFRED and LUMINOUS

In this part, we illustrate the details when we apply LUMINOUS for ALFRED challenge.

Task parser

The task parser is applied to deduce the indoor scene description I_i for an ALFRED trajectory T_i. Specifically, the task parser would go through the low-level actions in T_i, and

- extract the $action\ args$ as required objects from actions including *GotoLocaiton*, *PickupObject*, *ToggleObjectOn*, and *OpenObject*. For example, if the $action\ args$ of *GotoLocaiton* is *DiningTable*, the task parser put *DiningTable* into the list.
- extract the $action\ args$ of *PickupObject* as scene constraints. For example, picking up an apple on the fridge means that initially *Apple* is in the *Fridge*.

Indoor scene sampling

For room structures of living rooms and bedrooms, LUMINOUS only keep *wall*, *ceiling*, *floor*, *window* and *door*. For room structures of kitchens and bathrooms, LUMINOUS further keeps *CounterTop*, *Sink*, *Cabinet*, and *Oven*, and *Bathtub*. Figure 2.6, 2.7, and 2.8 plot the scenes of different room types sampled by LUMINOUS.

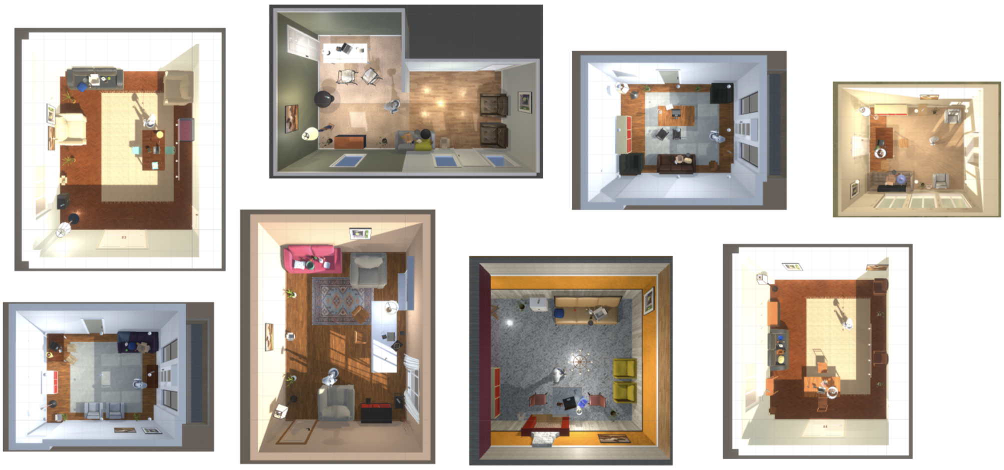


Figure 2.6: Living rooms sampled by LUMINOUS

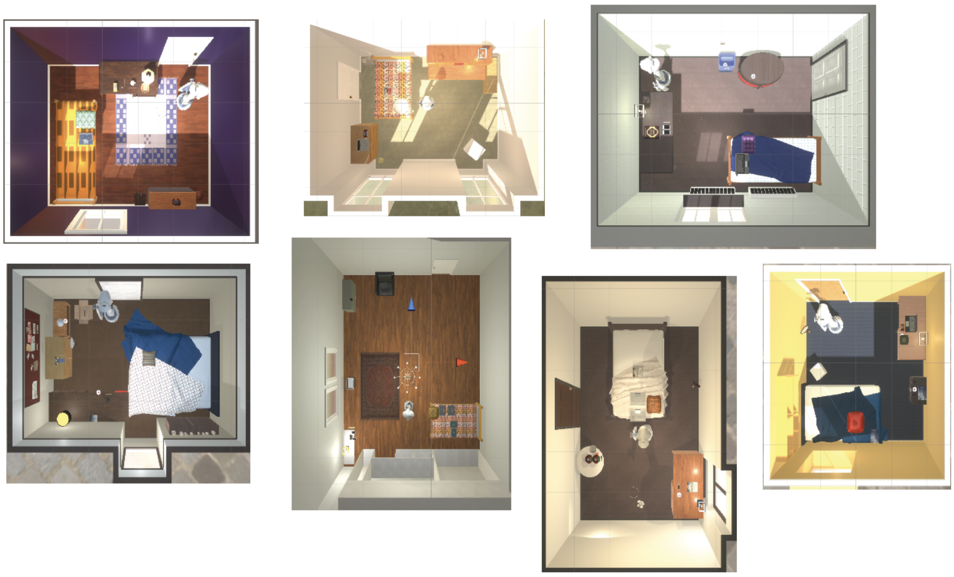


Figure 2.7: Bedrooms sampled by LUMINOUS

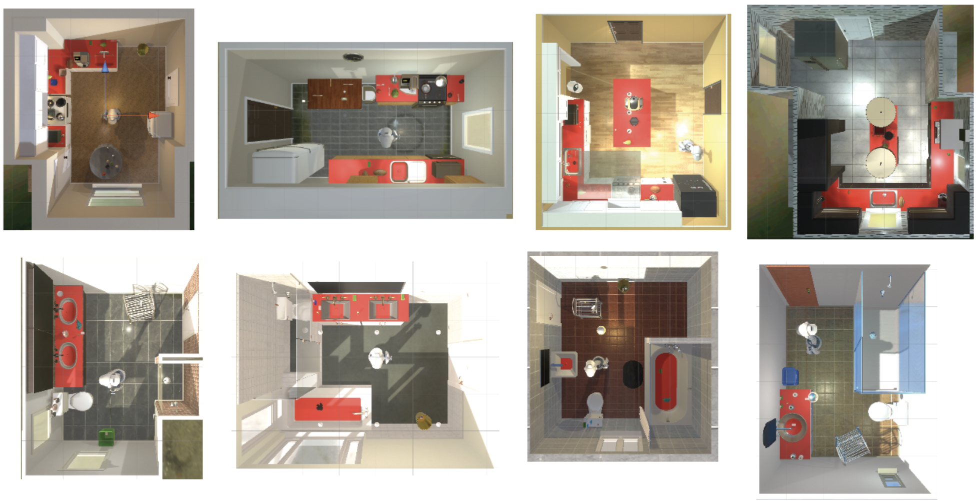
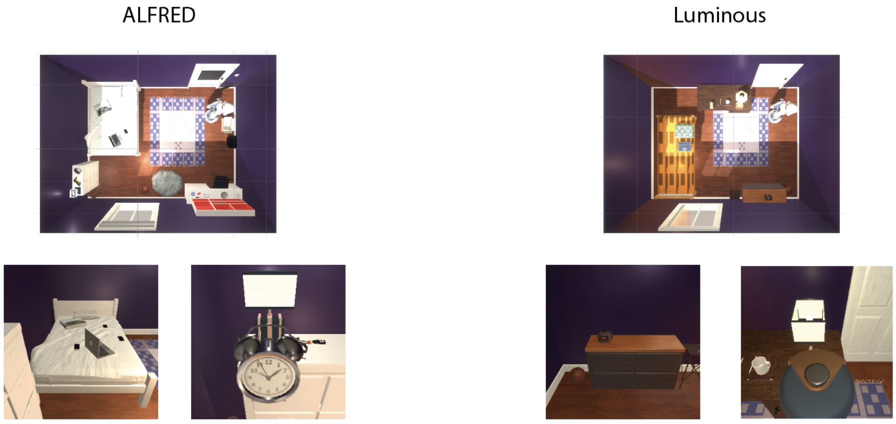


Figure 2.8: Kitchens and bathrooms sampled by LUMINOUS



"Turn left and face the dresser.",
"Pick up the alarm clock from the dresser.",
"Turn left, look and then face the lamp.",
"Turn the lamp on."

"turn left and walk to the small black table in front of you .",
"take the alarm clock.",
"turn around and walk to the small white table on the left .",
"turn on the lamp on the table ."

Figure 2.9: Comparison between ALFRED and LUMINOUS generated trajectories.

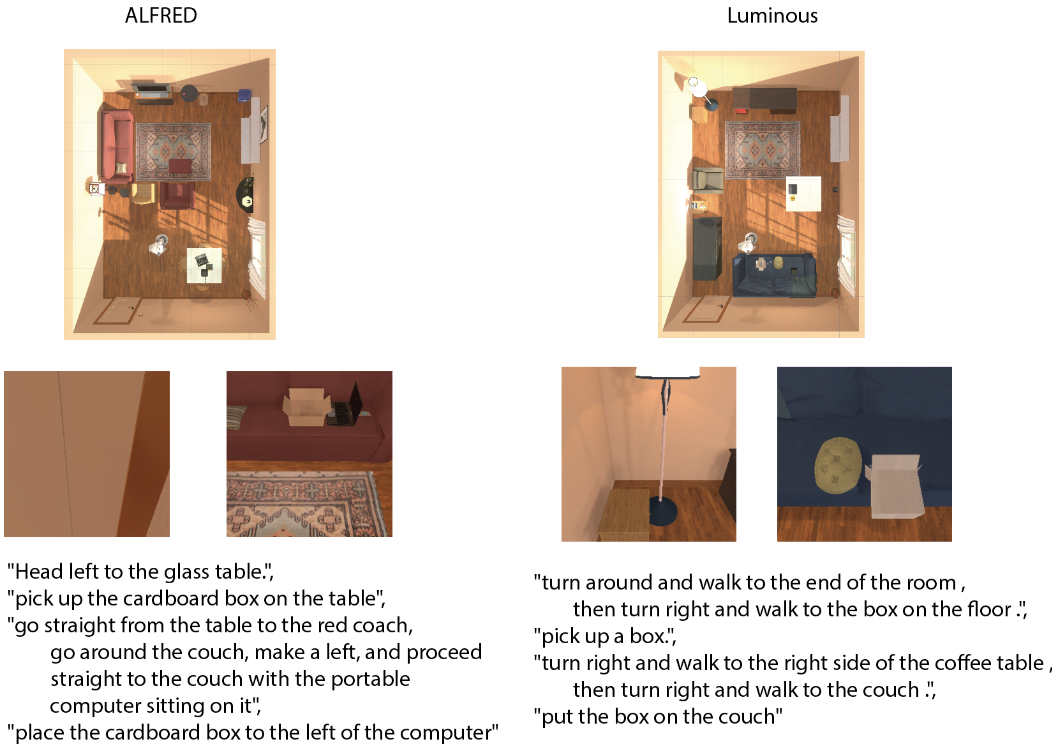
ALFRED trajectories v.s. LUMINOUS trajectories

We performance side by side comparison between ALFRED trajectories and LUMINOUS trajectories in Figure 2.9 and 2.10. We plot the scene layouts, initial camera images, images after task completion and language instructions for both.

Hard task analysis: Heat & Place / Cool & Place

We notice the low success rate for two types of tasks: *Heat & Place* and *Cool & Place* in LUMINOUS scenes. The *Cool* operation requires a fridge and the *Heat* operation needs a microwave. We compare the layout w.r.t. the fridge and microwave between AI2Thor scenes and LUMINOUS scenes, and we find a somewhat different set-up for them. Figure 2.11 compares the locations of the fridge and microwave. Since AI2Thor scenes are manually designed.

- In the task sampling stage (Table 2), the FF-Planner samples task trajectories from ground-truth knowledge of the environment and would not be influenced by visual discrepancies between ALFRED and LUMINOUS.
- In the EAI evaluation stage (Table 4), the EAI agent takes the input as RGB images and images look visually different between manually designed scenes and synthesized scenes, making the agent harder to complete heat and cool tasks.



"Head left to the glass table.",
"pick up the cardboard box on the table",
"go straight from the table to the red coach,
go around the couch, make a left, and proceed
straight to the couch with the portable
computer sitting on it",
"place the cardboard box to the left of the computer"

"turn around and walk to the end of the room ,
then turn right and walk to the box on the floor .",
"pick up a box.",
"turn right and walk to the right side of the coffee table ,
then turn right and walk to the couch .",
"put the box on the couch"

Figure 2.10: Comparison between ALFRED and LUMINOUS generated trajectories.

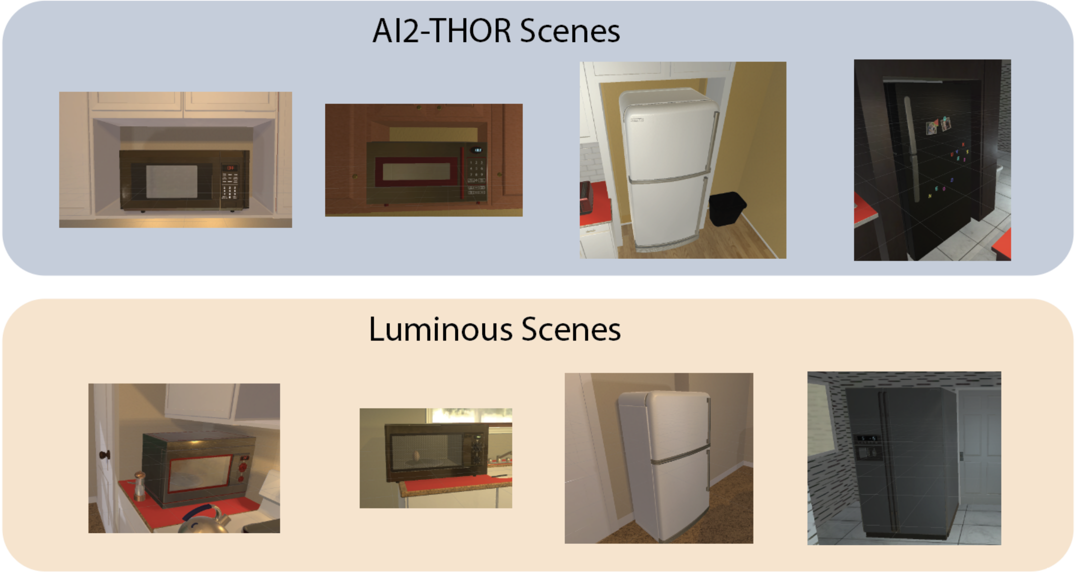


Figure 2.11: Different locations of the microwave and fridge in AI2Thor scenes and LUMINOUS scenes. In AI2THOR, most microwaves and fridges are embedded in the structure of the room; in LUMINOUS, microwaves are preferred to be placed on a countertop and fridges most likely locates in a relatively open area. Such difference brings different visual experience to EAI agents.

Challenge	Navigation?	Interaction?	Language understanding?	physics understanding?
ObjectNav (habitat, ai2thor)	YES	NO	NO	NO
Multi-On/Rearrangement (habitat, ai2thor)	YES	PART OF	NO	YES
InteractiveNav (iGibson)	YES	YES	NO	YES
ALFRED (ai2thor)	YES	YES	YES	YES

Table 2.7: Comparison between ALFRED with other EAI tasks. Different simulators may have different requirements to EAI agents including navigation (to navigate an agent from one place to another), interaction (to interact with an object in the environment), language understanding (to follow language instructions from users), and affordance or physics understanding (to gain some knowledge for the affordance map in the scene).

Large-Scale Evaluation Experiments

In this section, we conduct an additional large-scale evaluation with respect to the number of scenes. We generated 216 scenes with the same room structure as training scenes in ALFRED (including walls, floor, and windows) but randomized layouts and objects as the evaluation environments for ALFRED-like tasks. We summarize the statistics of our evaluation datasets and performance of three state-of-the-arts in Table 2.8. The second column presents the number of unique configurations (including room layouts, small object locations) of tasks in each task type. The third column shows the number of unique scenes/layouts (same room layout with different small object locations count as the same scene). Comparing the results in Table 2.4 and Table 2.8, the success rate in $S+$ column evaluated by 40 scenes and 216 scenes maintain the similar relative performance. Based on the above observation, we further strengthen our conclusions obtained in Section 2.4.2 that LUMINOUS can provide more robust and consistent evaluation results.

			ALFRED Inference Model					
			MOCA		**ET**		**HiTUT**	
Task	**# Trajs**	**# Scenes**	S	S+	S	S+	S	S+
Pick	1124	192	.295	.139	.500	.205	.359	.296
Cool	885	44	.261	.000	.532	.009	.190	.043
Stack	1002	126	.052	.002	.296	.028	.122	.058
Heat	786	54	.158	.000	.458	.005	.140	.061
Clean	923	98	.223	.000	.482	.109	.500	.232
Examine	1263	84	.202	.000	.426	.056	.266	.124
Pick Two	944	168	.112	.013	.419	.034	.177	.097
Overall	7074	-	.186	.025	.448	.068	.252	.137

Table 2.8: Success rate on ALFRED tasks. # Trajs: number of unique task configurations; # Scenes: number of unique scene layouts in each task type; S: ALFRED *seen*; S+ *Seen Plus via* LUMINOUS.

Chapter 3

Utility-Driven Door-opening Benchmark

We present OPEND, a novel benchmark that teaches hand-controlled cabinet doors or drawers opening using a physics-based simulation environment. The simulation is designed to respond to natural language commands and incorporates detailed physical collision, sophisticated hand friction, and randomized environmental factors. To achieve this challenging task, we propose a multi-step planner consisting of a deep neural network and rule-based controllers. The network extracts spatial relationships from images and interprets the semantics of natural language instructions, while the controllers execute the plan based on this spatial and semantic understanding. We evaluate our system by measuring its zero-shot performance on a test dataset. Our experimental results demonstrate the effectiveness of our multi-step planner for different hand configurations, while highlighting the challenges posed by language understanding, spatial reasoning, and long-term manipulation. We plan to release OPEND and host challenges to encourage further research in this exciting area.

3.1 Introduction

Embodied AI research [DYT22] is playing a crucial role in advancing intelligent robotic systems. Thanks to the development of simulation engines, the availability of manipulation benchmarks [ZCJ22, MLX21, STG20, EHH21, MXW21] has sparked the emergence of new models and algorithms that help bridge the domain gap between virtual and physical spaces, thereby improving robotics research in object manipulation.

Although existing benchmarks in simulation aim to cover a wide range of tasks, they often

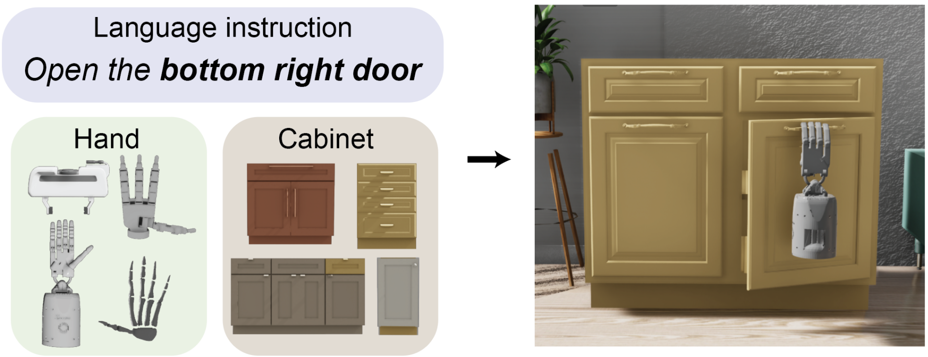


Figure 3.1: Problem setting. In a simulated scene, the task is to open a cabinet door or drawer by hand corresponding to the language instruction and the camera image. OPEND provides 174 different cabinets, 372 pieces of language instructions, and 4 types of hands.

over-simplify the tasks themselves. This simplification is typically achieved by altering object configurations [MLX21, ZCJ22], reducing the complexity of collisions [STG20, SLL22], or abstracting away the intricacies of grasping [STG20, EHH21]. Especially, by assuming binary or discrete object and robot states [SCU21, SLL22, EHH21], the system is not required to carefully reason about object geometry and physical laws. Furthermore, training in a simple, background-free environment [KT15, LWO20] restricts the learned model's potential for real-world applications.

To address the limitations of current benchmarks, we introduce OPEND, a benchmark designed to facilitate learning of hand manipulation skills for mobilizing articulated objects using both visual and language inputs. Our benchmark focuses on a single task: opening a cabinet (via its door or drawer), and offers a high degree of realism and scalability in terms of both visualization and physics.

Figure 3.1 shows that OPEND can import high-topology objects with diverse geometries and high-quality room backgrounds. Currently, the software includes hundreds of cabinets that have been automatically processed, as well as several manually designed rooms that feature randomized floor and wall materials. To accurately simulate physics, OPEND takes into account both collisions and friction between the robot hand fingers and the cabinet, without simplifying their geometries. This poses a significant challenge, as four typical robot hands are provided for the manipulation task, which requires controlling the hand via all its joints. In addition, OPEND aims to promote language-instructed learning by generating hundreds of instructions that are

Benchmark	Realistic grasping	Various cabinet type	Realistic cabinet size	6-DOF hand	Various robot/hand type	Photo-realistic background	Multi-task
ManipulaTHOR [EHH21]	✗	✓	✓	✓	✗	✓	✓
VLMBench [ZCJ22]	✓	✗	✗	✓	✗	✗	✓
ManiSkill [MLX21]	✓	✓	✗	✓	✗	✗	✓
Calvin [MHR22]	✓	✗	✓	✓	✗	✗	✓
Robomimic [MXW21]	✓	✗	✓	✓	✗	✓	✓
OpenD	✓	✓	✓	✓	✓	✓	✗

Table 3.1: Comparison with other manipulation benchmarks.

automatically parsed from the cabinet's geometry. This presents another concurrent challenge: correctly understanding and interpreting the language instructions.

Our goal is to tackle the challenges of long-horizon planning and high-dimensional control with a model that is both efficient and stable. To achieve this, we propose a baseline model that utilizes a multi-step planner, which combines network training and motion planning. Specifically, our network integrates the strengths of Faster R-CNN [RHG15] and CLIPort [SMF22] for recognition tasks. In combination with a set of pre-defined controllers, our baseline model offers precise control of hand joints.

We evaluate our approach on a previously unseen dataset where the cabinet is not revealed during the training or validation phases. Only the image of the cabinet captured by a camera and the corresponding language instructions are provided to the model. Experiments show that the models trained for different hands produce varied success rates on test data ranging from 14.6% to 30.3%. The results suggests that our baseline can help perform long-term manipulation to control the hand and fingers to open the target cabinet.

3.2 Related work

Simulation Environment. There is a large body of work on simulating indoor household activities for training and evaluating AI agents [KMH17, PRB18, LXM21, SCU21]. Most of these simulators emulate high-level instructions and post-effects of agent behaviors using simplified state and action representations. Some use simplified abstract discrete action spaces [STG20] which can reduce the task's difficulty. However, models trained in this setting without any awareness of the low-level geometry and dynamics of objects are limited in their ability to transfer to real-world applications. For example, grasping is often simplified by attaching a nearby item to the gripper [EHH21, STG20, SLL22, SCU21]. In contrast, in this work, hands are controlled at the joint level, and friction-based grasping is powered by a state-of-the-art physics engine, PhysX 5.0 [NVI22b].

Manipulation Task: Mastering the manipulation skills of a robot typically requires an understanding of vision, language, and robotics. This field has recently received much attention across different disciplines. In addition to applying imitation learning or reinforcement learning to train a robot to grasp and manipulate objects [MHR22, MLX21], recent studies have proposed end-to-end networks that can learn skillful controls, which require precise spatial reasoning or language understanding [SMF22].

Learning a model from image and language input in a simulation environment with continuous states is still considered challenging [DYT22, MHB22], as the simulation engine needs to provide constant rendering and simulation results. Therefore, for manipulation benchmarks, compromises are often made by giving the model full knowledge of the environment and objects to make training easier [MHB22], reducing the difficulty of grabbing items [STG20], and providing a limited number of items without varying their materials or backgrounds for simpler evaluation [ZCJ22, MHR22].

Our work simulates the continuous states of articulated objects and uses camera sensing and language instructions as model inputs. Without relying on abstract grasping or object resizing, we train our hands to open the cabinet by hybridizing the end-to-end network with the motion planner. The advantage of the neural network lies in its powerful spatial reasoning capability, and the motion planner helps ensure stable performance based on this reasoning.

Language Guided Manipulation. Relating human language to robot actions has been of interest in recent research [MHR22, LS20, STG20, GGG22, SGT21]. Natural language presents specification, providing an intuitive way to refer to abstract concepts concerning spatial, temporal, and causal relationships.

We focus on the language that describes the object type and spatial relationships in our task. Due to the presence of multiple interactive parts on the cabinet, language plays a crucial role in addressing ambiguity.

3.3 Simulation environment

In this section, we will describe the setup of our challenge, which involves retaining a wide range of simulated hands, hundreds of drawers and cabinets, and a variety of scenes with randomized materials and backgrounds.

Engine. We selected OMNIVERSE [NVI22a] as the ideal platform for developing the OPEND challenge due to its advanced simulation capabilities. OMNIVERSE provides efficient and reliable simulation of rigid bodies, soft bodies, articulated objects, and fluids. Additionally, the platform offers a Python scripting environment, which enables easy integration of open-source and third-party Python libraries. Moreover, OMNIVERSE's advanced ray tracing technology allows for the creation of impressive rendering effects.

Assets. Our challenge utilizes the cabinets from the SAPIEN PartNet-Mobility dataset[XQM20]. Comprising of rigid bodies, robots, and articulated objects, the dataset offers 346 storage furniture pieces as cabinets, which are well-detailed with rendering material. For creating photo-realistic simulation backgrounds, we designed nine synthetic indoor scenes with randomized lighting, floor materials, wall types, and decorations. Each scene is manually crafted to ensure its uniqueness and authenticity.

Hand. Robotic hands, which are typically programmable, are designed to mimic the functionality of human hands. In the simulation environment, OPEND offers control for four representative robot hands, including commercially available options such as the *Franka gripper*, *Allegro hand*, and *Shadow hand*. Additionally, the *Skeleton hand* is a modeled version based on the biological structure of the human hand.

In OMNIVERSE, hand modeling and rigging is achieved through the use of three types of joints. Figure 3.2 provides an overview of this process. The prismatic joint enables two bodies to slide along a shared axis, while the revolute joint permits rotation along a common axis. Finally, the D6 joint is utilized specifically for hand rigging, allowing for rotation along the y-axis and z-axis while restricting movement along the x-axis.

Table 3.2 provides a comprehensive list of the joint components and corresponding degrees of freedom required to control the fingers of each hand. In addition, six degrees of freedom are needed to regulate the position and rotation of the hand root (also known as the articulation root). To fully describe the state of the hand, we consider its position (p), rotation (r), and joint positions ($d = \{d_i\}$).

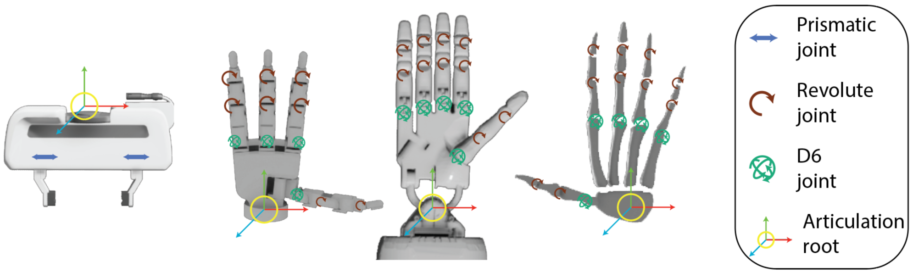


Figure 3.2: Hand riggings. From left to right: Franka gripper, Allegro hand, Shadow hand, and Skeleton hand.

Table 3.2: Hand joint components and degrees of freedom.

	Prismatic joint	**Revolute joint**	**D6 joint**	**DoF**
Franka gripper	2	0	0	2
Allegro hand	0	8	4	16
Shadow hand	0	10	5	20
Skeleton hand	0	10	5	20

3.4 The task

The goal of OPEND is to accurately open a cabinet drawer by a specified distance of δ meters or rotate a cabinet door by a specified angle of θ degrees. This is achieved by utilizing a camera placed in front of the cabinet and a language instruction that describes the desired opening action.

Therefore, the task is to plan the movement of the hand over time, given an image I and language instruction S:

$$(I, S) \rightarrow (p_t, r_t, d_t)$$

Camera setup. To capture a full view of the cabinet, a camera has been positioned in front of it with a slight offset. The resulting image is in full color, using an RGB format. Additionally, the camera also generates a depth map that specifies the exact distance between the cabinet and the camera.

Language instruction. Since a cabinet may have multiple doors and drawers, the language instruction specifies what and where to open. We apply Algorithm 1 to generate template language as the spatial description for the drawers and doors on one cabinet.

The key idea is to perform an iterative comparison between the target positions vertically and horizontally. However, the algorithm may not be effective for cabinets with an excessive number of doors and drawers. To keep our instructions simple and manageable, we exclude these more complex cabinets from our analysis. Figure 3.3 lists some generated instructions.

Algorithm 1 Generate language instructions for one cabinet

```
Require Obtain the number of drawers and doors $n \geq 0$, and their positions $\{u_i \mid u_i = (x_i, y_i, z_i)\}_{i=1,2,...,n}$
(1) For $i = 1, 2, 3, ..., n$
(2) If $i \geq 2$
Compare $y_i$ with $\{y_1, ..., y_{i-1}\}$;
Update description for $\{u_1, ..., u_i\}$ with {left, second left, middle, second right, and right}
Compare $z_i$ with $\{z_1, ..., z_{i-1}\}$;
Update description for $\{u_1, ..., u_i\}$ with {top, second top, middle, second bottom, and bottom};
(1) If find the same description for two positions
return invalid cabinet.
return descriptions for $\{u_1, ..., u_n\}$
```

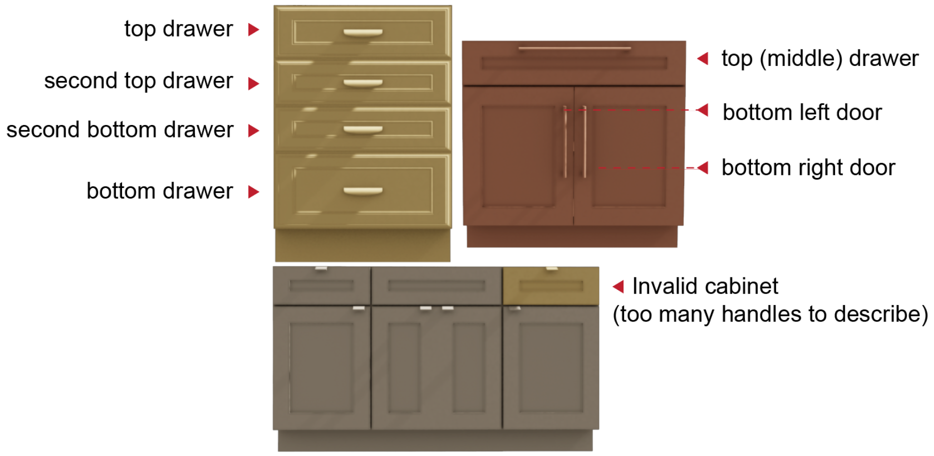


Figure 3.3: Language description examples. On the top, language descriptions are validly generated. On the bottom, Algorithm 1 fails because of too many doors and drawers to describe.

Task Statistics. The original dataset contains 346 pieces of the storage furniture. After filtering out low-quality meshes, URDF format (as as described in [Arr17]) errors, and invalid

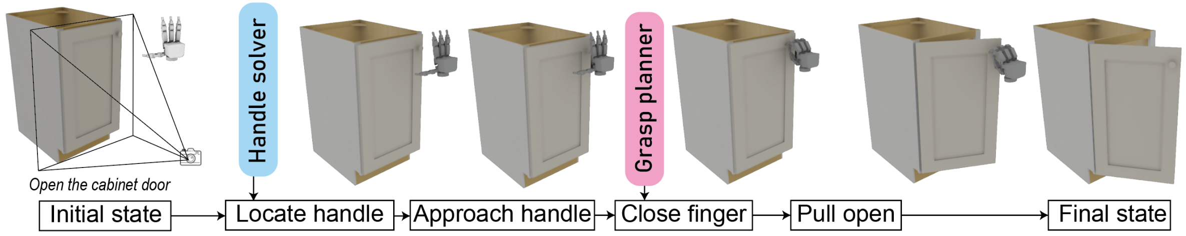


Figure 3.4: Multi-step planner to open a cabinet. We apply such a planner to all hands. All the hands share the same handle solver to locate the handle position, while they differ in the grasp planner.

cabinets from Algorithm 1, we were left with 174 unique cabinets, comprising a total of 372 doors and drawers, for training and testing with accompanying descriptions.

The train-test split is shown in table below.

	Count	Train	Test
Cabinet drawer	167	138	29
Cabinet door	205	145	60
Cabinet	174	135	39

Table 3.3: Data split for training and testing in *OpenD*

3.5 Model

Humans use visual perception to locate cabinets and understand verbal commands to determine which doors and drawers to open. Then, we open the desired door or drawer by controlling the movement of our hands and finger joints.

To address the challenges introduced by OpenD, we apply a multi-step planner to divide the overall task into several parts. As Figure 3.4 illustrates, the **handle solver** handles the challenges related to image perception and language understanding, while the **grasp planner** precisely controls the movement of the finger joints.

3.5.1 The multi-step planner

Initial state. The planner receives the inputs as RGB+D sensing *I* of the cabinet and language *S*.

Locate handle. To accurately recognize the grasp position, a handle solver identifies and localizes the correct handle on the door or drawer from the image and language input. We characterize a handle by its bounding box $bbox = (y_0, y_1, z_1, z_1)$ and determine the its center c and posture r:

$$c = ((y_0 + y_1)/2, (z_0 + z_1)/2)$$
$$r = vertical \text{ if } (z_1 - z_1) > (y_1 - y_0) \text{ else } horizontal$$

Approach handle. Combining depth sensing on the x-axis and the bounding box, we recover the handle's world position relative to the camera. Then, we drive the hand to the front of the handle, preparing to grasp and pull.

Close finger. A grasp planner defines the movement of the finger joints $\{d_i\}$ to grasp to handle.

Pull open. We drive the robot hand forward on the x-axis to pull open the target.

Final state. A task checker built in the OPEND determines the task's success based on the cabinet drawer's translation δ or cabinet door's rotation θ.

We define success conditions as follows: (1) the hand correctly opens the drawer or door described by the instruction, and (2) either the door open ratio $\theta/180$ or the drawer open ratio $\delta/(\text{drawer length})$ is greater than 20%.

3.5.2 Handle solver

The handle solver helps locate the cabinet's correct handle to open, providing the RGB image I and sentence S. To solve this problem, we employ two powerful models that utilize deep neural networks.

The first model, Faster R-CNN [RHG15] (see Figure 3.5), uses a region proposal network to perform efficient object detection. Given an image input, the Faster R-CNN model predicts the bounding boxes of all handles in the image. We then apply Algorithm 1 to generate language instructions for the predicted handles. Finally, we match the generated language instructions with the sentence input S to obtain the bounding box of the predicted handle.

The second model, **CLIPort** [SMF22] (see Figure 3.6), combines learning generalizable semantic representations for vision and understanding necessary spatial information for finegrained manipulation. CLIPort has a language-conditioned learning module to learn broad semantic knowledge (what) and the spatial precision (where) to transport. Since we already know what to open (the handle), we only use half of CLIPort to learn where to grasp according to the predicted affordance map.

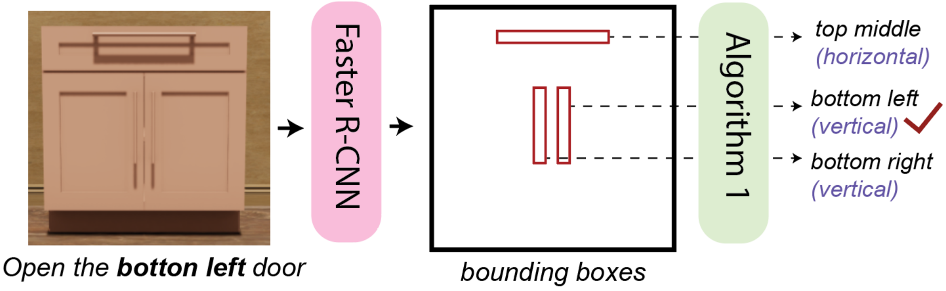


Figure 3.5: Training the handle solver with Faster-RCNN. This model uses a two-step strategy to merge image and language information.

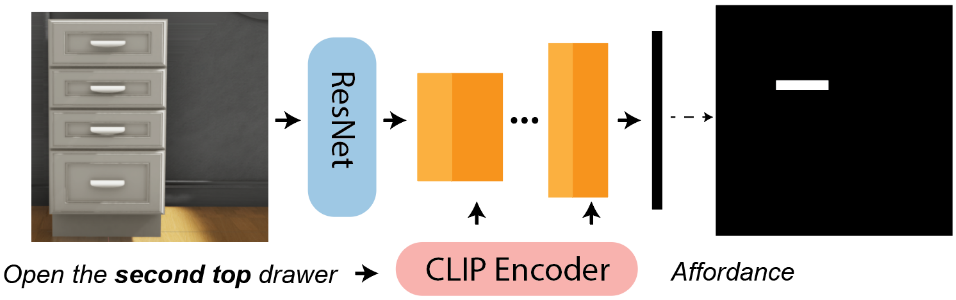


Figure 3.6: Training the handle solver with CLIPort. This model gets the affordance of the bounding box directly from the image and language inputs

3.5.3 Grasp planner

Once the grasp planner has determined the appropriate location for grasping within a defined local region (bounding box), it drives the hand to close the fingers. To assist with determining

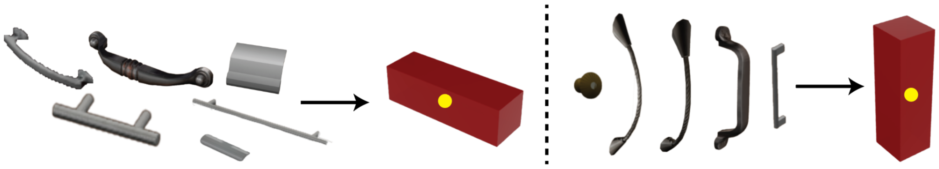


Figure 3.7: Handle simplification. Our model ignores the handle's shape and only considers the center position and posture. Left: horizontal handles. Right: vertical handles. The simplification is only for the grasp searching algorithm and does not influence the simulation.

the necessary finger motion and placement on the handle, we utilize a context-independent grasp planner [SS12].

We simplify the original implementation in two ways. First, we only consider the object's center c and posture r. Figure 3.7 shows some simplified examples of vertical and horizontal handles. Please note that this simplification only applies to the grasp search algorithm, as detailed collision and friction are still fully enabled in simulation. Second, we detach the hand from the robot arm, and therefore, we do not consider the configuration that the robot should be able to access.

The grasp planner assists in determining the ultimate joint state of the hand. As a result, we interpolate the initial and final joint positions to calculate the intermediate joint state.

3.6 Experiment

We perform experiments in OMNIVERSE to answer the following questions: 1) How to train the handler solver using the architecture of Faster R-CNN or CLIPort? 2) How does the performance of the grasp planner vary across different hands? 3) How generalizable is our multi-step planner when we combine the handler solver and the grasp planner?

Training the Faster R-CNN and CLIPort. To prepare the training data, we first render images of the cabinets from the camera and retrieve bounding boxes for the handles from the engine. We introduce randomization to the training data to encourage the development of more generalizable models. First, we randomly shift the camera position up to 10 centimeters on each axis. Then, we give the handle a random color with a 50% probability. The data augmentation procedure that combines this randomization results in 1740 training images, each with a resolution of 256 x 256 pixels. The handle solver learns from the images and the bounding boxes of the handles projected onto the images. We allocate 80% of the images for training and 20% for validation.

The Faster R-CNN model first loads a pre-trained backbone (ResNet50, MobileNetV3, or Low-resolution MobileNet) on the COCO (train2017) dataset [LMB14]. We fine-tune the model for 20 epochs on our training dataset, based on the regression loss (mean square error) of bounding box predictions. The Faster R-CNN models with the ResNet50, MobileNetV3, and

MobileNet backends report training losses of 0.110, 0.201, and 0.207, and validation losses of 0.158, 0.157, and 0.153. Although ResNet50 does not provide the best validation performance, we use it as the handle solver backend. We have empirically found that the ResNet backend works well with Algorithm 1 to predict bounding boxes, especially when we set a threshold of 0.8 for each bounding box detection score.

To train a CLIPort model, we start by encoding the language instructions with the CLIP text encoder [RKH21] and the images with ResNet-18. Our prediction target is the affordance map highlighted by the bounding box. To fine-tune the vision-language fusion layers of CLIPort, we use binary cross-entropy loss on our training data. We compared CLIPort models that have different numbers of vision-language fusion layers. The models with four, three, and two layers performed similarly. After fine-tuning, they reported binary cross-entropy losses averaged across each pixel of 0.134, 0.135, and 0.132 during training, and 0.146, 0.148, and 0.152 during validation. We selected the 4-layer model as the second option for the handler solver.

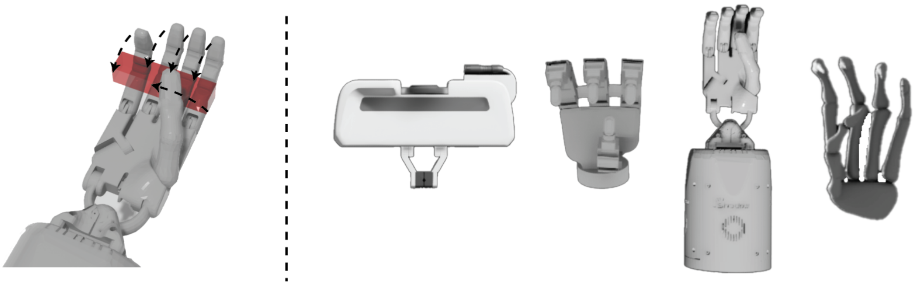


Figure 3.8: Grasp illustration. Left: grasp planner illustration. We regard the handle as a cuboid and the grasp planner searches the joint positions through curling fingers. Right: final grasping states deduced by the grasp planner.

Searching the grasp planner.

The grasp planner defines the movement required to close fingers for different types of hands. In our baseline approach, we designed one grasp planner per hand. To search for grasps, we used handles from the first three cabinets in the training dataset as samples. For the Franka gripper, the grasp planner works by simply closing the two fingers. For the other three hands, the grasp planner searches for the optimal curling of each finger until the hand can successfully grasp the sample handles.

Figure 3.8 depicts the illustration for grasp searching and the final grasp states. Table 3.4 reports the corresponding grasping success rate for each hand type.

Table 3.4: Grasping success rate for different hands

	Drawer	Door	Overall
Franka gripper	91.6%	58.5%	73.4%
Allegro hand	79.6%	66.2%	72.8%
Shadow hand	92.3%	24.4%	54.8%
Skeleton hand	50.3%	51.2%	50.8%

	Faster R-CNN						**CLIPort**					
	Drawer		**Door**		**Overall**		**Drawer**		**Door**		**Overall**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Franka gripper	19.6%	20.7%	23.4%	13.3%	21.6%	15.7%	43.5%	35.9%	19.3%	13.3%	31.1%	21.3%
Allegro hand	36.2%	48.3%	33.8%	21.7%	35.0%	30.3%	59.7%	50.0%	30.8%	20.0%	45.0%	30.0%
Shadow hand	47.1%	44.8%	18.6%	8.3%	24.7%	20.2%	17.3%	27.6%	30.8%	14.8%	17.6%	14.6%
Skeleton hand	15.9%	10.3%	24.8%	16.7%	20.5%	14.6%	21.0%	17.2%	26.9%	18.3%	24.0%	18.0%

Table 3.5: Task success rate for the multi-step planner.

If the hand is placed correctly at the grasping position, the Franka gripper and Allegro hand can successfully open over 70% of the targets, while the Shadow hand and Skeleton hand can only open around 50% of them. As the hand structure becomes more complex, the overall success rate drops, indicating that more precise control is required to handle interactions with handles (such as those of the Shadow hand and Skeleton hand). The experiment also shows that opening a cabinet door is more challenging than pulling open a drawer, as the hand may detach unexpectedly from the handle during door rotation, resulting in failure.

Overall results. Finally, we combine the well-trained handle solver and grasp planner to evaluate our baseline model on the test set. As shown in Table 3.5, we obtain two sets of experimental results using two different handle solvers while keeping the same grasp planner for each hand. Figure 3.9 plots one successful case for each hand during testing.

Comparing the experimental results, we found that the Allegro hand, which retains a hand shape while keeping a relatively simple structure, achieves the best testing performance (about 30% success rate). The results also confirm that our model has a harder time opening doors than opening drawers.

We also observed that the Franka gripper requires more precise identification of the grasp position due to its smaller fingers. It can successfully grasp 73.4% of the handles with ground-truth handle positions (see Table 3.4), but the performance drops drastically to 15.7% (Faster

R-CNN) and 21.3% (CLIport) when recognition is involved.

From our model, the performance of the other two hands (14.6% to 20.2% success rate) is not as good as that of the more straightforward Allegro hand (30.0%). Only knowing the position and orientation of the hand seems insufficient for accurate control.

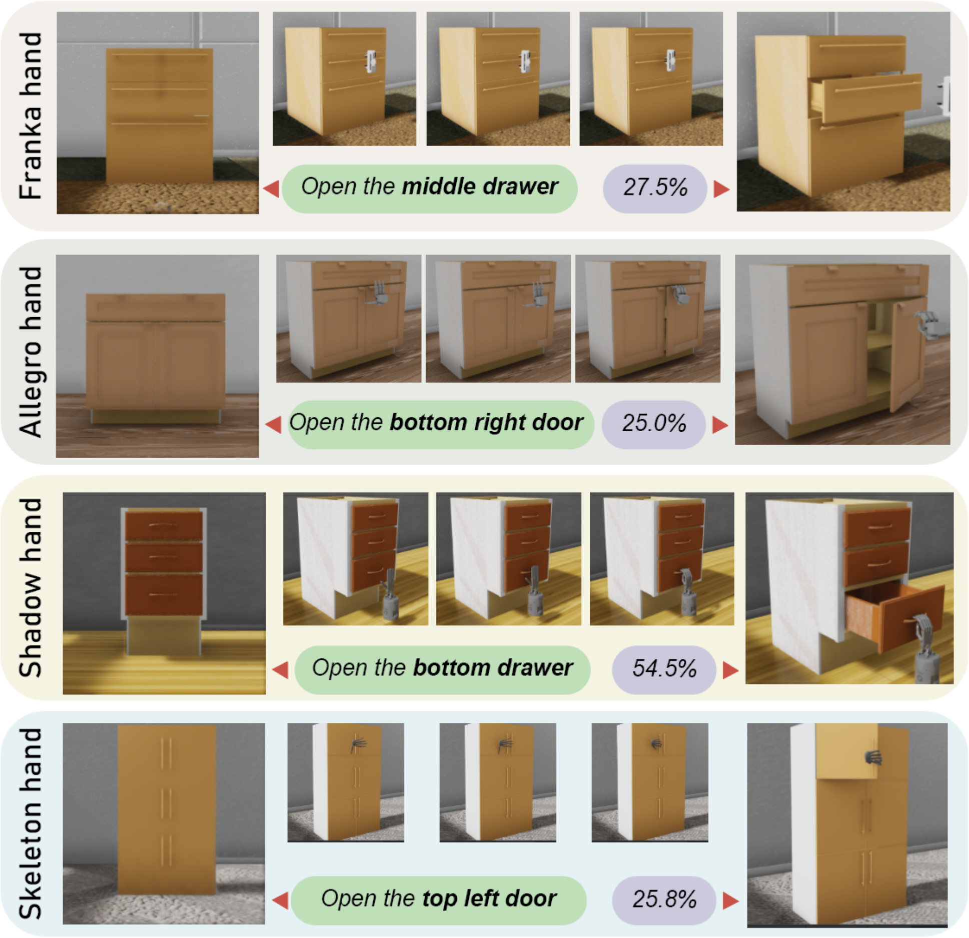

Franka hand
Open the **middle drawer**
27.5%
Allegro hand
Open the **bottom right door**
25.0%
Shadow hand
Open the **bottom drawer**
54.5%
Skeleton hand
Open the **top left door**
25.8%

Figure 3.9: Task solving examples in the testing phase. The green bubble highlights the language instruction, and the purple bubble shows the final open ratio. The leftmost image in each row, together with the instruction, is used as input for the handle solver.

3.7 Discussion

Human performance. We evaluated 10 randomly sampled cabinet doors and drawer directives from the dataset with 5 lab experts who completed 100 trajectories each for opening the drawer and door. Participants used a gamepad controller and achieved a 92% success rate for the drawer and 100% success rate for the cabinet. Failures were mainly due to impatience when grabbing the handle, indicating that tasks in OPEND are well-illustrated but leave room for model improvement.

Robot configuration. The placement of a robot is often the most important factor in determining whether it can successfully complete a task, particularly for robots with fixed bases. In the OPEND system, we begin by determining the hand position and rotation, which allows us to derive the possible robot configurations using the hand as the end effector. With the state of the end effector known, we can determine the entire robot's status via inverse kinematics [NSA22], sequential manipulation planning [Sti10], or deep learning [TE21].

Other approaches. To address the challenge, we also have tried two other approaches: transformer-based behavior cloning [MLX21] and offline reinforcement learning [LKT20]. However, the hundreds of demonstrations we collected for the human study are far from enough to train a transformer model. For OPEND, we set the physics update frequency to be 60 Hz. Planning and behavior cloning last at least 5 seconds on average to open the cabinet. Therefore, the algorithm for long-horizon planning and high-dimensional control for hands does not converge during reinforcement learning.

Chapter 4

Learning from the Human Values to Solve Mini Visual Task

Current pre-training methods in computer vision focus on natural images in the daily-life context. However, abstract diagrams such as icons and symbols are common and important in the real world. This work is inspired by Tangram, a game that requires replicating an abstract pattern from seven dissected shapes. By recording human experience in solving tangram puzzles, we present the Tangram dataset and show that a pre-trained neural model on the Tangram helps solve some mini visual tasks based on low-resolution vision. Extensive experiments demonstrate that our proposed method generates intelligent solutions for aesthetic tasks such as folding clothes and evaluating room layouts. The pre-trained feature extractor can facilitate the convergence of few-shot learning tasks on human handwriting and improve the accuracy in identifying icons by their contours.

4.1 Introduction

As many vision tasks are relevant, one would expect a model, particularly pre-trained from one dataset, to assist a different challenge. Traditionally, supervised pre-training on image classification has been employed to help object detection [SSS19] and semantic parsing [OKB19]. Moreover, popular unsupervised pre-training has recently produced remarkable results in visual tasks such as image classification [CRC20] and clustering [CGP20]. The common datasets to train basic models include PASCAL VOC [EVW10], ImageNet [DDS09], and COCO [LMB14], all of which contain photographs.

It is natural to start the pre-training process from real-life images to solve daily vision tasks. However, one of the underlying limitations of current works is their focus on content from natural images. Besides natural images, abstract diagrams, such as texts, symbols, and signs, also carry rich visual semantics and account for a large part of the visual world. For instance, it is shown that emojis can express rich human sentiments [FMS17], and diagrams like icons can map the physical worlds into symbolic and aesthetic representations [LGG19, MBT18, KBY20]. Furthermore, most of the tasks related to natural images can be accomplished by low-resolution vision [LN12] (see Figure 4.2). Therefore, training an enormous backbone (e.g., a deep residual network [HZR16]) to solve tasks related to abstract diagrams complicates the problem.

In this paper, we argue that we can solve the tasks related to abstract diagrams by learning from the process of replicating a tangram puzzle. The tangram, a dissection puzzle consisting of seven planar polygons (tans), is world-famous and has been used for many purposes, including art, design, and education. Although it only consists of seven tans, it can generate thousands of meaningful patterns such as animals, buildings, letters, and numbers. Solving a tangram puzzle associates with our cognitive and imaginative abilities.

We introduce the **Tangram**, a new dataset consisting of more than $10,000$ snapshots recording the steps to solve a total number of 388 tangram puzzles. A neural model can be pre-trained from the Tangram to solve two groups of downstream tasks.

The first group is about aesthetics. We introduce two toy tasks: folding clothes and organizing furniture (room layouts). Tuning the pre-trained network from several expert samples can generate an aesthetic landscape that helps make aesthetic judgments. Experiments show that our method performs best when cooperating with max-entropy inverse reinforcement learning [ZMB08] and generative adversarial imitation learning [HE16].

The second group includes several recognition tasks. In the N-way-K-shot setting, we show that conducting pre-training on the Tangram improves the performance of recognizing the human handwriting, including Omniglot [LST19] and Multi-digit MNIST [SCT20]. This method also improves the performance of icon recognition from contours.

This paper makes three major contributions:

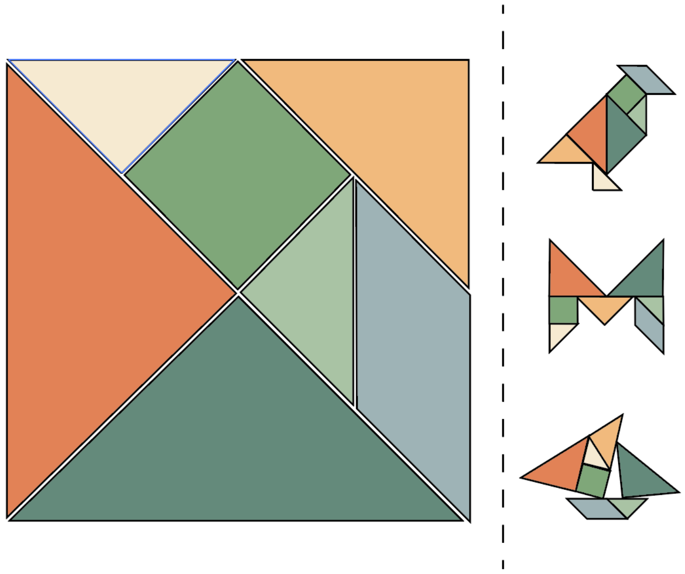


Figure 4.1: Square representation of the Tangram that consists of five triangles of three sizes, one parallelogram and one square. Tangram puzzles: a bird, the letter M and a sailboat.

- To our best knowledge, by introducing Tangram, we are the pioneers to suggest applying transfer learning from the human gaming experience to solve vision tasks.
- We demonstrate that pre-training from the Tangram can help solve both low-level aesthetics tasks and recognition tasks.
- We show that pretraining on the Tangram facilitates convergence in few-shot learning tasks, and improves the performance of recognition under low-level vision.

4.2 Related Work

An abundance of related work inspires our work, including pre-training in computer vision, rating image aesthetics with deep learning, and few-shot learning.

4.2.1 Pre-training

Pre-training methods can be either supervised or unsupervised. The supervised pre-training on ImageNet is conventional for object recognition, localization, and segmentation [HGD19]. Inspired by the success of unsupervised pre-training in natural language processing, the community has gained much interest in studying unsupervised pre-training in computer vision, such

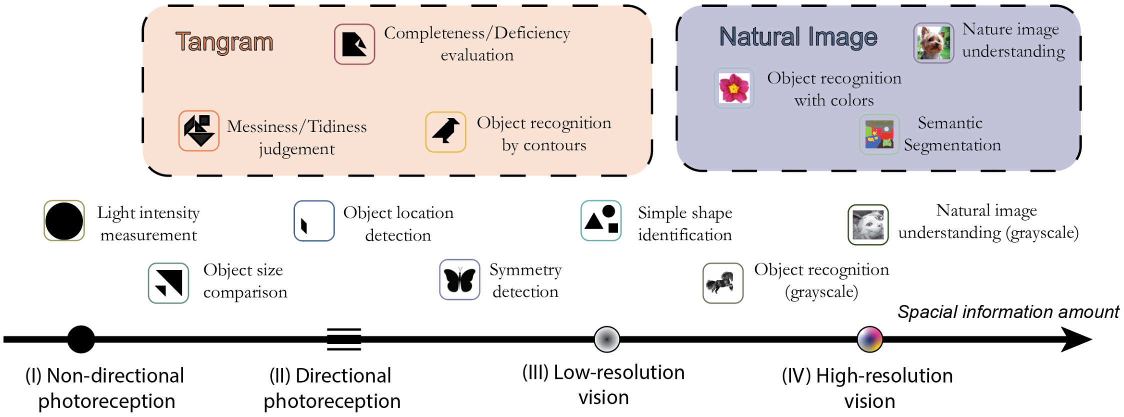

Tangram
Completeness/Deficiency evaluation
Messiness/Tidiness judgement
Object recognition by contours
Natural Image
Nature image understanding
Object recognition with colors
Semantic Segmentation
Light intensity measurement
Object location detection
Simple shape identification
Natural image understanding (grayscale)
Object size comparison
Symmetry detection
Object recognition (grayscale)
Spacial information amount
(I) Non-directional photoreception
(II) Directional photoreception
(III) Low-resolution vision
(IV) High-resolution vision

Figure 4.2: Visual perception tasks ranked by the amount of spatial information. In biology, visual perception tasks are divided into four levels based on the number of photoreceptors [LN12]. Our Tangram dataset relates to many low-resolution visual tasks, while current works usually focus on high-resolution natural images.

as contrastive training [CKN20], self-supervised training [JT20]. In many tasks, fine-tuning from a pre-trained model is faster than training from scratch. Pre-training can also help when high-quality labeled data is scarce.

4.2.2 Image Aesthetics

Image aesthetics assessment attempts to quantify an image's beauty. Image quality is influenced by numerous factors such as color [NOS11], lighting [Fre07], texture [KTJ06], and image composition [DLT17]. While subjective judgment by human eyes is the most reliable way to evaluate image quality, the beauty of an image can also be assessed by well-established photographic theories [ZM20]. Recent research has shown that data-driven approaches can be more efficient, especially those that employ feature extraction by multi-column convolutional neural networks (CNNs) [LLJ15, DSM19]. Popular databases for image quality assessment (IQA) are mainly collected as photos (natural images), such as the Photo.Net database [JDF11] and the CUHK-PhotoQuality database [LWT11]. Some emerging databases consist of images from virtual contents such as screen content image quality database (SCIQ) [NMZ17] and compressed Virtual reality image quality database (CVIQ) [SMZ19].

4.2.3 Few-shot Learning

The main goal of few-shot learning is to learn new tasks with a few support examples while maintaining the ability to generalize. Recently, there has been a growing interest in achieving the goal by learning prior knowledge from previous tasks, especially training feature extractors that can efficiently segregate novel classes [HGP20].

We apply our Tangram dataset to train the feature-extracting parts of optimization-based meta-learning algorithms such as MAML [FAL17] and ANIL [RRB19]. Besides, since the Tangram only contains shapes and contours, we perform experiments on the few-shot learning tasks that are color-free and texture-free, for example, the Omniglot challenge [LST19].

4.3 Pre-training from the Tangram

4.3.1 Data Collection

To collect the process of solving puzzles from human experience, an interactive labeling tool is developed using the Unity game engine [Haa14]. The labeling tool can record every step of moving, rotating, or flipping of one tan as a snapshot. Seven lab technicians spent weeks on completing a total number of 776 solutions to 388 unique puzzles, capturing more than $10,000$ snapshots.

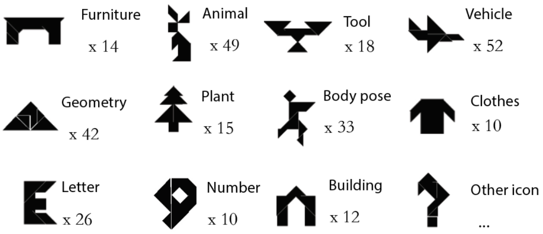


Figure 4.3: Collected examples of different categories in the Tangram dataset.

Figure 4.3 illustrates an overview of the puzzles types and their counts. The Tangram dataset consists of diverse tangram patterns including animals, plants, letters, numbers, buildings, human poses, and some everyday objects. It requires necessary perceptive recognition and elementary geometry skills to solve them. We will release the dataset to the public to encourage further study into abstract image understanding.

4.3.2 Learning from Puzzles

Denote the order set $(I_1, I_2, ..., I_{n_p})$ as the process to solve a tangram puzzle P, where each $I_i, i \in \{1, ..., n_p\}$ is an image representing one step toward the solution, and n_p is the total number of steps. Since a tangram pattern only has shapes and contours, I_i is a binary image with size $H \times W$.

What can we learn from the puzzles, and how can we use the solving steps? We argue that the Tangram reveals two pieces of information:

- The step-by-step solving process leads to more complete and tidy shape combinations, containing the perception of beauty.
- There is a connection between the pattern and the name of the object due to correspondence between the final completed pattern and a real-world object.

Therefore, we formulate two learning goals and assign two loss functions.

Let $f_\theta : \{0,1\}^{H\times W} \mapsto [0,1]$ be the function indicating the degree of completeness of step I_i. We define the **completeness contrast loss** (CCL) for the process $(I_i)_{i=1}^{n_p}$ as

$$\begin{aligned} \text{CCL}(I_1, ..., I_p) = &(0 - f_\theta(I_1))^2 \\ &+ \sum_{t=1}^{n_p-1} \Big(f_\theta(I_t) - f_\theta(I_{t+1})\Big)^2 \\ &+ (f_\theta(I_{n_p}) - 1)^2. \end{aligned} \tag{4.1}$$

By the Cauchy–Schwarz inequality, CCL reaches minimum value $\frac{1}{n_p+1}$ when $f_\theta(I_i) = i/(n_p + 1), i = 1, 2, ..., n_p$. Minimizing CCL results in a right order for $(I_i)_{i=1}^{n_p}$.

Let $g_\phi : \{0,1\}^{H\times W} \mapsto \mathbb{R}^{N_{\text{word}}}$ map the binary image to the word embedding W_P of a pattern P, where N_{word} is the dimension of the embedding space. The **puzzle meaning loss** (PML) for the final step I_{n_p} is defined as

$$\text{PML}(I_{n_p}) = |g_\phi(I_{n_p}) - W_P|^2. \tag{4.2}$$

Figure 4.4 depicts an implementation of the two loss functions described above. Panel

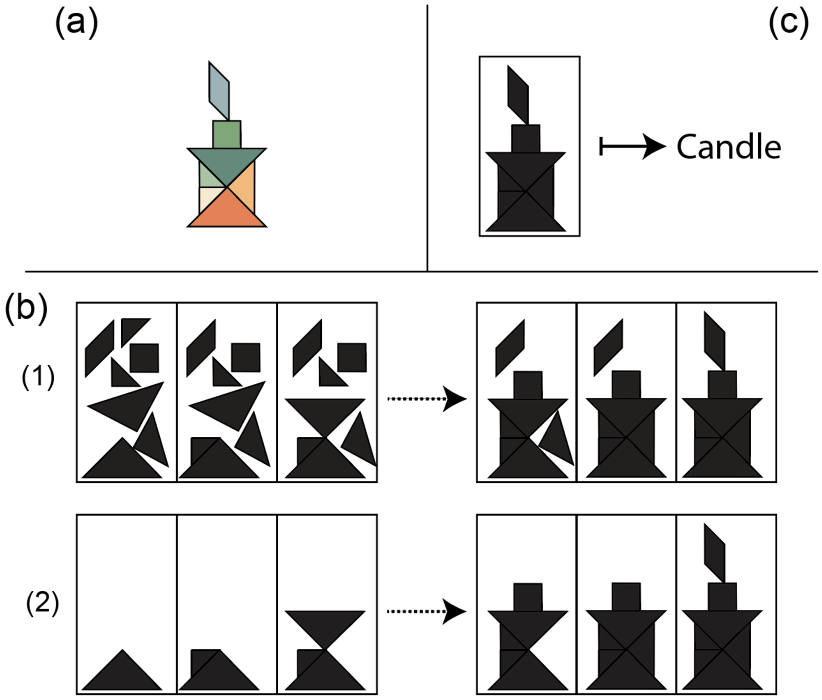


Figure 4.4: (a) The expected solution of a tangram puzzle. (b) The process of solving the puzzle with its two variants. (c) The final completed puzzle image and the meaning of the item.

(b) demonstrates two variants of the puzzle-solving processes. The first variant traces all tans, recording progression from disorganization to neatness; the second variant traces only the final state of moved tans and represents a progression from fragmentation to completeness.

To train the functions f_θ and g_ϕ, we use a simple convolutional neural network with only four 3×3 convolutional layers. Each image is resized into 28×28. We apply the 50-dimension GloVe embedding [PSM14b] for pattern names, and we assign 80% of the weight on CCL and 20% on PML. The feature extraction part of the network is transferred to achieve other challenges.

4.4 Experiments

We define **mini visual tasks** as the vision tasks that only require learning from low-resolution binary images. We divide mini visual tasks into two categories: aesthetic tasks and recognition tasks. We choose folding clothes and generating room layouts (organizing furniture) as representatives for the first category, identifying human hand-writings and recognizing icons for the second.

4.4.1 Folding Clothes

Folding clothes is a classic task in robotics that has received heated discussion among various works. Prevalent methods include grounding human demonstration from videos [YLF15], employing random decision forests and probabilistic planning [DKK14], using deep reinforcement learning [JAT20], and designing a modifiable stochastic grammar [XSX16].

We abstract the clothes-folding challenge as a purely visual task: the contour of the dress/suit/shirt/pants is represented by a binary image, and folding clothes is characterised by manipulating images. Figure 4.5 shows an image-like abstraction of folding a dress.

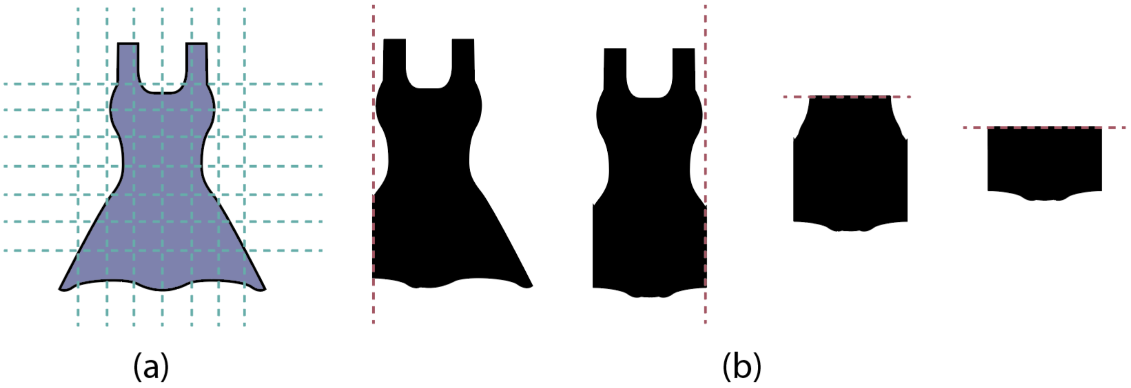


Figure 4.5: A dress with folding axes and folding steps.

The current state of the clothes s is represented by a binary image I from image space $\mathscr{S} = \{0,1\}^{H\times W}$, and an action a leads to fold the image along a certain axis (see figure 4.5). We also regard this task as a few-shot learning problem: as we are only given a few expert trajectories $\pi_E = \{\tau_{E_1}, \tau_{E_2}, ..., \tau_{E_{ne}}\}$, where each trajectory τ_{E_i} is represented by the order sequence of states $(s_{E_{i1}}, s_{E_{i2}}, ...)$ towards the solution, the problem is how we can fold other arbitrary clothes we have not seen before.

We try several different ways to solve this task, including directly minimizing the CCL for expert trajectories and drawing on the popular algorithms from inverse reinforcement learning (IRL). The algorithms listed below can be applied not only to perform clothes-folding and furniture-organizing, but to solve a wide range of challenges related to robotics.

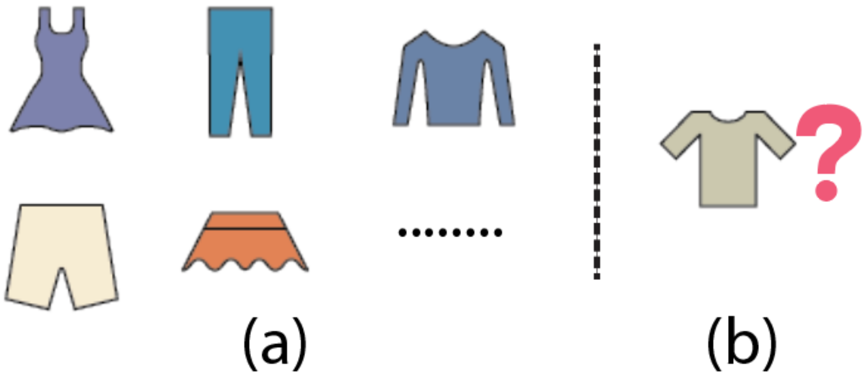


Figure 4.6: Expert sample clothes and A T-shirt unseen before.

- **Score learning** (SL): we can direclty give a score to a state $V_\delta : \mathscr{S} \mapsto [0,1]$, by learning from expert trajectories with the CCL (see equation 4.1):

$$V_\delta(s) := f_\theta(s). \tag{4.3}$$

- **Max-entropy inverse reinforcement learning** (ME-IRL) [ZMB08]: suppose a trajectory $\tau_i = (s_1, s_2, \ldots)$ is sampled from the current cloth-folding policy π_i, and $F_\psi : \mathscr{S} \mapsto [0,1]$, is the evaluation function for state s, we can calculate the gradient of ψ by

$$\frac{\partial \mathscr{L}_\psi}{\partial \psi} = \mathbb{E}_{s \sim \tau_E}\left[\frac{\partial F_\psi(s)}{\partial \psi}\right] - \mathbb{E}_{s \sim \tau_i}\left[\frac{\partial F_\psi(s)}{\partial \psi}\right], \tag{4.4}$$

where $\mathscr{L}_\psi = P(\tau | \pi_i, \tau \in \pi_E)$ is the likelihood function of taking expert trajectories under the current policy.

- **Generative adversarial imitation learning** (GAIL) [HE16]: after initializing the discriminator function $D_\omega : \mathscr{S} \mapsto [0,1]$ to distinguish states between expert and sampling trajectories, we can update ω with gradient

$$\begin{aligned}\frac{\partial \mathscr{L}_\omega}{\partial \omega} = \mathbb{E}_{s \sim \tau_E}&\left[\frac{\partial \log D_\omega(s)}{\partial \omega}\right] \\ &+ \mathbb{E}_{s \sim \tau_i}\left[\frac{\partial \log(1 - D_\omega(s))}{\partial \omega}\right]\end{aligned} \tag{4.5}$$

where $\mathscr{L}_\omega$ is the adversarial loss [HE16] and τ_i shares the same meaning as above. Notice that we make a modification to GAIL by only distinguishing the state s instead of the state-action pair (s,a) since we are not given enough state-action pairs under few-shot settings.

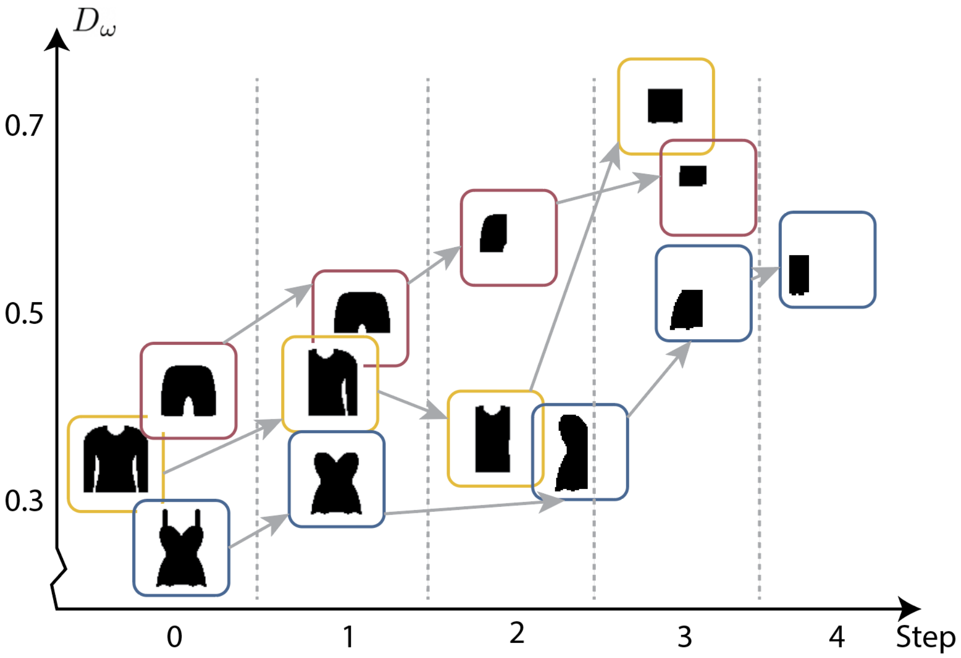


Figure 4.7: Aesthetic scores induced by D_w (pre-trained).

For simplicity, we regard the greedy policy deduced by the value of V_δ, F_ϕ and D_ω as the propagated policy π_i for SL, ME-IRL and GAIL. We assume that the clothes are put straight initially and they can only be folded along vertical and horizontal axes. The size of the image I representing the state s is 28×28 and there are ten vertical and ten horizontal folding axes evenly distributed in the image.

We apply the network of the same structure in Section 4.3 *Pre-training from the Tangram* for feature extraction to calculate V_δ, F_ψ and D_ω. Three different ways along with pre-training or non-pre-training cases provide us with six different models. The models are trained on the expert trajectories from a total number of 18 clothes, including dresses, long shirts, T-shirts, trousers, short pants, and skirts (three for each type). Then, models are tested on six new clothes from the aforementioned types and three clothes from other types.

We refer to V_δ, F_ϕ and D_ω derived from equations 4.3, 4.4, and 4.5 as the *aesthetic scores* of cloth-folding. Figure 4.7 illustrates that D_ω increases as the clothes-folding process goes along. We compare the performance between different models by calculating the ranking of the ordered states $(s_{E_{i1}}, s_{E_{i2}}, ...)$ of expert trajectories based on V_δ, F_ϕ and D_ω. Since on average the length of expert trajectories is around four, we only consider the precision at K (P@K) with $K \leq 3$. Recall at K as gives similar results.

Table 4.1 compares the overall difference in P@K between the pre-trained model and the non-pre-trained model (training from scratch) for the training expert samples (see the detailed comparison for each model in the Appendix). In general, we can see that pre-training improves the training precision and reduces the variance. We select the best models of the six methods and test them once on the nice clothes that are unseen before. Table 4.2 shows the mean and standard deviation of testing P@K. Except that ME-IRL without pre-training outperforms the pre-trained one w.r.t. P@1, pre-training improves the overall test accuracy, and the high precision on each value ($K = 1,2,3$) implicates overall better aesthetic scores.

ME-IRL and GAIL are common data-driven algorithms in the IRL domain. As with SL, their performance is heavily dependent on the amount of expert data given for training. Therefore, tuning from a pre-trained model can alleviate data reliance.

	P@1	**P@2**	**P@3**
From scratch	0.54 ± 0.5	0.66 ± 0.3	0.76 ± 0.2
Pre-training	0.77 ± 0.4	0.84 ± 0.3	0.86 ± 0.2

Table 4.1: The mean and standard deviation of training P@K: a comparison between models with or without pre-training.

	P@1	**P@2**	**P@3**
SL	0.22±0.46	0.44 ± 0.46	0.55 ± 0.47
+ Pre	**0.89 ± 0.33**	0.78 ± 0.26	0.81 ± 0.18
ME-IRL	**0.89 ± 0.33**	0.78 ± 0.26	0.74 ± 0.22
+ Pre	0.67 ± 0.50	**0.94 ± 0.17**	0.96 ± 0.11
GAIL	0.33 ± 0.25	0.61 ± 0.33	0.74 ± 0.22
+ Pre	**0.89 ± 0.33**	**0.94 ± 0.17**	**1.00 ± 0.00**

Table 4.2: The mean and standard deviation of testing P@K.

4.4.2 Evaluating Room Layouts

Generating room layouts is different from folding clothes in that the latter focuses on the shape change of a single object, while the former requires arranging multiple objects. These two pre-training exercises may correspond to the two variants of a replicating process of a tangram puzzle(see figure 4.4).

The study of the layout generation has been active in various domains such as architectural design [NCC20, BYM13] and game level design [MVL14, HMV13]. We focus on the task of generating content for indoor scenes, especially furniture arrangement [YYT11, RWL19, QZH18], and abstract it as a purely visual task as shown in Figure 4.8.

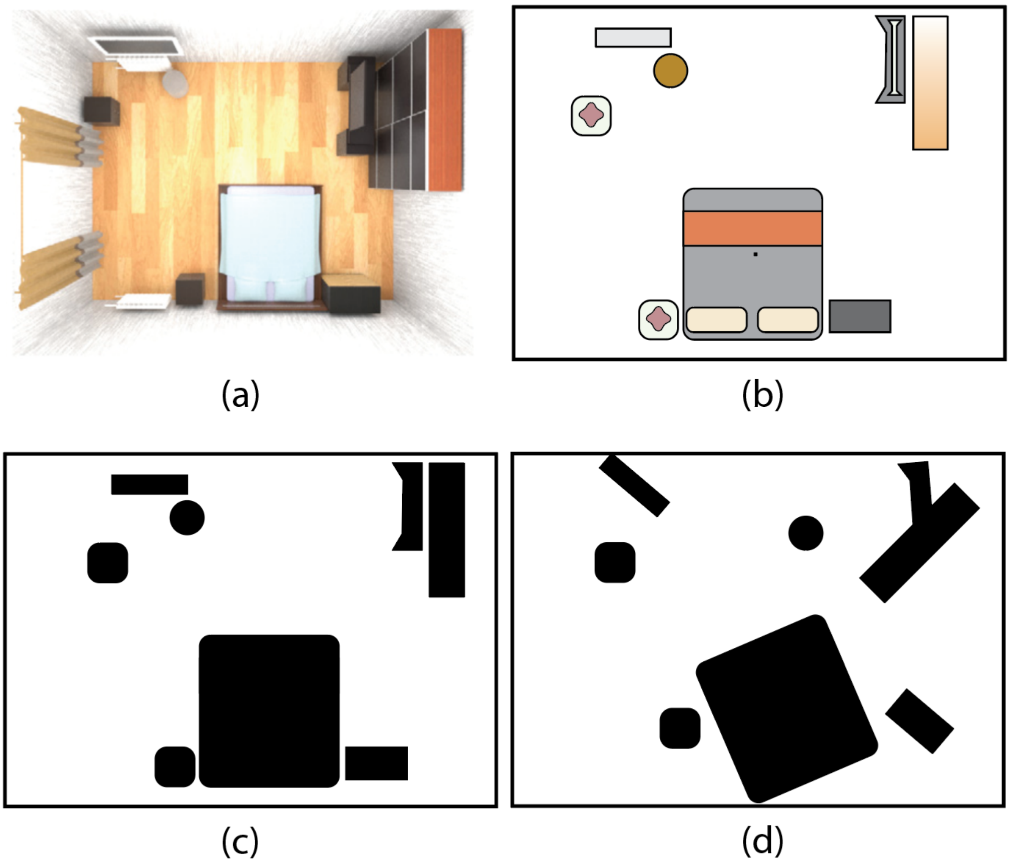


Figure 4.8: (a) Original indoor scene sample from [QZH18]. (b) Abstract room layout. (c) Binary image representation. (d) Room messed up.

We apply the state-of-the-art indoor scene synthesis using stochastic grammar [QZH18] to generate the ground truth. Step by step, we perturb the room layout by the action a that changes the position (10 pixels each step) and angle (15° each step) of the furniture, and the reversed steps generate an expert trajectory τ_{E_i} to tidy up the room.

	P@1	**P@2**	**P@3**
From scratch	0.18 ± 0.2	0.23 ± 0.3	0.32 ± 0.3
Pre-training	0.23 ± 0.4	0.28 ± 0.4	0.39 ± 0.4

Table 4.3: Training P @ K comparison between models with or without pre-training.

	Original	**Perturbed**
GAIL (from scratch)	0.25 ± 0.45	0.23 ± 0.38
GAIL (pre-trained)	0.31 ± 0.41	0.29 ± 0.35

Table 4.4: Testing accuracy (P@1) of ranking the best room layout.

As in the previous experiment, we use a binary image I to represent the current state s, and apply the three functions V_δ, F_ψ and D_ω to generate the aesthetic landscapes of the room. We only train our methods from 30 generated expected trajectories and test them on 10 groups of new room organizing trajectories.

Table 4.3 shows the overall training improvement by pre-training. As in the previous experiment, pre-training improves the training accuracy. We select the best model GAIL from training, and we test it on identifying the best room layout from the testing trajectories. We also perturb each room in the trajectory a little to test the robustness of the model. Table 4.4 compares GAIL with/without pre-training on the testing challenges. The results indicate that pre-training on the Tangram improves performance in choosing the best room layout.

4.4.3 Few-shot Learning

The goal of few-shot learning is to utilize new data having seen only a few samples. In this section, we focus on the N-way-K-shot classification: a typical problem to discriminate between N classes with only K samples from each to train from.

The method we propose follows the paradigm of meta-learning [SLC19]: we first train a feature extractor as a base-learner, which is later fine-tuned for another task through a meta-learner. As in previous experiments, a base learner is trained from the Tangram dataset, and then we perform a meta-test on the challenge of Omniglot [LST19] and Multi-digit MNIST [CCM18], where a binary image brings enough information to do classification.

We select three methods: MAML [FAL17], ANIL [RRB19] and Prototypical Networks [SSZ17] to train the meta-learner from our base-learner. MAML is a popular meta-learning algorithm for few-shot learning, achieving competitive performance on several benchmark few-shot learning problems. ANIL simplifies MAML by alleviating the inner training loop but keeping the training procedure for the task-specific part. Prototypical networks learn to map the prototypes to a metric space, and then distances between prototypes and encoded query inputs are used to make the classification. To test the base-learner (feature extractor) trained on our Tangram data, we compare it with base-learners trained from EMNIST [CAT17] and Fashion-MNIST [XRV17][1]. All base-learners share the same network structure.

	Omniglot	**Double-MNIST**
Random	$33.7\% \pm 2.0\%$	$7.3\% \pm 1.5\%$
EMNIST	$55.0\% \pm 5.4\%$	$26.8\% \pm 2.2\%$
Fashion-MNIST	$43.9\% \pm 4.1\%$	$30.1\% \pm 1.2\%$
Tangram	$\mathbf{56.0}\% \pm \mathbf{4.7}\%$	$\mathbf{36.0}\% \pm \mathbf{2.7}\%$

Table 4.5: Five-way-five-shot learning: the mean and the standard deviation of testing accuracy (logistic regression only).

	Omniglot	**Double-MNIST**
Random	$8.0\% \pm 0.7\%$	$6.1\% \pm 0.1\%$
EMNIST	$\mathbf{22.1}\% \pm \mathbf{1.2}\%$	$7.5\% \pm 0.1\%$
Fashion-MNIST	$15.6\% \pm 1.4\%$	$9.2\% \pm 0.5\%$
Tangram	$22.0\% \pm 1.0\%$	$\mathbf{10.5}\% \pm \mathbf{1.0}\%$

Table 4.6: Twenty-way-five-shot learning: the mean and the standard deviation of testing accuracy (logistic regression only).

Before moving on to fine-tuning, we compare the feature extractors obtained by training on the above datasets. We train only the last layer of the network as logistic regression. As can be seen from Table 4.5 and Table 4.6, feature extractors pre-trained on the Tangram, EMNIST, and Fashion-MNIST perform a lot better than the randomly initialized feature extractor. Except that the base-learner trained on EMNIST performs best in the 5-way-5-shot task on Omniglot, base-learners trained on the Tangram are powerful on other tasks, demonstrating their better adaptability.

[1] we did not train the base-learner on MNIST[Den12] because it is highly related to Multi-digit MNIST.

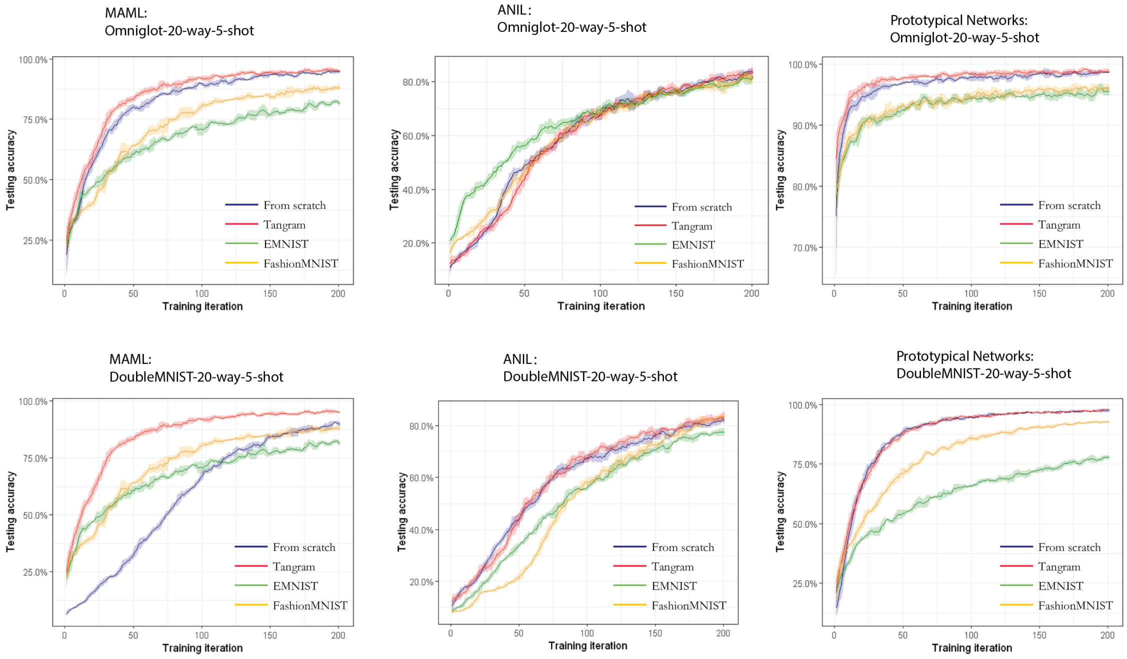


Figure 4.9: Testing accuracy of base-learners for different algorithms on different tasks.

Figure 4.9 compares the tuning process of different base-learners. Tuning the baser-learners pre-trained from the Tangram dataset guarantees the final performance compared with learning from scratch, while in some tasks it even speeds up convergence. However, for the other two feature extractors trained from EMNIST and FashionMNIST, although they may have a good start in some tasks, overall they tend to undermine the convergence speed and the final results, which reflects the difficulty of tuning a baser-learner for an irrelevant task. This result also demonstrates the importance of selecting a proper fundamental learning dataset in transfer learning.

Table 4.8 and Table 4.9 compare the final training results between training from scratch and pre-training from Tangram, where we apply ANIL as the tuning algorithm. The results shown are trained after 500 epochs. From the tables, we can see that pre-training from the Tangram provides slightly better results than training from scratch.

4.4.4 Icon Recognition

In this section, we study the recognition of abstract icons. While recognition tasks in natural pictures have been booming in the literature, visual abstraction receives comparably less attention.

	Flowers-17	
	ResNet-18	***EfficientNet-b0***
From Scratch	73.5% ± 3.4%	76.1% ± 1.4%
Tangram	**76.3% ± 3.8%**	76.0% ± 1.2%
	Flowers-102	
	ResNet-18	***EfficientNet-b0***
From Scratch	50.5% ± 1.3%	51.7% ± 1.5%
Tangram	**51.1% ± 0.8%**	50.6% ± 1.1%
	Icons-50	
	ResNet-18	***EfficientNet-b0***
From Scratch	51.7% ± 1.5%	86.5% ± 0.4%
Tangram	**87.1% ± 1.1%**	85.0% ± 1.0%

Table 4.7: Classification results between training from scratch and pre-training from the Tangram. The inputs are binary images representing the contours only.

	Omiglot	**Double MNIST**
From scratch	97.1% ± 1.4%	98.4% ± 1.3%
Tangram	98.1% ± 1.0%	98.5% ± 1.0%

Table 4.8: Five-way-five-shot testing accuracy after training by ANIL.

At first glance, icon recognition is a relatively straightforward task compared to the recognition task in natural images, since most icons are simple shapes that are not affected by light or blocking. However, it is worth considering how these abstract icons are formed, and how these seemingly simple icons can convey a variety of meanings. In this part, we wonder whether pre-training on the Tangram dataset assists in recognition of icons. Icons-50 [HD18] is a collection with 50 types of icons and thousands of training samples. We run the experiments with Icons-50 and test our methods on Flowers-17 and Flowers-102 [NZ08].

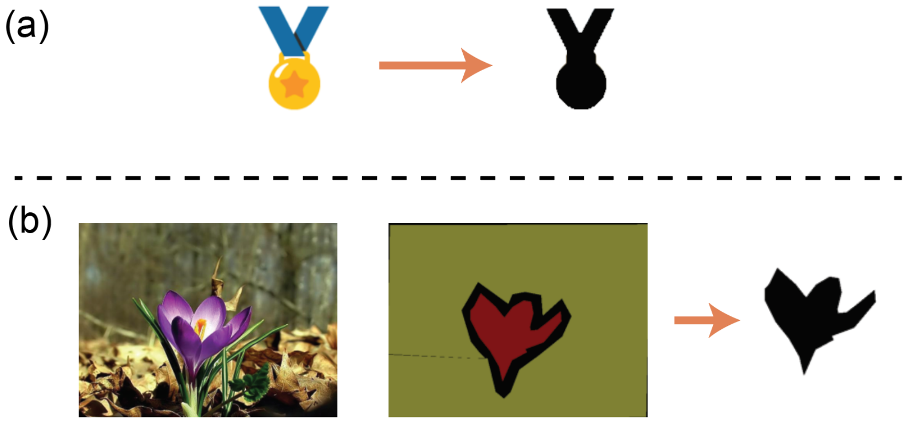


Figure 4.10: Data processing for (a) icons and (b) flowers.

	Omiglot	**Double MNIST**
From scratch	$92.4\% \pm 1.0\%$	$98.2\% \pm 0.3\%$
Tangram	$93.5\% \pm 0.9\%$	$98.2\% \pm 0.2\%$

Table 4.9: Twenty-way-five-shot testing accuracy after training by ANIL.

For Icons-50, we select icons with a white background coverage greater than 40% and draw their contours, which results in a total number of $2,450$ samples. Flowers-17 and Flower-102 are well labeled with flower contours. Flowers-17 contains 17 flower types and 849 samples, and Flowers-102 has 102 flower types and $8,189$ samples. For each dataset, 80% of the samples are used for training and the remaining 20% for testing. We use ResNet-18 [HZR16] and EfficientNet-b0 [TL19] as the network architectures for icon classification. The inputs of the network are binary images of the size 224×224. Table 4.7 compares the model trained from scratch and the model pre-trained from Tangram. Although training Efficient-n0 from scratch brings good performance, the pre-trained model with ResNet-18 shows overall better testing accuracy.

4.5 Discussion

In this part, we apply the pre-training on the Tangram beyond mini visual tasks to discuss its potential and limitations.

Despite the excellent performance of pre-training on the Tangram in the above experiments, this method does not work in some similar tasks such as the mini-ImageNet challenge [VBL16]. Each image in the dataset is rich in color, background, and contextual details. As a result, accurately determining the content of the images require multiple visual skills to perform semantic segmentation, noise removal, and recognition. Because replicating a tangram puzzles does not require the advanced visual skills above, pre-training on the Tangram could not contribute significantly to speeding up convergence or improving the final performance.

If we empirically rank visual tasks in terms of the amount of spatial information (see Figure 4.2), we find that supervised pre-training from our Tangram cannot help solve the tasks that involve higher resolution. We may draw an inference: transferring pre-trained feature extractors from a low-information task to a high-information task seems unhelpful, even harmful

	Epoch 100	**Epoch 500**
From sractch	**39.3%±4.3%**	**79.5%±1.6%**
EMNIST	33.4%±1.4%	67.1%±2.3%
FashionMNIST	32.9%±2.6%	40.2%±1.5%
Tangram	36.4%±4.4%	71.3%±2.0%

Table 4.10: Testing accuracy on mini-ImageNet: five-way-five-shot learning with ANIL.

to the convergence and testing results. From a biological perspective, some visual tasks are easy to complete given the least amount of information from the light, while others need more photoreceptors (cells that respond to light) to accomplish. The visual behaviours gradually require more sophisticated sensory organs and neural processing when tasks become more complicated [Nil13]. Therefore, to help accomplish more sophisticated visual tasks, images used for pre-training usually need to have more details, and the pre-training process tends to be more complex.

4.6 Appendix

Pre-training Network

Networks for pre-training, folding clothes, organizing furniture and Omniglot/Multi-digit MNIST challenge share the same structure as feature extractors. The following figure plots the detailed structure of the feature extractor. The feature extractor has about 112k trainable parameters.

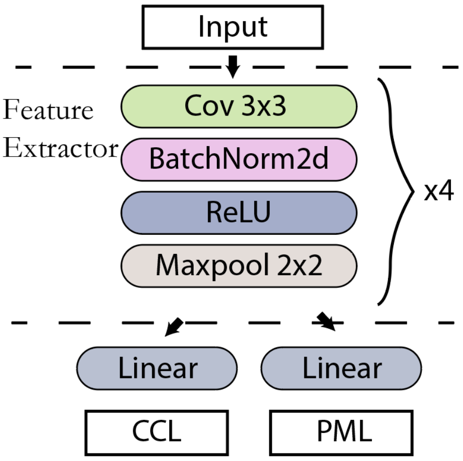
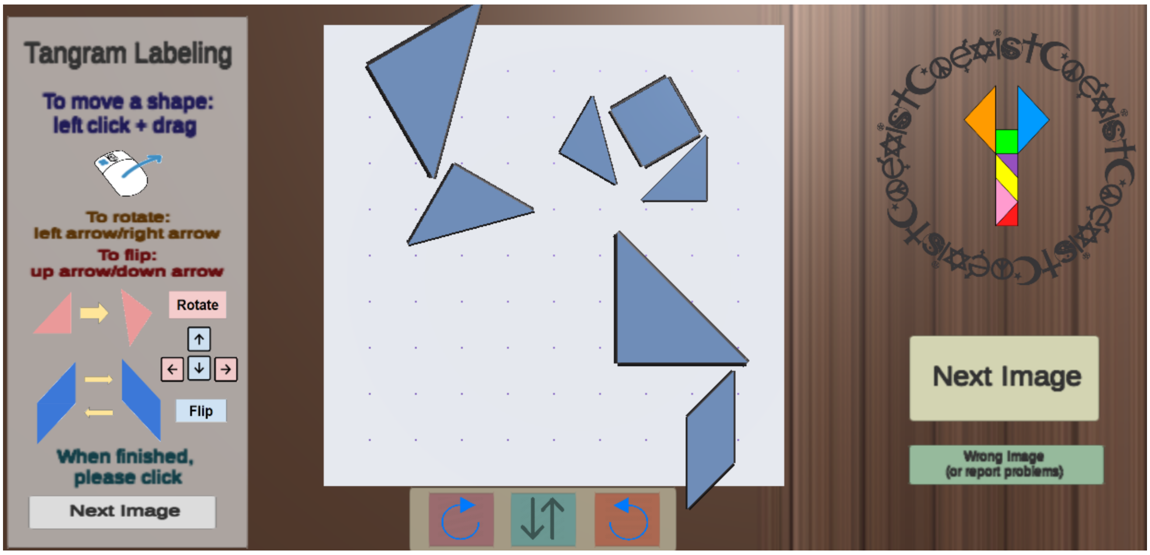


Figure 4.11: Network architecture for pre-training



Tangram Labeling

To move a shape:
left click + drag

To rotate:
left arrow/right arrow
To flip:
up arrow/down arrow

When finished,
please click

Figure 4.12: Tangram labeling tool.

The above image shows the tool for labeling tangram puzzles, where a board in the middle contains 100 × 100 grid points and seven tans. One tan can be moved on the grid point, be rotated every 15° and be flipped vertically. Every step of the process is recorded.

Clothes-Folding details

We use a total number of 18 clothes with their folding steps for training, including trousers, dresses, T-shirts, skirts, pants and jackets. The clothes were all drawn by hand on Adobe Illustrator.

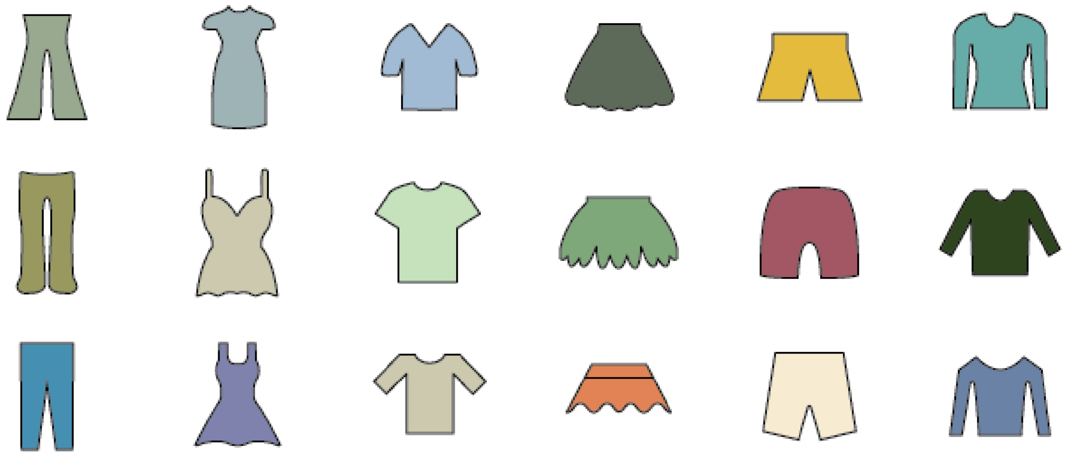


Figure 4.13: Training clothes

The following figure plots nine clothes used for testing, including three vests which do not belong to any type of training clothes.

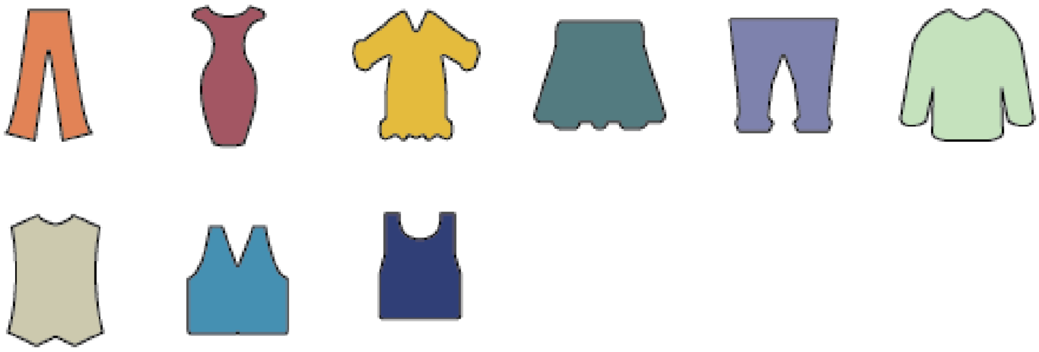


Figure 4.14: Testing clothes

The reason that we call V_δ, F_ϕ and D_ω aesthetic landscapes instead of value functions is they give a intuitive guidance for aesthetic tasks. A greedy policy based on them can generate results good enough. Moreover, because we have not clearly defined the rewards and they are not calculated by Bellman equation, to call them value functions seem inappropriate. An example (folding a pair of short pants) with the aesthetic landscape is illustrated in the following table.

Generally, GAIL with pre-training on our Tangram generates reasonable landscapes.

	Step 0	**Step 1**	**Step 2**	**Step 3**
SL	0.4643	0.4673	0.5434	0.4979
SL(Pre)	0.5015	0.4608	0.4967	0.5063
ME-IRL	0.0001	0.0004	0.9722	0.9999
ME-IRL(Pre)	0.0014	0.0081	0.0183	0.033
GAIL	0.4783	0.4039	0.4886	0.5071
GAIL(Pre)	0.4117	0.4938	0.6186	0.6727

Table 4.11: Aesthetic landscapes when folding a pair of short pants.

Organizing room layouts

We generated 40 room layouts including living room, bedrooms, kitchens and bathrooms from *Human-centric Indoor Scene Synthesis* [QZH18], by ruling out the following configurations: *door*, *rug*, *ottoman*, *cutting board*, *fence*, *clock*, *vase*, *television*, *partition*, *person*, *garage door*, *picture frame*, *toy*, and *shelving*. The following figure plots some samples.

We use 30 of them for training and 10 for testing.

We generate expert trajectories by randomly moving and rotating the furniture in the room. In each step, one desk/chair/TV stand/... can be moved 10 pixels randomly to left/right/up/down, or be rotated by 15° clockwise/counter-clockwise from its center. On average, there are 15 types of furniture in one room therefore the average length of expert trajectories is 15. We apply the same method as cloth-folding to solve aesthetic evaluation for room layouts.

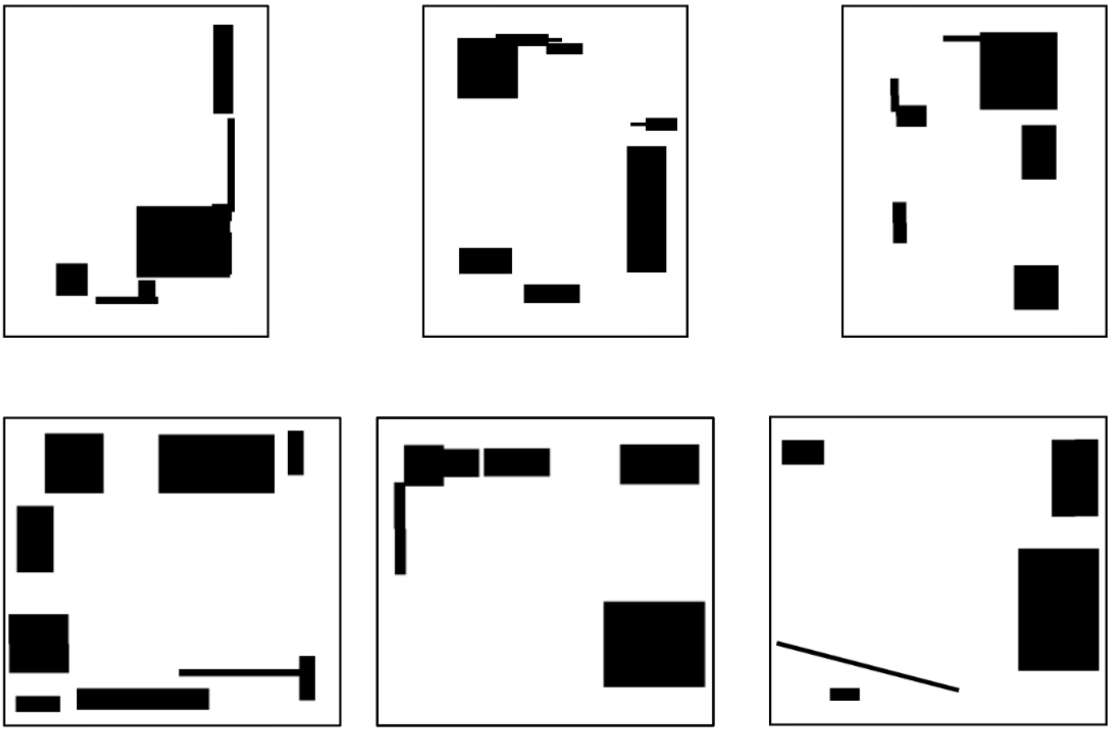


Figure 4.15: Room layout samples

The following tables show the detailed mean and standard deviation of the precision in the training. Except that ME-IRL performs better if it is trained from scratch w.r.t. P @ 1, methods incorporating with pre-training on our Tangram generally perform better.

	P @ 1	**P @ 2**	**P @ 3**
SL-Mean	0.54	0.659	0.767
SL-SD	0.252	0.103	0.045

Table 4.12: Mean and standard deviation of Score learning.

Omniglot and Multi-digit MNIST

We recommend reader to learn more about the Omniglot and Multi-digit MNIST for few-shot learning tasks. Standard N-way-K-shot learning tasks often run experiments as 20-way-1-shot,

	P @ 1	**P @ 2**	**P @ 3**
SL(Pre)-Mean	0.778	0.849	0.862
SL(Pre)-SD	0.176	0.07	0.035

Table 4.13: Mean and standard deviation of Score learning (Pre-trained).

	P @ 1	**P @ 2**	**P @ 3**
ME-Mean	1	0.905	0.841
ME-SD	0	0.201	0.201

Table 4.14: Mean and standard deviation of Max-entropy IRL.

	P @ 1	**P @ 2**	**P @ 3**
ME(Pre)-Mean	0.952	0.976	0.937
ME(Pre)-SD	0.218	0.109	0.134

Table 4.15: Mean and standard deviation of Max-entropy IRL (Pre-trained).

	P @ 1	**P @ 2**	**P @ 3**
GAIL-Mean	0.19	0.452	0.667
GAIL-SD	0.402	0.269	0.211

Table 4.16: Mean and standard deviation of GAIL.

	P @ 1	**P @ 2**	**P @ 3**
GAIL(Pre)-Mean	0.714	0.929	0.873
GAIL(Pre)-SD	0.463	0.179	0.166

Table 4.17: Mean and standard deviation of GAIL (Pre-trained).

20-way-5-shot, 5-way-1-shot and 5-way-5-shot.

The networks used for training MAML, ANIL and PrototypeNet share the same structures for extracting features as the one in Appendix A. They all apply Adam as the network optimizer and learning rate is 0.001.

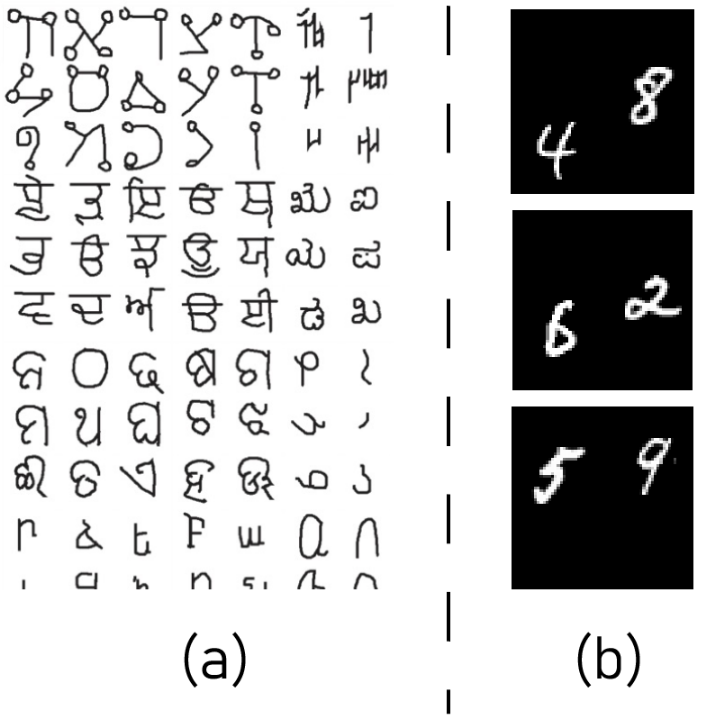


Figure 4.16: Human handwriting samples from Omniglot and Multi-digit MNIST.

Chapter 5

VALUENET: A New Dataset for Human Value Driven Dialogue System

Building a socially intelligent agent involves many challenges, one of which is to teach the agent to speak guided by its value like a human. However, value-driven chatbots are still understudied in the area of dialogue systems. Most existing datasets focus on commonsense reasoning or social norm modeling. In this work, we present a new large-scale human value dataset called VALUENET, which contains human attitudes on 21,374 text scenarios. The dataset is organized in ten dimensions that conform to the basic human value theory in intercultural research. We further develop a Transformer-based value regression model on VALUENET to learn the utility distribution. Comprehensive empirical results show that the learned value model could benefit a wide range of dialogue tasks. For example, by teaching a generative agent with reinforcement learning and the rewards from the value model, our method attains state-of-the-art performance on the personalized dialog generation dataset: PERSONA-CHAT. With values as additional features, existing emotion recognition models enable capturing rich human emotions in the context, which further improves the empathetic response generation performance in the EMPATHETICDIALOGUES dataset. To the best of our knowledge, VALUENET is the first large-scale text dataset for human value modeling, and we are the first one trying to incorporate a value model into emotionally intelligent dialogue systems.

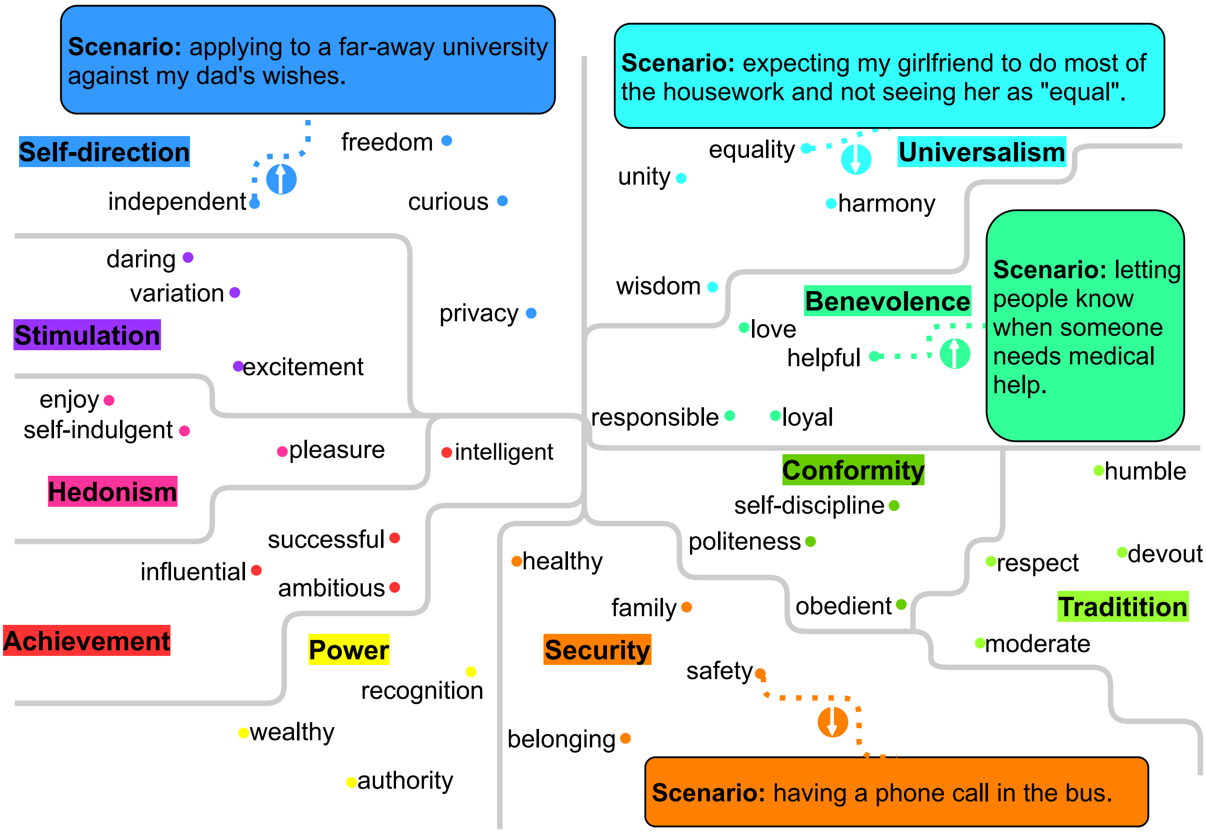


Figure 5.1: The presented VALUENET dataset with curated social scenarios organized by Schwartz values [Sch12].

5.1 Introduction

Value refers to desirable goals in human life. They guide the selection or evaluation of actions, policies, people, and events. A person's value priority or hierarchy profoundly affects his or her attitudes, beliefs, and traits, making it one core component of personality [Sch12]. In dialogue systems, modeling human values is a critical step towards building socially intelligent chatbots [QZL21]. By considering values, we can estimate user behavior and cognitive patterns from their utterances and generate responses that conform to the robot's persona configuration. For example, the robot is set to be aware of human values, and it invites Jerry to drink beers, but Jerry replies, "*You know that is tempting but is not good for our fitness*". The bot could read from the dialogue that Jerry prefers a healthy and self-disciplined lifestyle and steer its recommendation to healthier options in the future.

The development of socially intelligent chatbots has been one of the longest-running goals in artificial intelligence. Early dialogue systems such as Eliza [Wei66], Parry [CWH71], and more recent SimSimi[1], Panda Ichiro [OS18], Replika [FSR18], XiaoIce [ZGL20], were designed to mimic human behavior and incorporate emotional quotients (EQ) to some extent. There are also datasets and benchmarks for studying related problems, such as emotion recognition [MVC10, HCK18, PHM19, GMG20], personalized dialogue generation [ZDU18, LCC20], and empathetic dialogue generation [RSL19]. Even though value plays a fundamental and critical role in human EQ, there is a lack of explicit modeling of values in the dialogue domain, based on social domain theory. We have seen recent efforts about crowdsourcing social commonsense knowledge base or benchmarks [FHS20, SRC19, LBC21, HBB20, HBB21, GBS21]. However, it is not clearly shown how an agent can leverage this knowledge to estimate the users' value priorities or guide its own speaking and actions. In this paper, we aim to alleviate this problem and investigate the usage of a learned value function.

We start the study by curating a knowledge base of human values called VALUENET. Samples with value-related scenarios were identified based on value-defined keyword searching. Next, we asked Amazon Mechanical Turk workers about how the provided scenarios will affect one's value. This is based on the assumption that values underlie our attitudes; they are the guideline by which we evaluate things. Workers assess behaviors/events positively if they promote or protect the attainment of the goals we value. Behaviors/events are evaluated negatively if they hinder or threaten the attainment of these valued goals. The whole process gives us a large-scale (over 21k samples) multi-dimensional knowledge base of value. Figure 5.1 shows the overall structure of VALUENET. Each split represents a value dimension identified in the theory of basic human values [Sch12]. The figure also illustrates the value-related keywords and scenarios. The circular arrangement of the values represents a motivational continuum. By organizing data in such a structure, we anticipate the VALUENET to provide comprehensive coverage of different aspects of human values.

Next, we develop a Transformer-based value model to evaluate the utility score suggesting the positive or negative judgment given an utterance. We provide a detailed analysis of learning with multiple Transformer variants. Then we conduct a wide range of experiments to demonstrate that the value model could benefit EQ-related dialogue tasks: *(i)* By finetuning a generative

[1] https://simsimi.com/

agent with reinforcement learning and the reward from our value model, the method achieves state-of-the-art performance on the personalized dialogue dataset: PERSONA-CHAT [ZDU18]; *(ii)* By incorporating values as additional features, in EMPATHETICDIALOGUES [RSL19], we improve the emotion classification accuracy of existing models, which further facilitates the empathetic response generation; *(iii)* Visualization of the value model shows that it provides a numerical way of user profile modeling from their utterances.

In all, our contributions are two-fold. First, we present a large-scale dataset VALUENET for the modeling of human values that are well-defined in intercultural research. Second, we initiate to develop the value model learned from VALUENET to several EQ-related tasks and demonstrate its usage for building a value-driven dialogue system. Our methodology can be generalized to a wide range of interactive situations in socially aware dialogue systems [ZRR18], and human-robot interactions [YL17, LHA21].

5.2 Related Work

An abundance of related work inspires our work. Our work aims to make contributions to dialogue systems by incorporating the theory of human value. The dataset we collect shares a similar nature with multiple social commonsense benchmarks and knowledge bases. Besides, we apply our VALUENET for various dialogue tasks related to EQ.

5.2.1 Theory of Human Value and Utility

In the field of intercultural research, [Sch12] developed the theory of basic human values. The theory identifies ten basic personal values that are recognized across cultures and explains where they come from, as shown in Figure 5.1. The closer any two values in either direction around the circle, the more similar their underlying motivations are; the more distant, the more antagonistic their motivations. Note that dividing the value item domain into ten distinct values is an arbitrary convenience. It is reasonable to partition the value items into more or less fine-tuned distinct values according to the needs and objectives of one's analysis[2]. Similarly, in the economics

[2] A refinement of the theory [SCV12], partitions the same continuum into 19 more narrowly defined values that permit more precise explanation and prediction. We use the original 10-dimension version for simplicity in this paper.

field, the concept of utility [Fis70] is initially defined as a measure of pleasure or satisfaction in economics and ethics that drives human activities at all levels. Therefore, when we teach agents to speak and act in a socially intelligent way, an approach considering human value utilities should be adopted. In this paper, we aim to learn a utility function for each dimension of value and steer the dialogue system response generation accordingly.

5.2.2 Social Commonsense Benchmarks

[HBB20] present the ETHICS dataset, a benchmark that assesses a language model's knowledge of basic concepts of morality. SCRUPLES [LBC21] is a large-scale dataset with ethical judgments over real-life anecdotes, motivated by descriptive ethics. SOCIAL-CHEM-101 presented by [FHS20] is a corpus that catalogs rules-of-thumb as basic concept units for studying people's everyday social norms and moral judgments. They also propose Neural Norm Transformer to reason about previously unseen situations, generating relevant social rules-of-thumb. SOCIAL IQA [SRC19] is a large-scale benchmark for commonsense reasoning about social situations. [HBE17] present a task and corpus for predicting the preferable options from two sentences describing the scenarios that may involve social and cultural situations. Instead, in this work, we release a new dataset VALUENET that provides annotation of human attitudes from different value aspects.

5.2.3 Emotionally Intelligent Dialogue Datasets

Several datasets are presented to study emotion dynamics in dialogues. DailyDialog [LSS17] is a multi-turn dialogue dataset, which reflects the way of daily communication and provides emotion labels for speakers. [HCK18] present EmotionLines with emotions labeling on all utterances in each dialogue based on their textual content. MELD [PHM19] is an extension of EmotionLines for multi-modal multi-party emotion recognition. [MVC10] record a corpus SEMAINE of emotionally coloured conversations. [GMG20] propose a framework COSMIC for emotion recognition in conversations by considering mental states, events, actions, and cause-effect relations. DialogRE [YSC20] is the first human-annotated dialogue-based dataset for social relation inference [QLZ21]. PERSONA-CHAT [ZDU18] (revised in ConvAI2 [DLM20]) provides natural language profiles of speakers. Based on PERSONA-CHAT, [LCC20] propose a transmitter-receiver-based framework with explicitly human understanding modeling to enhance the quality of personalized dialogue generation. EMPATHETICDIALOGUES [RSL19] is a dataset that provides 25k conversations grounded in emotional situations. Each dialogue is grounded in a specific situation where a speaker was feeling a given emotion.

5.3 The VALUENET Dataset

During decision-making, people tend to pick the choice that aligns more with their own values. This work aims to provide a transferable knowledge base for human value modeling in natural language. To collect the VALUENET dataset, we curated social scenarios with value-related keywords and further annotated them via Amazon Mechanical Turk. Each sample in VALUENET is a social scenario description labeled with the annotator's attitude through a specific value lens.

The entire dataset is organized in a circular structure as shown in Figure 5.1, aligning with the theory of basic human values [Sch12]. The theory identifies ten universal values that are recognized throughout major cultures. The circular structure reflects the dynamic relations among these values, *i.e.,* the pursuit of some value may result in either accordance with another value or a conflict with another value. The ten distinct values can be further organized into four higher-order groups.

- **Openness to change**: self-direction, stimulation
- **Self-enhancement**: hedonism, achievement, power
- **Conservation**: security, conformity, tradition
- **Self-transcendence**: benevolence, universalism

We describe the collection details of the VALUENET in the following sections.

5.3.1 Social Scenario Curation

We curated a set of 21,374 social scenarios from the large-scale social-related database SOCIAL-CHEM-101 [FHS20]. Value-related scenarios are retrieved with value keywords after lemmatization and stemming. There are three sets of keywords identified for each dimension of Schwartz value: (1) the keywords in the original definition of each value in Schwartz's paper [Sch12]; (2) words that share a similar meaning, words that are often used to describe the original keywords, and words that are triggered by (strongly associated with) the original keywords[3]; (3) words that are near the original keywords in the GloVe [PSM14a] embedding space. The value keywords are verified and confirmed by humans as listed in Figure 5.2.

5.3.2 Value-Aspect Attitude Annotation

We crowdsourced people's attitudes to the curated scenarios on Amazon Mechanical Turk (AMT). Figure 5.3 shows an example.

We follow a strict procedure to select qualified workers and ensure the workers understand the concept of each value we ask. In Figure 5.3, the definition of `BENEVOLENCE` is shown to the workers throughout the entire annotation process. To further help the understanding, we include three examples in each assignment with correct answers being “yes", “no", and “unrelated", respectively. The worker is then required to answer a prerequisite question correctly to proceed to the formal survey. The formal survey is composed of ten questions, including two hidden qualification checking questions. Before publishing on the AMT, two Ph.D. students prepared the qualification questions by annotating a small subset of the curated scenarios. Their agreed samples (100 in total) were randomly inserted into the survey for worker selection. The selection procedure was done in the value dimensions with more scenarios to get a large pool of qualified workers and a relatively balanced final dataset across different values. The complete Mechanical Turk interface is attached in the Appendix for reference.

A total of 681 experienced AMT workers participated in our VALUENET annotation. 443 of them passed the qualification test. Each scenario is assigned to four different workers. The original inter-annotator agreement is 64.9%, and the Fleiss' kappa score [Fle71] among the workers is 0.48, which considers the possibility of the agreement by chance. Keeping the scope of VALUENET in commonly-agreed attitudes towards social scenarios, we only retain the samples with three or more agreements. Figure 5.4 shows the sample size of each value split

[3] We use datamuse (`https://www.datamuse.com/api/`) for this purpose.

ValueNet	train	valid	test	total
# samples	16,030	3,206	2,138	21,374
average # tokens	12.05	12.09	12.26	12.07
unique # tokens	12,452	5,292	4,112	14,143

Table 5.1: Statistics of the ValueNet dataset.

		F1(-1)	**F**1(0)	**F**1(1)	**P**(-1)	**P**(0)	**P**(1)	**R**(-1)	**R**(0)	**R**(1)	**Acc.**↑	**MSE**↓
ValueNet (original)	fastText	0.70	0.46	0.43	0.65	0.47	0.55	**0.76**	0.44	0.35	0.58	0.66
	BERT	**0.73**	0.50	0.51	0.72	0.46	0.71	0.74	0.55	0.39	0.61	0.39
	DistilBERT	0.71	0.52	0.47	0.74	0.45	0.69	0.68	0.62	0.36	0.60	**0.37**
	RoBERTa	0.65	0.51	0.34	0.74	0.40	0.71	0.58	0.69	0.22	0.55	0.41
	BART	0.00	**0.76**	0.54	0.00	0.70	0.60	0.00	**0.83**	**0.49**	**0.67**	0.52
ValueNet (balanced)	fastText	0.70	0.48	0.43	0.64	0.50	0.54	**0.76**	0.45	0.36	0.59	0.68
	BERT	0.67	0.48	0.51	0.73	0.42	0.61	0.62	0.58	0.43	0.57	0.40
	DistilBERT	0.66	0.49	0.50	0.74	0.41	0.61	0.60	0.60	0.43	0.57	0.40
	RoBERTa	0.65	0.51	0.34	0.74	0.40	0.71	0.58	0.69	0.22	0.55	0.41
	BART	0.00	0.75	0.51	0.00	0.72	0.57	0.00	0.77	0.47	0.65	0.55
ValueNet (augmented)	fastText	0.58	0.52	0.29	0.72	0.40	0.65	0.49	0.75	0.18	0.52	0.59
	BERT	0.67	0.55	0.41	0.78	0.43	**0.78**	0.58	0.76	0.28	0.58	0.38
	DistilBERT	0.68	0.57	0.41	**0.79**	0.44	**0.78**	0.59	0.78	0.28	0.60	0.38
	RoBERTa	0.70	0.56	0.41	0.78	0.45	0.75	0.64	0.74	0.28	0.61	0.40
	BART	0.00	0.74	**0.57**	0.00	**0.75**	0.49	0.00	0.73	0.66	0.64	0.46

Table 5.2: Value modeling performance in the ValueNet dataset. Bold items are the best in each metric column.

and their label distribution.

The data is split into the train (75%), valid (15%), and test (10%). Similar to the polarity in sentiment analysis [KWM11], we quantify the annotated labels into numerical values: yes (positive): +1, no (negative): -1, unrelated (neutral): 0. We denote the numerical values as **utility** to describe the effect of a scenario on one's value. In other words, for people who appreciate a certain value, actions with a higher utility in this value dimension would be more desirable to them.

Table 5.1 shows more statistical details about the ValueNet dataset. In total, we collected 21,374 samples covering a wide range of scenarios in daily social life.

Acc.	ACH	BEN	CON	HED	POW	SEC	SD	STI	TRA	UNI
ValueNet (original)	**0.56**	**0.68**	0.82	**0.63**	0.35	**0.52**	0.45	**0.58**	0.60	**0.51**
ValueNet (balanced)	0.53	0.58	**0.83**	**0.63**	**0.41**	0.50	0.42	0.53	0.61	0.50
ValueNet (augmented)	0.48	0.66	0.82	0.58	0.33	0.47	**0.48**	0.49	**0.64**	0.42

Table 5.3: Accuracies of the BERT [DCL18] value model across different value dimensions in the ValueNet dataset.

5.4 Value Modeling

We experiment using Transformer-based pre-trained language models for modeling human values from the VALUENET dataset.

5.4.1 Task Formalization

Given a social scenario *s*, we wish to learn a value function that models the utility distribution of *s* from the ten Schwartz value dimensions:

$$\mathbf{V}(s) = [V_{\texttt{SEC}}(s), V_{\texttt{POW}}(s), V_{\texttt{ACH}}(s), V_{\texttt{HED}}(s), V_{\texttt{STI}}(s), V_{\texttt{SD}}(s), V_{\texttt{UNI}}(s), V_{\texttt{BEN}}(s), V_{\texttt{CON}}(s), V_{\texttt{TRA}}(s)]$$

where $V_{\texttt{\$VALUE}}(\cdot) \in [-1, 1]$ and $V_{\texttt{\$VALUE}}(\cdot) \in \mathbb{R}$.

5.4.2 Model

Pre-trained language model variants: BERT [DCL18], RoBERTa [LOG19], DistilBERT [SDC19], BART [LLG19] are investigated for learning the value function. A custom input format constructed as '`[CLS] [$VALUE] s`' is fed into a Transformer encoder, *i.e.,*

$$V_{\texttt{\$VALUE}}(\texttt{s}) = \text{TRM}(\texttt{[CLS] [\$VALUE] s}), \tag{5.1}$$

where TRM denotes the Transformer encoder, `[CLS]` is the special token for regression or classification, and `[$VALUE]` are special tokens we define to prompt the language models the value dimension we are interested in [LL21, BMR20, LR21]. In order to get the ten-dimensional output $\mathbf{V}(s)$, a batch size of 10 is forwarded through the model. For the BERT, DistilBERT, and RoBERTa, a regression head is put on top of the models and they are trained with the Mean Squared Error (MSE) loss. We use the regression model with *sigmoid* activation to get a continuous estimation of the utility in the range of $[-1, 1]$. To evaluate the effect of different loss functions, we train the BART model with three output classes and the cross-entropy loss.

5.4.3 Result and Analysis

The learning performance of using fastText[4] [JGB17] and Transformer variants are reported in Table 5.2. All Transformers are trained for 40 epochs with a learning rate of 5e$-$6. The prediction precision, recall, **F**1 score, and accuracy for regression models are computed by the utility rounded to the nearest integer.

In general, pre-trained language models perform better than the fastText baseline. However, there is not a noticeable difference between the Transformer variants. The prediction accuracy of BART is the highest among all models because it is explicitly trained for classification purposes. BERT and DistilBERT get the lowest MSE in terms of regression performance.

Observing the sample imbalance across different value splits and labels (Figure 5.4), we release another two versions of VALUENET: VALUENET (balanced) and VALUENET (augmented). The original dataset is balanced by subsampling the negative and neutral data of the largest value split (`BENEVOLENCE`). Moreover, we augment the neutral class of the original VALUENET by assigning AMT results with less worker agreement to “unrelated". Data distribution of the balanced and augmented versions of VALUENET are illustrated in the Appendix. By analyzing the prediction accuracy in different value splits (Table 5.3), we find that reducing the sample number of `BENEVOLENCE` hurts the model performance in that dimension. Looking at the **F**1 score of each class in Table 5.2, we conclude that augmenting the neutral class improves the **F**1(0) but reduces **F**1(1) and **F**1(-1). We leave it a future work to further improve the value modeling performance.

In the next sections, we show how the learned value function could benefit EQ-related tasks and help build a value-driven dialogue system.

5.5 PERSONA-CHAT

As values are closely related to one’s personality, we first assess our value model on a personalized dialogue dataset: PERSONA-CHAT [ZDU18]. The PERSONA-CHAT dataset contains multi-turn dialogues conditioned on personas. Each persona is encoded by at least 5 sentences

[4] `https://github.com/facebookresearch/fastText`

Model	Original			Revised		
	Hits@1(%) ↑	**Ppl.**↓	**F1(%)** ↑	**Hits@1(%)** ↑	**Ppl.**↓	**F1(%)** ↑
SEQ2SEQ-ATTN	12.5	35.07	16.82	9.8	39.54	15.52
$\mathscr{P}^2$BOT [LCC20]	–	15.12	19.77	–	18.89	19.08
GPT2 (MLE) [RWC19]	$14.51_{[0.05]}$	$17.23_{[0.03]}$	$18.74_{[0.01]}$	$10.31_{[0.07]}$	$20.64_{[0.11]}$	$18.29_{[0.05]}$
GPT2 + Value (Ours)	$16.44_{[0.10]}$	$16.83_{[0.06]}$	$18.76_{[0.02]}$	$12.19_{[0.03]}$	$19.98_{[0.06]}$	$17.88_{[0.05]}$
DialoGPT (MLE) [ZSG19]	$20.20_{[0.04]}$	$14.38_{[0.05]}$	$20.16_{[0.04]}$	$15.80_{[0.03]}$	$17.35_{[0.05]}$	$19.08_{[0.08]}$
DialoGPT + Value (Ours)	$\mathbf{20.97}_{[0.08]}$	$\mathbf{13.84}_{[0.03]}$	$\mathbf{20.22}_{[0.01]}$	$\mathbf{18.83}_{[0.03]}$	$\mathbf{17.01}_{[0.03]}$	$\mathbf{19.79}_{[0.10]}$

Table 5.4: Next Utterance Prediction Performance on PERSONA-CHAT [ZDU18]. We report the standard deviation $[\sigma]$ (across 5 runs) of the models we trained.

of textual description, termed a profile. Example profile sentences are “I like to ski", “I enjoying walking for exercise", “I have four children", *etc.* The dataset is composed of 8,939 dialogues for training, 1,000 for validation, and 968 for testing. It also provides *revised* personas by rephrasing, generalizing or specializing the *original* ones. The dataset we use for experiments is public available in ParlAI[5].

5.5.1 Task Formalization

Given the agent’s self persona profile $\mathbf{p} = [p_1, p_2, ..., p_N]$ and the dialogue history up to the t-th turn $\mathbf{h}_t^s = (x_1^u, x_1^s, ..., x_t^u)$, x_i^u is the i-th utterance by Person 1 played by the user, x_i^s is the i-th utterance by Person 2 played by the system, we evaluate the model’s performance on predicting the next utterance x_t^s.

5.5.2 Model

A decoder-only Transformer-based model is used to estimate the generation distribution $p_\theta(x_t^s \mid \mathbf{h}_t^s, \mathbf{p})$, where θ is the model parameter. Following the practice proposed in [GLC18], the model is firstly trained with Maximum Likelihood Estimation (MLE) to ensure generating fluent responses. Then we took an interleaving of supervised training (MLE) and reinforcement learning. We use the REINFORCE policy gradient algorithm [Wil92] in our experiment, and the reward assignment is described as following.

Denote $\mathbf{V}(p_i)$ and $\mathbf{V}(\hat{x}_i^s)$ to describe the estimation of the agent’s value from its profile sentence p_i and generated response $\hat{x}_i^s$, respectively. We want the reward to promote the alignment of the agent’s profile and utterances in the value space. For instance, if the agent has profile ‘*I*

[5] `https://parl.ai/projects/convai2/`

Algorithm 2 Personalized Dialogue Value Matching

```
Input: $[\mathbf{V}(p_1),...,\mathbf{V}(p_N)],[\mathbf{V}(\hat{x}^s_1),...,\mathbf{V}(\hat{x}^s_T)]$
Output: reward $R$
  for $t = 1,2,...,T$ do
      $r_t \leftarrow -1$
      $m_t \leftarrow -1$
      for $i = 1,2,...,N$ do
          if $\mathbf{V}(p_i)\cdot\mathbf{V}(\hat{x}^s_t) > r_t$ then
              $r_t \leftarrow \mathbf{V}(p_i)\cdot\mathbf{V}(\hat{x}^s_t)$
              $m_t \leftarrow i$
          end if
      end for
  end for
  $\gamma_i \leftarrow 1, i = 1,2,...,N$
  for $t = 1,2,...,T$ do
      $\gamma_{m_t} \leftarrow \gamma_{m_t} + 1$
  end for
  $R \leftarrow 0$
  for $t = 1,2,...,T$ do
      $R \leftarrow R + \text{sign}(r_t)\cdot|r_t|^{\text{sign}(r_t)\cdot\gamma_{m_t}}$
  end for
  return $R/N$
```

like venture' and '*I have a dog*', and it says '*I plan to ski this weekend*' and also '*Do you like skiing*'. Both utterances should be aligned with the first persona. Here we propose a simple yet effective searching algorithm (Algorithm 2) to find a match between $[\mathbf{V}(p_1),\mathbf{V}(p_2),...,\mathbf{V}(p_N)]$ and $[\mathbf{V}(\hat{x}^s_1),\mathbf{V}(\hat{x}^s_2),...,\mathbf{V}(\hat{x}^s_T)]$ and return a reward R. N is the number of profile sentences and T is the length of the generated dialogue. $\mathbf{V}$ is normalized to ensure $|r_t| \leq 1$. Intuitively, the discount argument γ prevents the language model from repeating the same fact in the agent's profile.

5.5.3 Setup

We evaluate the same generative model in both generation and ranking settings. In the response ranking setup, the candidates are scored with their log-likelihood. For the GPT-2 [RWC19] and DialoGPT [ZSG19] we have finetuned, we train them for 5k steps with a training batch size of 8. The learning rate is set to 2e−6. For an illustration of computational requirements, the training with MLE on 4 NVIDIA Tesla V100 takes ∼1 hours, and the reinforcement learning takes ∼30 minutes.

5.5.4 Result and Analysis

Following [ZDU18] and [LCC20], we report the **Hits@1**, **Perplexity** and **F1** to evaluate the methods in Table 5.4. By the submission of this paper, $\mathscr{P}^2$BOT [LCC20] is the state-of-the-art model reported in this task. We also include a generative baseline using SEQ2SEQ with attention mechanism [BCB14] for comparison. As observed, in terms of all the metrics we evaluated, finetuning GPT2 or DialoGPT2 models with our value function provides a significant performance boost compared to simply training them with MLE. Our DialoGPT + Value model achieves new state-of-the-art performance on perplexity and **F1**.

5.6 EMPATHETICDIALOGUES

EMPATHETICDIALOGUES [RSL19] provides 25k conversations grounded in emotional situations. It aims to test the dialogue system's capability to produce empathetic responses. Each dialogue is grounded in a specific situation where a speaker was feeling a given emotion, with a listener responding. In this section, we demonstrate how we could leverage VALUENET to improve the emotion classification accuracy and further improve the empathetic response generation.

5.6.1 Emotion Classification

An auxiliary task that is highly related to empathetic dialogue generation is emotion classification. In EMPATHETICDIALOGUES, each situation is written in association with a given emotion label. A total of 32 emotion labels were annotated to cover a broad range of positive and negative emotions.

Model

Given the situation context s, a pre-trained BERT model encodes s and gets the sentence representation from its pooling layer of the [CLS] token. The same context is parsed by our pre-trained value model to get a ten-dimensional vector, which serves as an additional feature for the classification:

$$
\begin{aligned}
h_s &= \text{BERT}(s), \\
v_s &= \mathbf{V}(s), \\
e &= softmax(\mathbf{W} \cdot ([h_s; v_s]) + \mathbf{b}),
\end{aligned}
\tag{5.2}
$$

where **W** and **b** are learnable parameters.

Result

We compare the performance between our implementation and the baseline that directly applies the BERT model for emotion classification. As shown in Table 5.5, the additional value information benefits emotion classification from both the DistilBERT and BERT models. Our method obtains a **relative** improvement of 5.2% on DistilBERT and 6.4% on BERT.

Model	**Accuracy** (σ)	
fastText	42.27 ± 0.3%	
DistilBERT	41.81 ± 0.2%	
DistilBERT + Value	43.98 ± 0.2%	+**2.17**%
BERT	42.93 ± 0.1%	
BERT + Value	**45.67** ± 0.3%	+**2.74**%

Table 5.5: Emotion classification performance in EMPATHETICDIALOGUES [RSL19].

5.6.2 Empathetic Dialogue Generation

We further check whether our value model helps the empathetic dialogue generation. EMPATHETICDIALOGUES applies PREPEND-K, a strategy to add supervised information to data, when predicting the utterance given the dialogue history and the situation. We apply the strategy of prepending the top-k emotion labels for dialogue generation. The top predicted label from the classifiers of emotion is prepended to the beginning of the token sequence as encoder input, as below:

- **Original**: “I finally got promoted!"
- **Prepend-1 emotion**: “*proud* I finally got promoted!"

Result

The results are shown in Table 5.6. As observed, prepending emotion tokens provides extra context and improves the generation performance of GPT2 and DialoGPT. Since incorporating value improves the emotion classification accuracy, it further improves the generation quality.

Model	**Ppl.**↓
EmoPrepend-1 [RSL19]	24.30
GPT	14.74
GPT + Emotion (w/o Value)	14.46
GPT + Emotion (w/ Value)	14.01
DialoGPT	13.48
DialoGPT + Emotion (w/o Value)	12.32
DialoGPT + Emotion (w/ Valued)	**12.12**

Table 5.6: Empathetic dialogue generation in EMPATHETICDIALOGUES [RSL19]. EmoPrepend-1: input prepending emotion from an external classifier.

5.7 Value Profiling

For a more comprehensive understanding, we visualize the 10-dimensional value of four example scenarios in Figure 5.5. As shown, the value model provides a numerical speaker profile. For instance, saying “forcing my daughter to sleep in her own bed" implies that the speaker values power and conformity; saying “I miss mom" implies that the speaker values benevolence; saying “not wanting people to use my property without permissions" implies the speaker is self-directed and values security. The last example “I forgot how to be happy" results a small radar graph. It suggests that even the model could predict the overall polarity pretty well, there is still space to improve its capability of distinguishing different values.

5.8 Appendix

Here we provide the value descriptions [Sch12].

Self-Direction

Defining goal: independent thought and action–choosing, creating, exploring. Self-direction derives from organismic needs for control and mastery and interactional requirements of autonomy and independence. (creativity, freedom, choosing own goals, curious, independent) [self-respect, intelligent, privacy]

Stimulation

Defining goal: excitement, novelty, and challenge in life. Stimulation values derive from the organismic need for variety and stimulation in order to maintain an optimal, positive, rather than threatening, level of activation. This need probably relates to the needs underlying self-direction values. (a varied life, an exciting life, daring)

Hedonism

Defining goal: pleasure or sensuous gratification for oneself. Hedonism values derive from organismic needs and the pleasure associated with satisfying them. Theorists from many disciplines mention hedonism. (pleasure, enjoying life, self-indulgent)

Achievement

Defining goal: personal success through demonstrating competence according to social standards. Competent performance that generates resources is necessary for individuals to survive and for groups and institutions to reach their objectives. As defined here, achievement values emphasize demonstrating competence in terms of prevailing cultural standards, thereby obtaining social approval. (ambitious, successful, capable, influential) [intelligent, self-respect, social recognition]

Power

Defining goal: social status and prestige, control or dominance over people and resources. The functioning of social institutions apparently requires some degree of status differentiation. A dominance/submission dimension emerges in most empirical analyses of interpersonal relations both within and across cultures. To justify this fact of social life and to motivate group members to accept it, groups must treat power as a value. Power values may also be transformations of individual needs for dominance and control. Value analysts have mentioned power values as well. (authority, wealth, social power) [preserving my public image, social recognition]

Both power and achievement values focus on social esteem. However, achievement values (e.g., ambitious) emphasize the active demonstration of successful performance in concrete interaction, whereas power values (e.g., authority, wealth) emphasize the attainment or preservation of a dominant position within the more general social system.

Security

Defining goal: safety, harmony, and stability of society, of relationships, and of self. Security values derive from basic individual and group requirements. Some security values serve primarily individual interests (e.g., clean), others wider group interests (e.g., national security). Even the latter, however, express, to a significant degree, the goal of security for self or those with whom one identifies. (social order, family security, national security, clean, reciprocation of favors) [healthy, moderate, sense of belonging]

Conformity

Defining goal: restraint of actions, inclinations, and impulses likely to upset or harm others and violate social expectations or norms. Conformity values derive from the requirement that individuals inhibit inclinations that might disrupt and undermine smooth interaction and group functioning. As I define them, conformity values emphasize self-restraint in everyday interaction, usually with close others. (obedient, self-discipline, politeness, honoring parents and elders) [loyal, responsible]

Tradition

Defining goal: respect, commitment, and acceptance of the customs and ideas that one's culture or religion provides. Groups everywhere develop practices, symbols, ideas, and beliefs that represent their shared experience and fate. These become sanctioned as valued group customs and traditions. They symbolize the group's solidarity, express its unique worth, and contribute to its survival (Durkheim, 1912/1954; Parsons, 1951). They often take the form of religious rites, beliefs, and norms of behavior. (respect for tradition, humble, devout, accepting my portion in life) [moderate, spiritual life]

Tradition and conformity values are especially close motivationally; they share the goal of subordinating the self to socially imposed expectations. They differ primarily in the objects to which one subordinates the self. Conformity entails subordination to persons with whom one frequently interacts—parents, teachers, and bosses. Tradition entails subordination to more abstract objects—religious and cultural customs and ideas. As a corollary, conformity values exhort responsiveness to current, possibly changing expectations. Tradition values demand responsiveness to immutable expectations from the past.

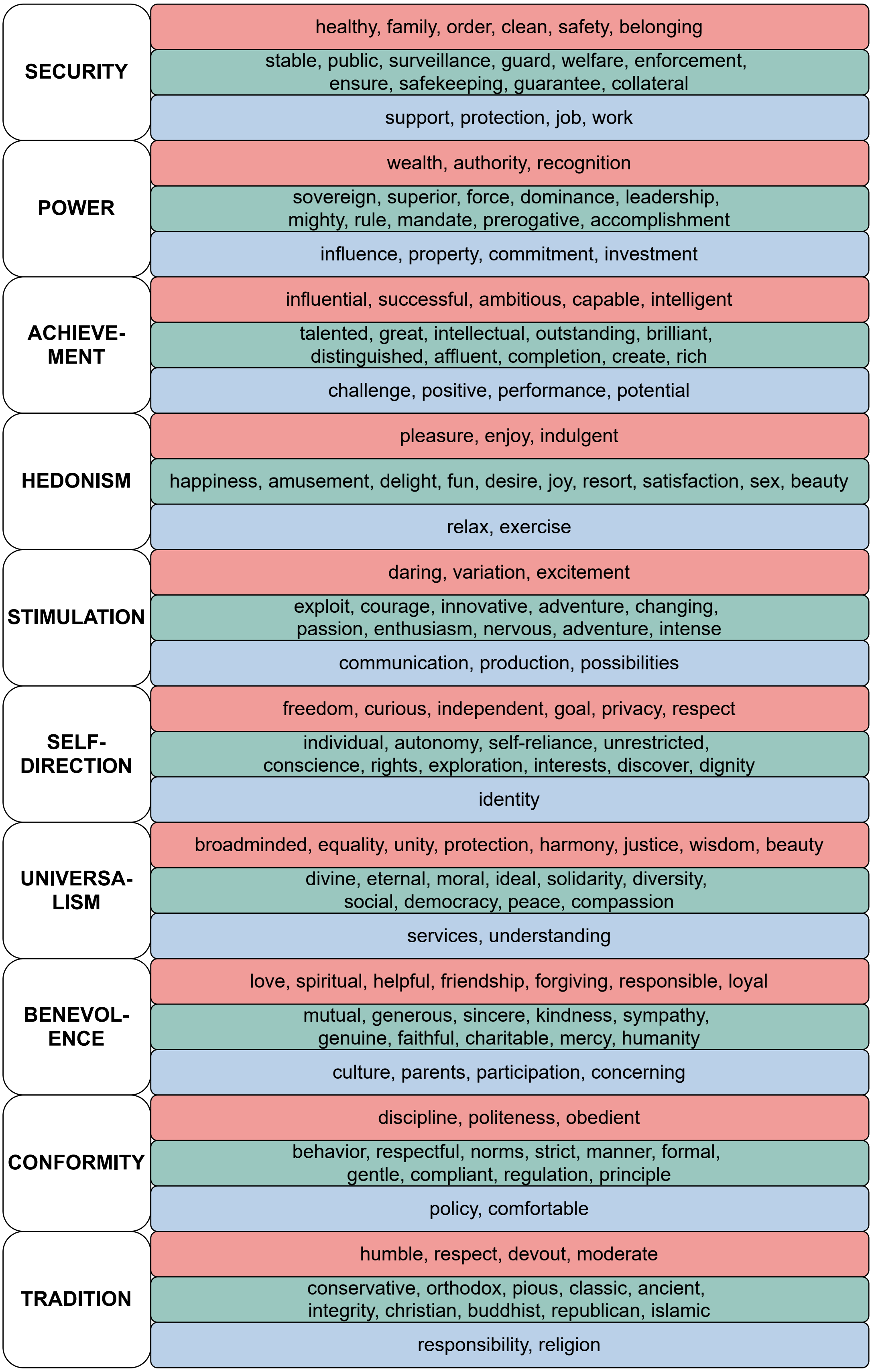
Benevolence

Defining goal: preserving and enhancing the welfare of those with whom one is in frequent personal contact (the 'in-group'). Benevolence values derive from the basic requirement for smooth group functioning and from the organismic need for affiliation. Most critical are relations within the family and other primary groups. Benevolence values emphasize voluntary concern for others' welfare. (helpful, honest, forgiving, responsible, loyal, true friendship, mature love) [sense of belonging, meaning in life, a spiritual life].

Benevolence and conformity values both promote cooperative and supportive social relations. However, benevolence values provide an internalized motivational base for such behavior. In contrast, conformity values promote cooperation in order to avoid negative outcomes for self. Both values may motivate the same helpful act, separately or together.

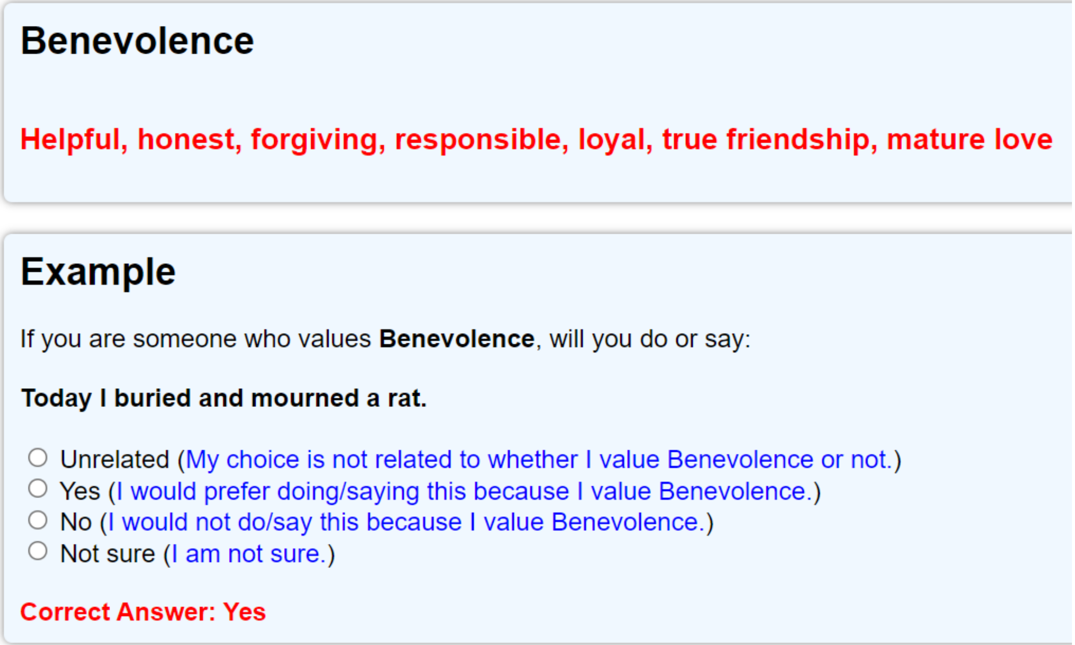
Universalism

Defining goal: understanding, appreciation, tolerance, and protection for the welfare of all people and for nature. This contrasts with the in-group focus of benevolence values. Universalism values derive from survival needs of individuals and groups. But people do not recognize these needs until they encounter others beyond the extended primary group and until they become aware of the scarcity of natural resources. People may then realize that failure to accept others who are different and treat them justly will lead to life-threatening strife. They may also realize that failure to protect the natural environment will lead to the destruction of the resources on which life depends. Universalism combines two subtypes of concern—for the welfare of those in the larger society and world and for nature (broadminded, social justice, equality, world at peace, world of beauty, unity with nature, wisdom, protecting the environment)[inner harmony, a spiritual life]



SECURITY	healthy, family, order, clean, safety, belonging	stable, public, surveillance, guard, welfare, enforcement, ensure, safekeeping, guarantee, collateral	support, protection, job, work
POWER	wealth, authority, recognition	sovereign, superior, force, dominance, leadership, mighty, rule, mandate, prerogative, accomplishment	influence, property, commitment, investment
ACHIEVE-MENT	influential, successful, ambitious, capable, intelligent	talented, great, intellectual, outstanding, brilliant, distinguished, affluent, completion, create, rich	challenge, positive, performance, potential
HEDONISM	pleasure, enjoy, indulgent	happiness, amusement, delight, fun, desire, joy, resort, satisfaction, sex, beauty	relax, exercise
STIMULATION	daring, variation, excitement	exploit, courage, innovative, adventure, changing, passion, enthusiasm, nervous, adventure, intense	communication, production, possibilities
SELF-DIRECTION	freedom, curious, independent, goal, privacy, respect	individual, autonomy, self-reliance, unrestricted, conscience, rights, exploration, interests, discover, dignity	identity
UNIVERSA-LISM	broadminded, equality, unity, protection, harmony, justice, wisdom, beauty	divine, eternal, moral, ideal, solidarity, diversity, social, democracy, peace, compassion	services, understanding
BENEVOL-ENCE	love, spiritual, helpful, friendship, forgiving, responsible, loyal	mutual, generous, sincere, kindness, sympathy, genuine, faithful, charitable, mercy, humanity	culture, parents, participation, concerning
CONFORMITY	discipline, politeness, obedient	behavior, respectful, norms, strict, manner, formal, gentle, compliant, regulation, principle	policy, comfortable
TRADITION	humble, respect, devout, moderate	conservative, orthodox, pious, classic, ancient, integrity, christian, buddhist, republican, islamic	responsibility, religion

Figure 5.2: Ten universal human values and related keywords for social scenario curation. Red: keywords in the original value definition [Sch12]; Green: associated keywords found with datamuse; Blue: associated keywords found with GloVe embedding.



Benevolence

Helpful, honest, forgiving, responsible, loyal, true friendship, mature love

Example

If you are someone who values **Benevolence**, will you do or say:

Today I buried and mourned a rat.

○ Unrelated (My choice is not related to whether I value Benevolence or not.)
○ Yes (I would prefer doing/saying this because I value Benevolence.)
○ No (I would not do/say this because I value Benevolence.)
○ Not sure (I am not sure.)

Correct Answer: Yes

Figure 5.3: Value-aspect attitude annotation in AMT.

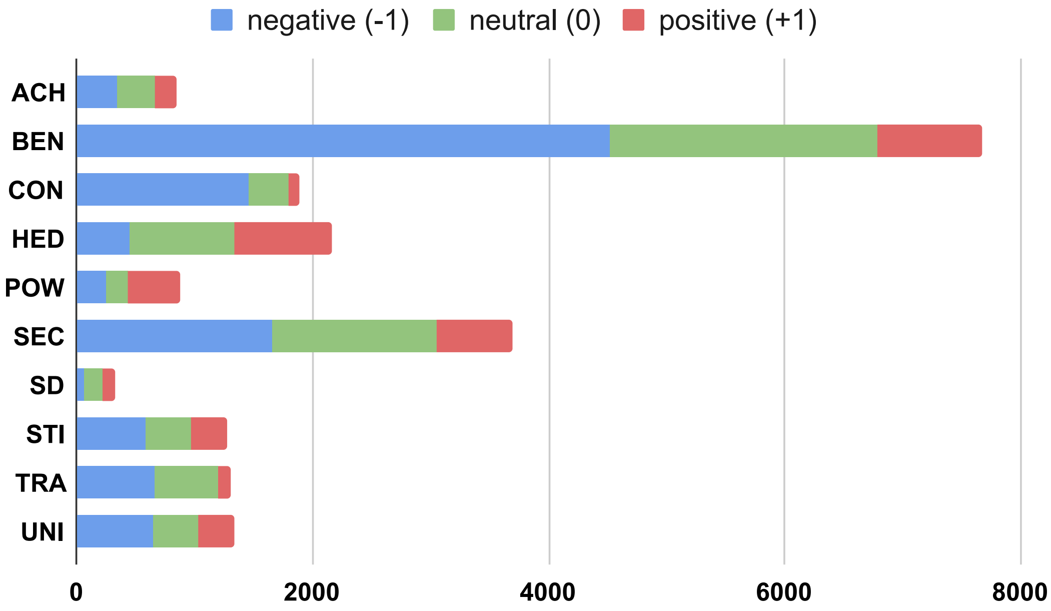


Figure 5.4: The sample number and label distribution of each value split in the VALUENET.

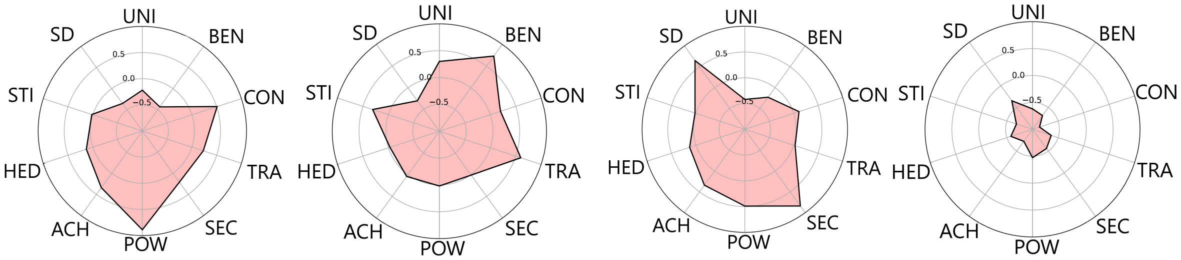
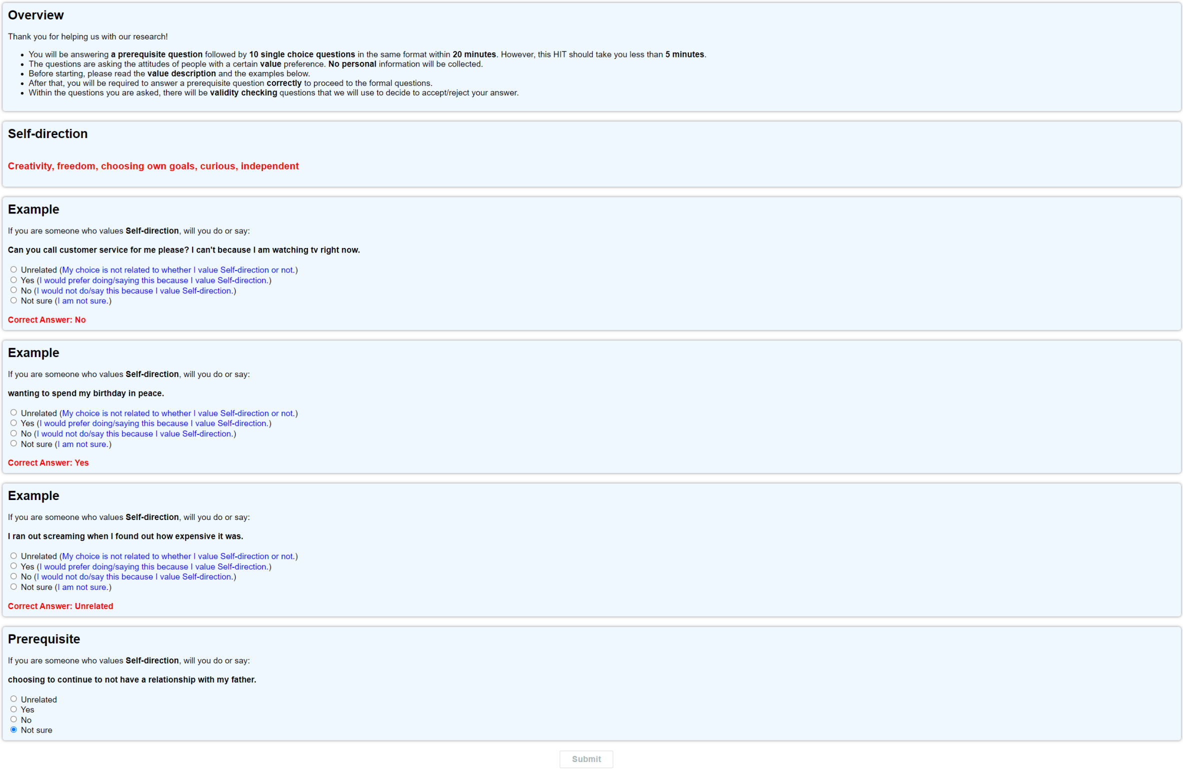


Figure 5.5: Value visualization of example scenarios: (a) forcing my daughter to sleep in her own bed; (b) I miss mom; (c) not wanting people to use my property without permissions; (d) I forgot how to be happy.



Overview

Thank you for helping us with our research!

- You will be answering **a prerequisite question** followed by **10 single choice questions** in the same format within **20 minutes**. However, this HIT should take you less than **5 minutes**.
- The questions are asking the attitudes of people with a certain **value** preference. **No personal** information will be collected.
- Before starting, please read the **value description** and the examples below.
- After that, you will be required to answer a prerequisite question **correctly** to proceed to the formal questions.
- Within the questions you are asked, there will be **validity checking** questions that we will use to decide to accept/reject your answer.

Self-direction

Creativity, freedom, choosing own goals, curious, independent

Example

If you are someone who values **Self-direction**, will you do or say:

Can you call customer service for me please? I can't because I am watching tv right now.

○ Unrelated (My choice is not related to whether I value Self-direction or not.)
○ Yes (I would prefer doing/saying this because I value Self-direction.)
○ No (I would not do/say this because I value Self-direction.)
○ Not sure (I am not sure.)

Correct Answer: No

Example

If you are someone who values **Self-direction**, will you do or say:

wanting to spend my birthday in peace.

○ Unrelated (My choice is not related to whether I value Self-direction or not.)
○ Yes (I would prefer doing/saying this because I value Self-direction.)
○ No (I would not do/say this because I value Self-direction.)
○ Not sure (I am not sure.)

Correct Answer: Yes

Example

If you are someone who values **Self-direction**, will you do or say:

I ran out screaming when I found out how expensive it was.

○ Unrelated (My choice is not related to whether I value Self-direction or not.)
○ Yes (I would prefer doing/saying this because I value Self-direction.)
○ No (I would not do/say this because I value Self-direction.)
○ Not sure (I am not sure.)

Correct Answer: Unrelated

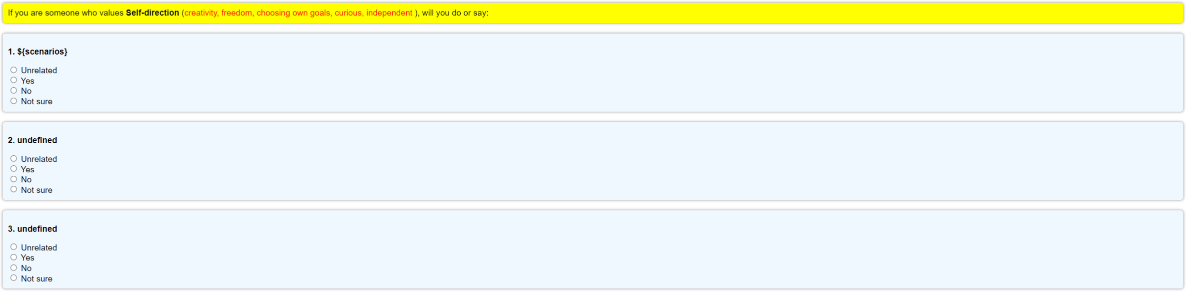
Prerequisite

If you are someone who values **Self-direction**, will you do or say:

choosing to continue to not have a relationship with my father.

○ Unrelated
○ Yes
○ No
● Not sure

Figure 5.6: Amazon mechanical turk interface (prerequiste).



If you are someone who values **Self-direction** (creativity, freedom, choosing own goals, curious, independent), will you do or say:

1. ${scenarios}

○ Unrelated
○ Yes
○ No
○ Not sure

2. undefined

○ Unrelated
○ Yes
○ No
○ Not sure

3. undefined

○ Unrelated
○ Yes
○ No
○ Not sure

Figure 5.7: Amazon mechanical turk interface (formal).

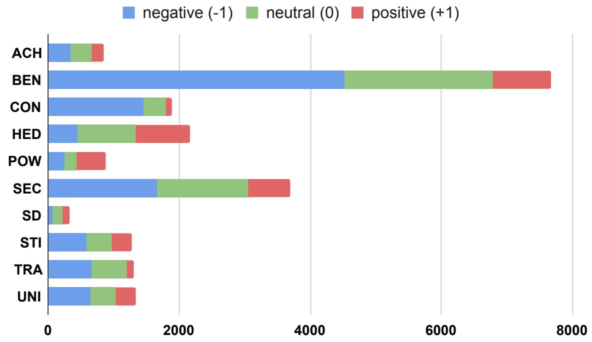


Figure 5.8: The sample number and label distribution of each value split in the VALUENET (original).

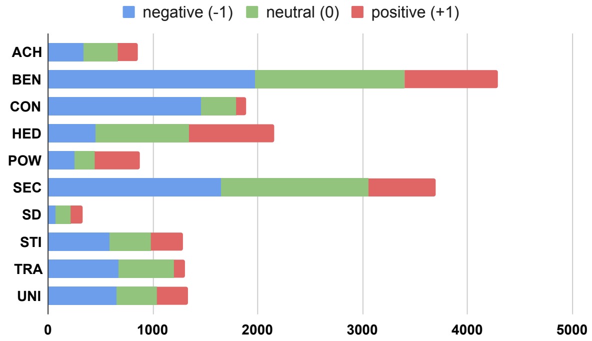


Figure 5.9: The sample number and label distribution of each value split in the VALUENET (balanced).

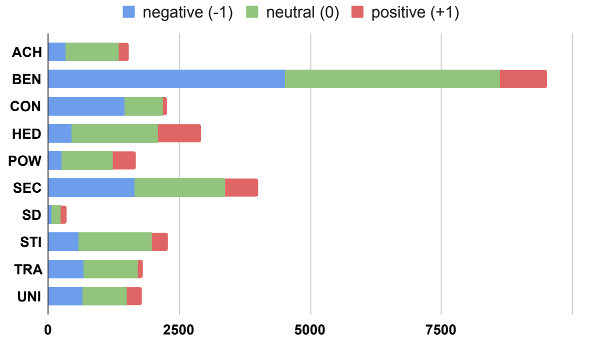


Figure 5.10: The sample number and label distribution of each value split in the VALUENET (augmented).

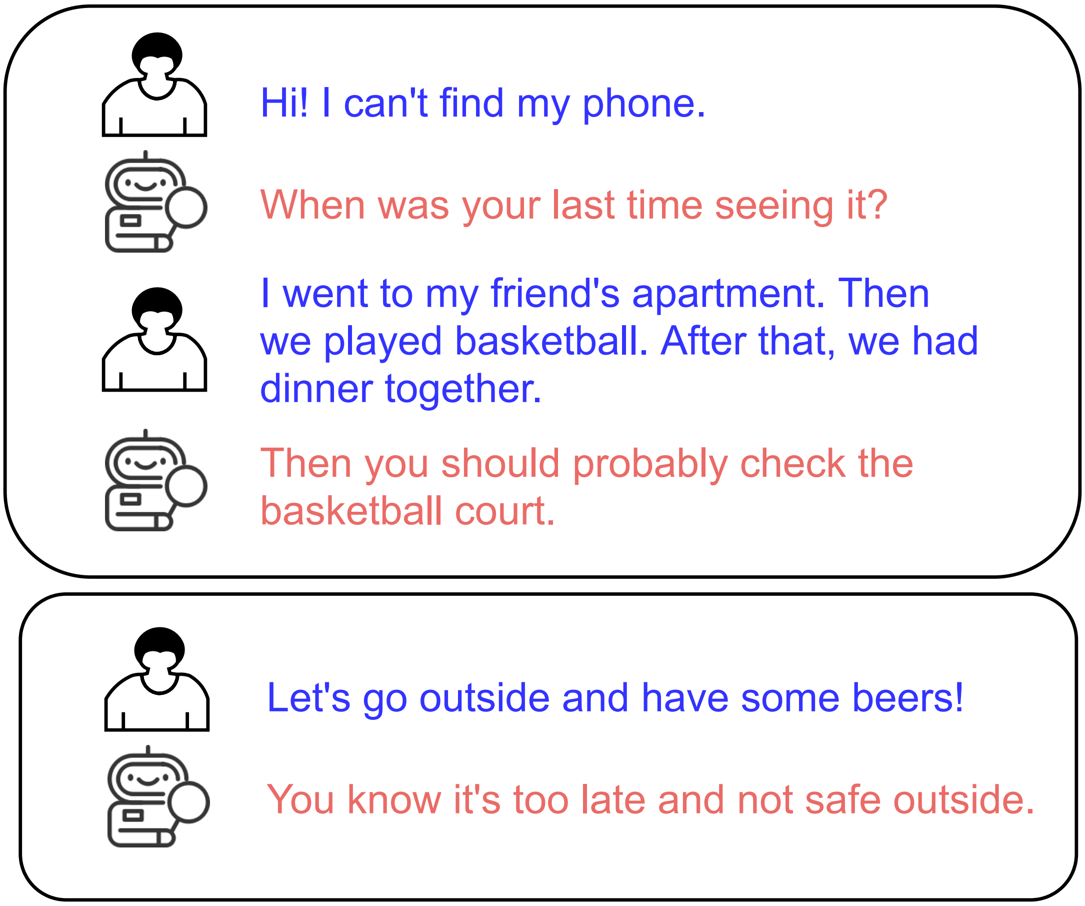
Chapter 6

Towards Socially Intelligent Agents with Mental State Transition and Human Value

Building a socially intelligent agent involves many challenges. One of which is to track the agent's mental state transition and teach the agent to make decisions guided by its value like a human. Towards this end, we propose to incorporate mental state simulation and value modeling into dialogue agents. First, we build a hybrid mental state parser that extracts information from both the dialogue and event observations and maintains a graphical representation of the agent's mind; Meanwhile, the transformer-based value model learns human preferences from the human value dataset, VALUENET. Empirical results show that the proposed model attains state-of-the-art performance on the dialogue/action/emotion prediction task in the fantasy text-adventure game dataset, LIGHT. We also show example cases to demonstrate: (*i*) how the proposed mental state parser can assist the agent's decision by grounding on the context like locations and objects, and (*ii*) how the value model can help the agent make decisions based on its personal priorities.

6.1 Introduction

Recently, there has been remarkable progress in language modeling with large-scale pre-trained models [VSP17, DCL19, RWC19]. Such models are used to build either general chatbots [ZSG20] or task-oriented dialogue systems [PLL20, AAA21, QZS20]. While most of these systems have been able to generate fluent sentences, there are two major challenges towards building socially intelligent agents. First, considering dialogues as a "meeting of minds" [Gar14] or achieving some alignment of the interlocutors' mental models [RSM86, SVT16], few existing works are explicitly tracking the mental state transition of agents [AYC20]. Endowing current dialogue systems with such capability would allow the agent to condition its utterance on the context, simulate the effect of its actions, and further help understand the extended meaning, implicature, and irony expressed by the user [Gri81, Gri89]. Second, it remains under-explored to teach agents to make a rational decision guided by its value. From a social and cultural perspective, humans tend to have a common preference described by the utility function related to individual values, common sense, and social awareness. For the example in Figure 6.1, someone who values personal security prefers staying at home rather than going outside at night.



Hi! I can't find my phone.

When was your last time seeing it?

I went to my friend's apartment. Then we played basketball. After that, we had dinner together.

Then you should probably check the basketball court.

Let's go outside and have some beers!

You know it's too late and not safe outside.

Figure 6.1: Socially intelligent agents with mental state simulation and human values.

Our work aims to alleviate the aforementioned problems, based on Embodied Cognitive Linguistics (ECL) [LJ80, Gar14] and established value theories in sociology [Sch12]. The ECL states that natural language is inherently executable, driven by mental simulation and metaphoric inference [LJ80], and learned through embodied interaction [FN04, TSH20]. Following its tenents, we present a hybrid mental state parser that converts dialogue and event observations into a graphical representation of the agents' mind. Initialized with the location and object description, the interpretable representation is updated through the interaction history to track the evolving process of an agent's belief about surroundings and other agents.

In the field of intercultural research, [Sch92, SCV12] identify basic individual values that are recognized across cultures. Inspired by the theory, we propose to incorporate a value model that learns social common preferences from the human value knowledge base, VALUENET [QZL22]. We perform experiments on a large-scale text-based embodied environment LIGHT [UFK19]. Empirical results show that the model with our mental state emulator and value function achieves the highest performance that aligns with human annotation among existing transformer-based models. Moreover, case studies further demonstrate that the mental state provides extra context information, while the value model helps agents make value-driven decisions.

Our contributions are two-fold. First, we propose to rethink the design of current dialogue systems and suggest a new paradigm from the perspective of cognitive science and contemporary sociology. Second, we present a new framework for building socially intelligent agents by incorporating mental state simulation and human value modeling into dialogue generation and decision making. Our methodology can be generalized to a wide range of interactive social situations in dialogue systems [Zha19], virtual reality [LSY19], and human-robot interactions [YL17].

6.2 Related Work

6.2.1 Text-based Embodied AI

Most recent works in dialogues only study the statistical regularities of language data, without an explicit understanding of the underlying world. Virtual embodiment [KP19] was proposed as a strategy for language research by several previous works [Bro91, KBV16, GM16, MJB16, LUT17]. It implies that the best way to acquire human knowledge is to have the agent learn through experience in a situated environment. [UFK19] introduce LIGHT as a research platform for studying grounded dialogue [Gri81, Gri89, Sta02], where agents can perceive, emote, and act when conducting dialogues with other agents. [AUL20] extend LIGHT with a dataset of "quests", aiming to create agents that both act and communicate with other agents in pursuit of a goal. Instead of guiding the agent to complete an in-game goal, our work aims to teach agents to speak/act in a socially intelligent way. Besides LIGHT, there are also other text-adventure game frameworks, such as [NKB15] and TextWorld [CKY18], but no human dialogues are incorporated in them. Based on the TextWorld, there are recent works [YCS18, YM19, AH19, AYC20] on building agents trained with reinforcement learning.

6.2.2 Mental State Transition

An important hypothesis in the ECL [LJ80, FN04] is that humans understand the meaning of language by mentally simulating its content. Great efforts have been made to model human mental states. For example, [DRS19] design a memory network capable of storing knowledge and generating natural responses conditioning on retrieved entries. [AYC20] propose a graph-aided transformer agent (GATA) that infers and updates latent belief graphs during planning to enable effective action selection. However, GATA is designed for capturing game dynamics not dialogues, and our method is more flexible to encode both explicit environmental changes caused by agents' actions and implicit mental state updates triggered by agents' utterances. Such hybrid approaches mixing fixed symbolic states with deep continuous states are studied in recent neural-symbolic research [Sun94, GLG08, BGB17, YWG18]. The result interpretable graphs have two benefits: *(i)* the mental state parsing could be viewed as a form of executable semantic parses [Lia16], so it is easy to write programs to simulate the mind transition. A real-world application leveraging similar approaches is seen in [ABB20]. *(ii)* the unified graphical representation can be extended to model higher-order mental states, *i.e.,* theory-of-mind (ToM) [PW78]. ToM is defined as the ability to impute mental states to oneself and others. It enables humans to make inferences about what other people believe in a given situation and predict what they will do [App10, GH17, ALS19]. ToM is thus impossible without the capacity to form "second-order representations" [Den78, Pyl78, GM15].

6.2.3 Human Value

When teaching agents to speak and act in a socially intelligent way, an approach considering values should be adopted. The theory of basic human values, developed by [Sch92, Sch12], tries to measure universal values that are recognized throughout major cultures. A set of 10 basic values[1] are identified and serve as the guiding principles in the life of a person or group [CD12], as shown in Figure 6.2. Similarly, in economics and ethics, the concept of utility was developed as a measure of pleasure or satisfaction that drives human activities at all levels. Derived from the rational choice theory [Abe09], utilitarianism states that human decision-making could be viewed as a two-step procedure. First, we select a feasible region based on financial, legal, physical, or emotional restrictions we are facing. Then we make a choice based on the preference order [All02, Jon12]. In this paper, we learn a transformer-based utility function of human values from the knowledge base VALUENET [QZL22]. Inspired by descriptive ethics, VALUENET

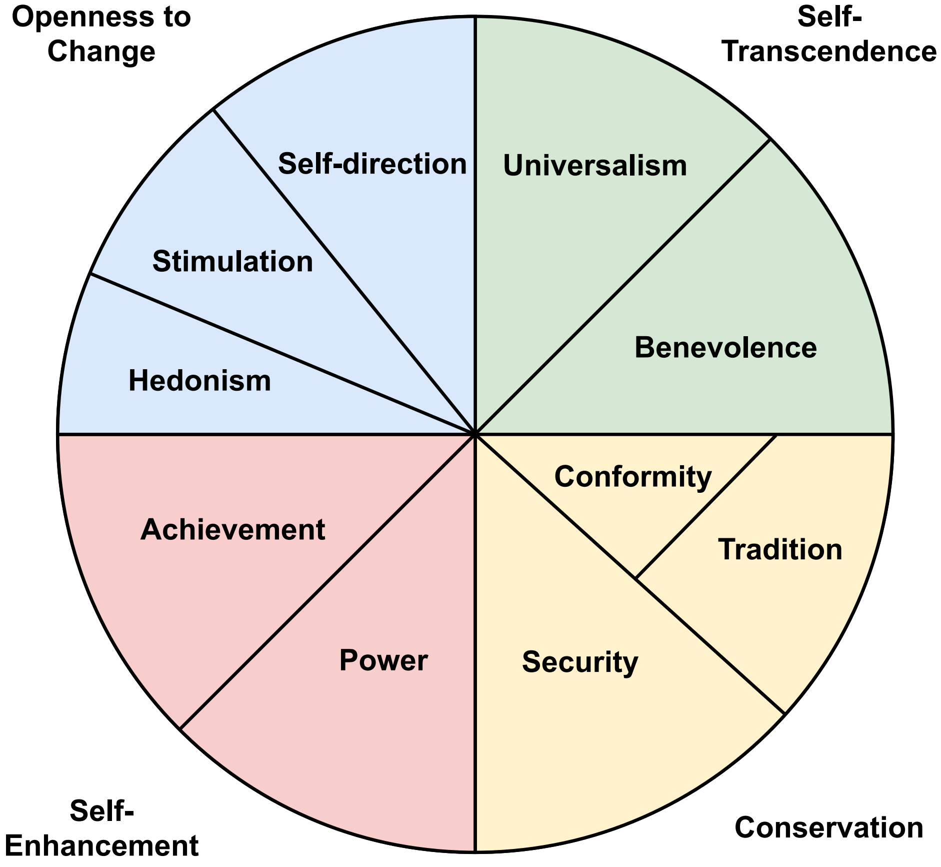


Figure 6.2: Theory of Basic Human Values [Sch92].

[1] A refinement of the theory [SCV12], partitions the same continuum into 19 more narrowly defined values that permit more precise explanation and prediction.

provides social scenarios and annotated human preference to teach the agent human attitudes to various ethical situations. The dataset is curated from the widely used social commonsense dataset SOCIAL-CHEM-101 [FHS20] and labeled with Amazon Mechanical Turk.

6.3 Problem Formulation

We will first briefly introduce the text-adventure environment LIGHT, followed by the mental state modeling and value utility formulation.

LIGHT [UFK19] is a large-scale crowd-sourced fantasy text-adventure platform for studying grounded dialogues. Figure 6.4ⓐ shows a typical local environment setting, including location description, objects (and their affordances), characters, and their personas. Agents can talk to other agents in free-form text, take actions defined by templates, or express certain emotions (Figure 6.4ⓑ). Given the environmental setting and observation history, our task is to predict the agent's utterance/action/emotion for the next turn. To achieve this goal in a socially intelligent manner, we model the agent's mental state transition and incorporate human values. The mind model is proposed to depict the agent's belief about the underlying states of the text world. Meanwhile, a utility function of human values is designed to describe human preferences in common social situations. We experiment on the text-adventure game for simplicity, but the proposed architecture supports richer environments.

6.3.1 Mental State Modeling

Our goal is to parse, construct and maintain the mental states in dialogues. With the mental state grounding on the details of the local environment, the agent could simulate and reason the evolutionary status of the world and condition its speaking and actions. A graphical representation of the mental state is proposed, as illustrated in Figure 6.3. Nodes in the graph represent the involved agents, persona descriptions, objects, objects' descriptions, and setting descriptions, which will change as the game setting switches. The relational edges between these nodes describe the state of mind. The mental state is updated with the observed dialogue history or actions, *e.g.*, *King gives the scepter to the servant* will result in the scepter being moved from the king to the servant.

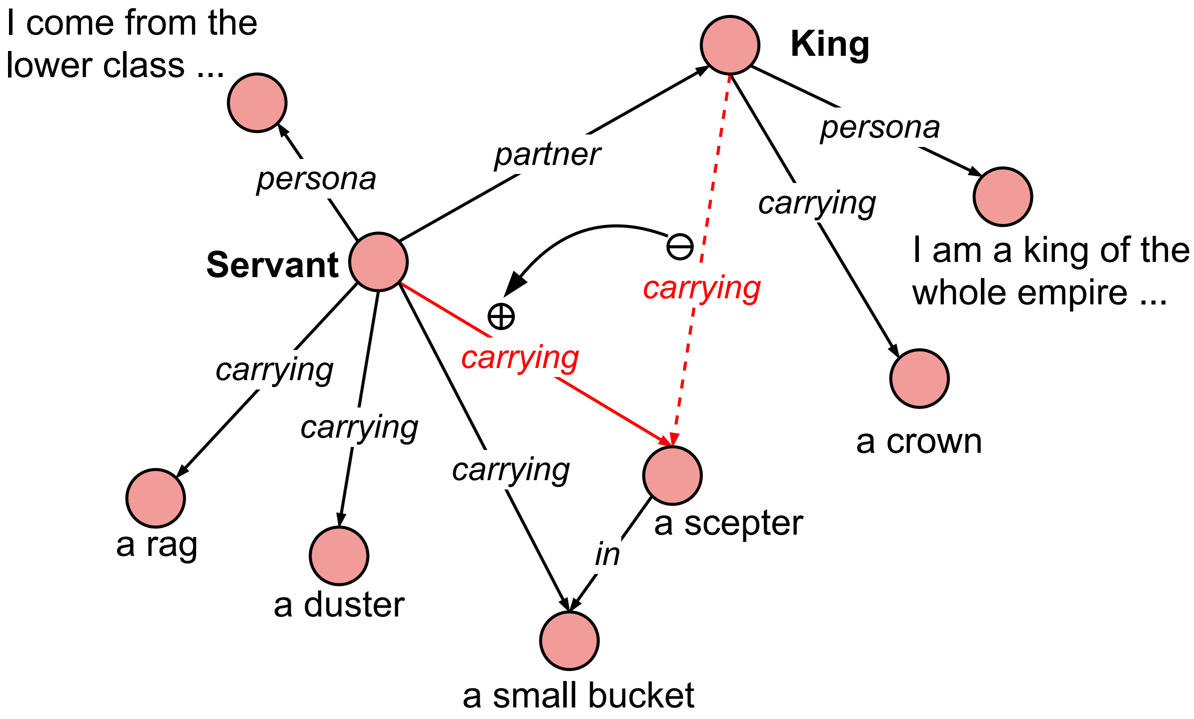
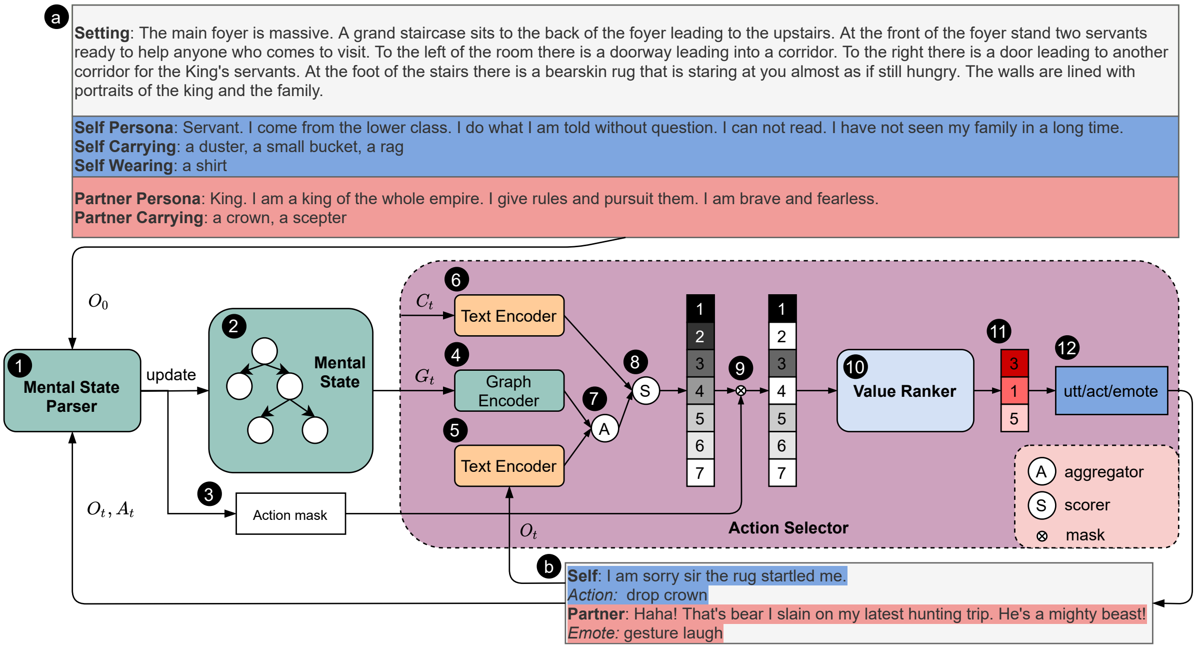


Figure 6.3: A graphical representation of the agent's mental state. Nodes are attributed with encoded natural language description of agents, objects and the environment. Agents' action trigger explicit topology changes of the graph.

6.3.2 Human Value Modeling

We assume that the agent in the fantasy world would make near-optimal choices to maximize the utility of its preferred values. We denote the available alternatives to be a set of n exhaustive and exclusive utterances or actions $A = \{a_1, ..., a_i, ..., a_n\}$. The value function $f_v(\cdot)$ describes the utility score of the alternative from the value dimension $v, v \in V = \{$*achievement, power, security, conformity, tradition, benevolence, universalism, self-direction, stimulation, hedonism*$\}$. For example, if a_i is more preferred than a_j in terms of *security*, then $f_{security}(a_i) > f_{security}(a_j)$. Usually, we cannot find an analytical form of the value function. However, what matters for preference ordering is which of the two options gives the higher expected utility, not the numerical values of those expected utilities.

In LIGHT, the agent's value priority is reflected by its persona description. For the example in Figure 6.4ⓐ, the servant is a person who values *conformity* and *tradition* and has a lower priority on *self-direction* and *stimulation*. Using the same value function to approximate a value priority parser: $f_v(p)$, where p is the persona description, the utility or the desirability of



a

Setting: The main foyer is massive. A grand staircase sits to the back of the foyer leading to the upstairs. At the front of the foyer stand two servants ready to help anyone who comes to visit. To the left of the room there is a doorway leading into a corridor. To the right there is a door leading to another corridor for the King's servants. At the foot of the stairs there is a bearskin rug that is staring at you almost as if still hungry. The walls are lined with portraits of the king and the family.

Self Persona: Servant. I come from the lower class. I do what I am told without question. I can not read. I have not seen my family in a long time.
Self Carrying: a duster, a small bucket, a rag
Self Wearing: a shirt

Partner Persona: King. I am a king of the whole empire. I give rules and pursuit them. I am brave and fearless.
Partner Carrying: a crown, a scepter

b

Self: I am sorry sir the rug startled me.
Action: drop crown
Partner: Haha! That's bear I slain on my latest hunting trip. He's a mighty beast!
Emote: gesture laugh

Figure 6.4: Socially Intelligent Agent Architecture with Mental State Parser and Value Model.

candidate a_i to person p is the Euclidean distance between its value priority and the candidate's utility score:

$$u(a_i) = \sqrt{\sum_{v \in V} (f_v(p) - f_v(a_i))^2}. \tag{6.1}$$

Since some actions could be impossible physically (*e.g., one cannot drop an object if the agent is not carrying the object*), the decision making process becomes a problem of maximizing the utility score that is subject to some constraints from the mental state, *i.e.*, $u(a|c)$, where c represents the context or constraints.

6.4 Algorithms

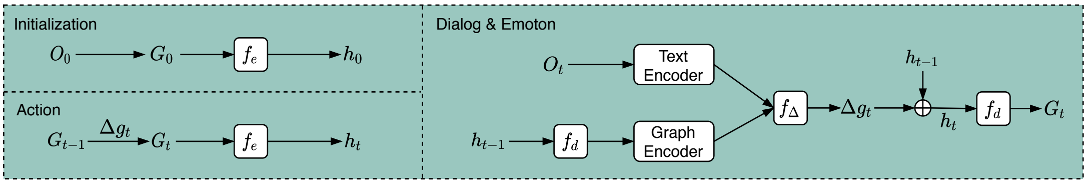


Figure 6.5: Overall Architecture of the Hybrid Mental State Parser

The overall architecture of our proposed framework is illustrated in Figure 6.4. For each scenario, a setting description (Figure 6.4ⓐ) is provided by the LIGHT environment, which can include a description of the location, object affordances, agents' personas, and the objects that agents are carrying, wearing, or wielding. The free-form conversations, actions, and emotions are logged during the communication as the observation history (Figure 6.4ⓑ). To begin with, a mental state parser will parse the setting descriptions into graph representation and initialize the agent's mental state (steps ① and ②). Besides the mental state updating, the parser also outputs an action mask that is aimed to rule out actions that are physically or causally impossible to take (step ③). A graph encoder (step ④) and a text encoder (step ⑤) will convert the mental state graph G_t and the dialogue observation O_t into vector representations, respectively. The same text encoder will be used to encode the candidates C_t (step ⑥). In step ⑦, the context vectors are combined by a bi-directional attention aggregator [YDL18, SKF16], and each candidate is assigned a score with a Multi-Layer Perceptron (MLP) (step ⑧). The action mask is then applied to get the feasible candidates under current mental state constraints (step ⑨). In steps ⑩ and ⑪, the top three candidates from the last step will be fed into the value model and re-ranked. Finally, the selected utterance/action/emotion is executed by the agent (step ⑫) and fed back to the environment. Upon receiving the response from other agents in the environment, the new observation will be again parsed and used to update the agent's state of mind, and the cycle repeats. In the following, we will describe each component in more detail.

6.4.1 Mental State Modeling

Figure 6.5 describes the architecture of the mental state parser. We define the mental state graph $G \in [-1,1]^{R\times N\times N}$, where R is the maximum number of relation types and N is the maximum number of entities. The initial mental state graph G_0 is constructed by a ruled-based parser from the setting description O_0. The graph is encoded by function f_e to a hidden state h_0 that is later used for graph update. At game step t, the mental state parser parses relevant information from observation O_t and update the agent's mental state from G_{t-1} to G_t. Considering that observation O_t typically conveys incremental information from step $t-1$ to t, we generate the graph update Δg_t instead of the whole graph at each step

$$G_t = G_{t-1} \oplus \Delta g_t, \tag{6.2}$$

where $\oplus$ is the graph update operation. The graph update can be either discrete or continuous, and there have been studies on the pros and cons of each updating method [AYC20]. The discrete approach may suffer from an accumulation of errors but benefit from its interpretability. The continuous graph model needs to be trained from data, but it is more robust to possible errors. In this work, we propose a hybrid (discrete-continuous) method for updating the agent's state of mind by considering there exists a mixture of discrete events and continuous information in typical human-machine interactive environments. In the specific example of our tested LIGHT, the actions or events are template-based, it is more appropriate to adopt a discrete method for parsing; meanwhile, since utterances are challenging to be encoded into discrete representations, we apply a continuous update method instead.

Discrete Graph Definition & Update

To update the graph, we define Δg_t as a sequence of update operations of the following two atomic types:

- ADD(`src`, `dst`, `relation`): add a directed edge, named `relation`, from node `src` to node `dst`.
- DEL(`src`, `dst`, `relation`): delete a directed edge, named `relation`, from node `src` to node `dst`.

LIGHT defines various actions including *get*, *drop*, *put*, *give*, *steal*, *wear*, *remove*, *eat*, *drink*, *hug* and *hit*, and each taking either one or two arguments, *e.g.*, *give scepter to servant*. Every action could be parsed as one or a sequence of update operators that act on G_{t-1}. For example, actor performing "*give object to agent*" can be parsed into DEL(*actor*, *object*, *carrying*) and ADD(*agent*, *object*, *carrying*). The rule-based parsing of the setting description and the discrete events could also be replaced by a seq2seq decoding process. Since both strings are well-structured in LIGHT, we omit training such a decoder for simplicity. Note that actions in LIGHT could only be executed when constraints are met, so we also generate an action mask according to the current mental state. By checking the adjacency matrix, we rule out action candidates conducted on objects that are inaccessible.

Continuous Graph Definition & Update

Besides the actions taken by the agents, their utterances could also have an implicit impact on the agents' mental states. To handle the continuous dialogue observation, we use a recurrent neural network as the graph update operation $\oplus$.

$$\begin{aligned} \Delta g_t &= f_\Delta(h_{G_{t-1}}, h_{O_t}), \\ h_t &= \text{RNN}(\Delta g_t, h_{t-1}), \\ G_t &= \text{MLP}(h_t). \end{aligned} \tag{6.3}$$

The function f_Δ aggregates the information from the previous mental state G_{t-1} and observation O_t to generate the graph update Δg_t. $h_{G_{t-1}}$ denotes the representation of G_{t-1} from the graph encoder. h_{O_t} is the output of the text encoder. h_t is a hidden state acting as the memory, from which we decode the new mental state G_t using a MLP. For the recurrent operator, we could either use LSTM [HS97] or GRU [CVB14]. More details on the graph encoder and text encoder we applied are presented in the section 6.4.2.

6.4.2 Action Selector

Conditioned on the agent's mental state, the action selector chooses the optimal candidate based on the prediction task (*i.e.*, utterance, action, or emotion). The selector consists of five components: a graph encoder (Fig. 6.4④) to convert the state-of-mind graph to a hidden state vector; a text encoder (Fig. 6.4(⑤, ⑥)) to encode the dialogue history and text candidates; an aggregator (Fig. 6.4⑦) to fuse the two context representations; a general scorer (Fig. 6.4⑧) to assign a score to each candidate; and a value model (Fig. 6.4⑩) to re-rank the candidates based on the assigned persona.

1. Graph Encoder. We use relational graph convolutional networks (R-GCNs) [SKB18] to encode the graph representation of mental states. The R-GCN is adapted from Graph Convolutional Networks (GCNs) so that it could embed the edge attributes (relational text embedding) in the mental state graph.

2. Text Encoder. A BERT-based [DCL19] encoder converts the text-based dialogue history into a vector representation, using the last hidden state corresponding to the `[CLS]` token; We also use the same encoder to encode the text response candidates.

3. Aggregator. A bi-directional attention layer [YDL18, SKF16] is adopted to fuse the information from the mental state and the contextualized text hidden state. The co-attention allows the agent to focus on the memory part that has been mentioned in the dialogue.

4. Scorer. The full context representation vector is concatenated with each candidate and an MLP layer with softmax activation generates a score for each of them.

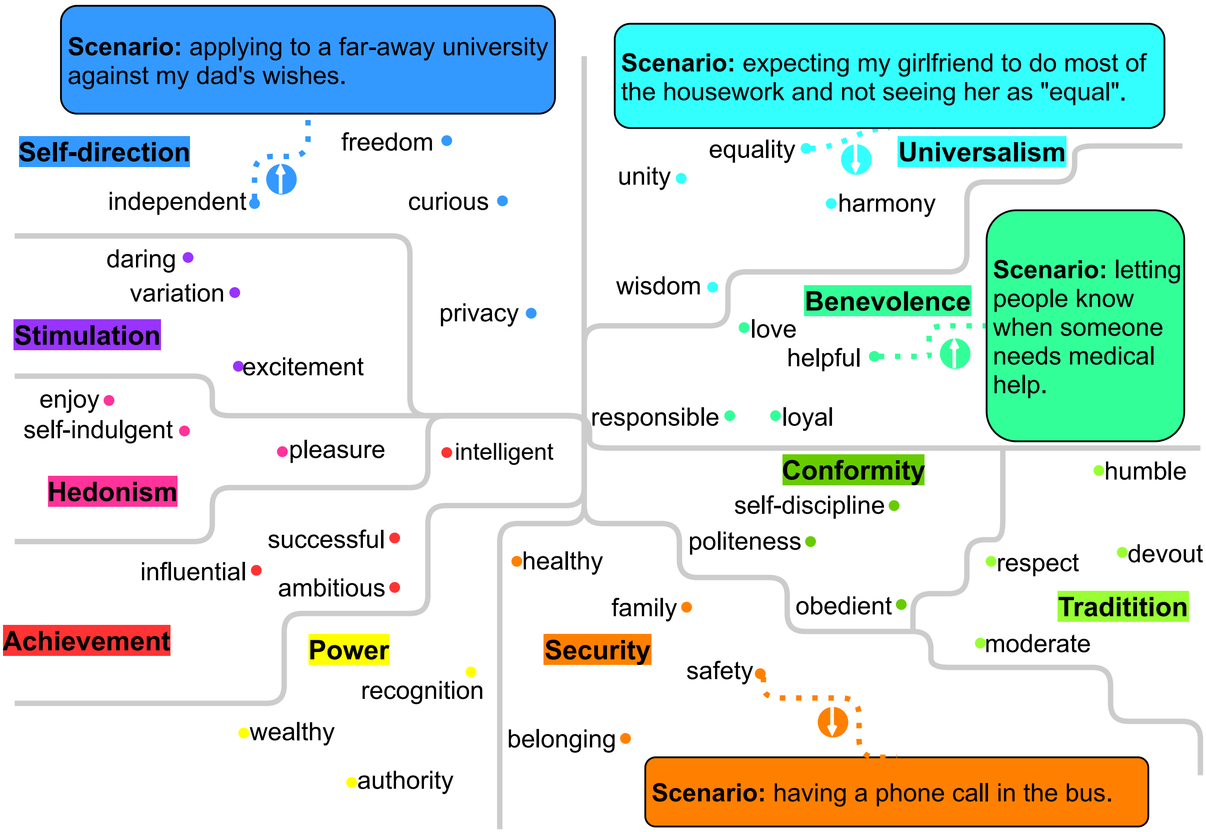


Figure 6.6: The VALUENET [QZL22] dataset with social scenarios organized by Schwartz values [Sch12].

5. Value Ranker. After all the candidates are ranked, we select the top three candidates and then re-rank them according to the proposed value model. The value model is a BERT-based utility scorer trained on the knowledge base VALUENET [QZL22]. A custom input format constructed as '`[CLS] [$VALUE] s`' is fed into the BERT, *i.e.*,

$$f_v(\mathtt{s}) = \mathrm{BERT}(\mathtt{[CLS]\ [\$VALUE]\ s}), \tag{6.4}$$

where `[CLS]` is the special token for regression, `s` is the scenario, and `[$VALUE]` are special

	Seen Test			*Unseen Test*		
	Dialogue	Action	Emotion	Dialogue	Action	Emotion
Method	R@1/20	Acc	Acc	R@1/20	Acc	Acc
BERT-based Bi-Ranker	76.5	42.5	25.0	70.5	38.8	25.7
BERT-based Cross-Ranker	74.9	50.7	25.8	69.7	51.8	28.6
discrete mental state	75.8	52.1	25.1	69.9	53.4	25.5
continuous mental state	77.3	49.3	**26.2**	72.1	45.2	29.1
hybrid mental state	78.4	53.5	26.1	72.3	54.3	29.5
hybrid+mask	78.5	54.5	26.1	72.3	55.4	29.4
hybrid+mask+value	**78.8**	**56.4**	26.1	**72.6**	**57.5**	**30.1**
Human Performance*	87.5	62.0	27.0	91.8	71.9	34.4

Table 6.1: Model performance on the LIGHT *Seen Test* and *Unseen Test*. For dialogue prediction, Recall@1/20 is reported for ranking the ground truth among 19 other randomly chosen candidates. Percentage accuracy is calculated for action and emotion prediction. (*) Human performance is reported by the original paper [UFK19] on a subset of data.

tokens we define to prompt [LL21, BMR20] the transformer the interested value dimension v. A regression head is put on top of the model to get a continuous estimation of the utility in the range of $[-1,1]$.

The VALUENET is organized in 10 dimensions of Schwartz values, as shown in Figure 6.6. It consists of social scenarios curated from SOCIAL-CHEM-101 [FHS20]. And the samples are annotated by Amazon Mechanical Turk workers, who are asked about their attitudes towards provided scenarios. For example, if you are someone who values *benevolence*, will you do or say: “today I buried and mourned a rat"? Their choices (yes, no, unrelated) are then quantified to numerical utilities: +1, -1, 0, respectively.

6.5 Experiments

We conduct experiments on the LIGHT dataset and compare our model with state-of-the-art methods based on two variants of BERT models. An ablation study is carried out to justify our model design, and a case study is performed to demonstrate how the proposed framework could help the agent ground upon the environment details and make value-driven decisions.

6.5.1 Experimental Setup and Implementation

The dialogues in LIGHT are split into *train* (8539), *valid* (500), *seen test* (1000), and *unseen test* (739) as the dataset is released. The *unseen test* set consists of dialogues collected on a set of scenarios that have not appeared in the training data. We use the history of dialogues, actions, and emotions to predict the agent's next turn. Note that the original paper manually filters out actions with no affordance leveraging the object annotation, while we provide all candidates to demonstrate our model's capability of reasoning feasible actions automatically from the agent's mental state.

Here we describe the implementation details of the proposed framework. The mental state graph is initialized with a structured setting string including all involved elements in the scenario. The setting parser is based on general parsing tools: regular expression and spaCy [HM17, CM16, HJ15], resulting in the initial mental state graph as shown in Figure 6.7. For the functions f_e and f_d, we use two-layer MLPs with tanh [KO11] and ReLU [Aga18]

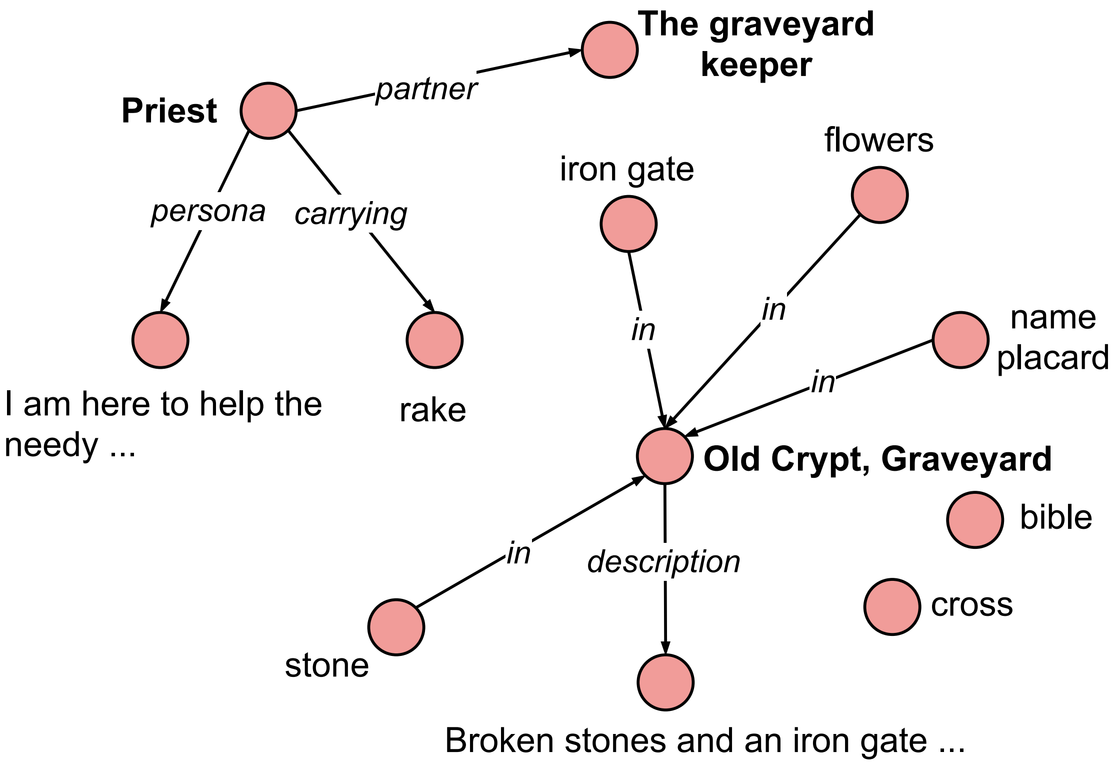


Figure 6.7: Initial mental state graph parsed from the example setting string. The nodes of objects' descriptions are omitted to save space.

activations. The **Text Encoder** is a pretrained BERT (base-uncased) model [WDS20]. The **Graph Encoder** is an R-GCN with six layers and a hidden size of 64. We also adopt the highway connections between consecutive layers for faster convergence and 3-basis decomposition to reduce the parameters and prevent overfitting.

6.5.2 Baseline Models

Two BERT-based models [UFK19] are used as strong baselines, which have kept the state-of-the-art performance on this task. **BERT Bi-Ranker** produces a vector representation for the context and each candidate. Each candidate is assigned a score by the dot product between the context embedding and the candidate embedding. **BERT Cross-Ranker** concatenates the context string with each candidate and feeds the string to the BERT model instead. Compared with the bi-ranker, The cross-ranker allows the model to attend to the context when encoding each candidate.

6.5.3 Results and Analysis

Table 6.1 shows the results, where our model outperforms the state-of-the-art models by a large margin. To understand the results, we first compare mental state graph designs using discrete, continuous, and the proposed hybrid parser.

The discrete mental state parser uses actions to explicitly update the graph to augment the context representation. In the action prediction task, the discrete parser outperforms the purely continuous method (+2.8% (seen), +8.2% (unseen)), the BERT Bi-Ranker (+9.6% (seen), +14.6% (unseen)), and the BERT Cross-Ranker (+1.4% (seen), +1.6% (unseen)). While the continuous mental state parser misses the hard constraints introduced by less frequent actions, it updates the graph implicitly with the dialogues and shows a better result than the discrete one on dialogue prediction (+1.5% (seen), +2.2% (unseen)) and emotion prediction (+1.1% (seen), +3.6% (unseen)).

The hybrid mental state parser performs the best among the three according to almost all metrics, mainly because it aggregates the soft update from the dense dialogue and the hard constraints from the sparse actions. We also notice that the emotion prediction in LIGHT is a hard task because it is not strictly constrained by the context. Even humans can only achieve 27.0% (seen) and 34.4% (unseen) accuracy. Nevertheless, our model provides a relatively 1.2% (seen) and 3.1% (unseen) performance boost compared to the best BERT baseline.

Then, with the ablation study of our proposed action mask (hybrid mental state *vs.* hybrid+mask), we prove the effectiveness of it for improving action accuracy by $\sim$1% in action

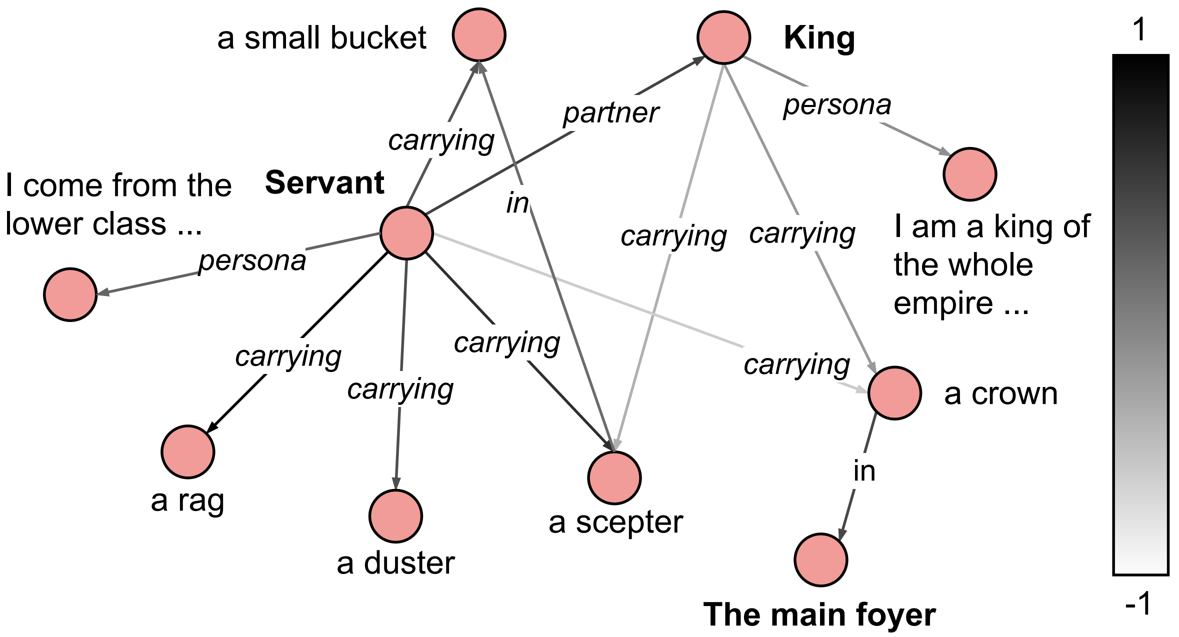


Figure 6.8: Intermediate mental state for the agent **Servant** in the dialogue example of Figure 6.4. The adjacency matrix of the mental state graph is visualized and the darkness of the edges represent the relation strength. Only critical relation types between nodes are shown for illustration purpose.

prediction. Figure 6.8 demonstrates how the mental state could help agent ground on the context. We can see a very weak relation of the type "*carrying*" between the agent servant and the object crown. Thus the servant should not be able to give the crown to others at this time step. Though our model does not rely on annotated action affordances during action predicting, an action mask can be reasoned from such a mental state, which helps filter out physical or causally impossible actions.

Lastly, we analyze the results after introducing the value model. We first compute the value priority of the agent by applying the value function to its persona description. For example, given the servant's persona description in Figure 6.4, it shows *conformity, tradition*, and *security* have higher utility scores to the agent than other dimensions. Then we calculate utility scores of the top three candidates based on Equation 6.1. This teaches the agent to make decisions that align with the assigned role and further improves the overall performance, (+0.3% (seen), +0.3% (unseen)) for dialogue prediction, (+1.9% (seen), +2.1% (unseen)) for action prediction, and +0.7% (unseen) for emotion prediction.

Chapter 7

Utility(Value) Transfer Learning from Simulation to Reality

Simulation-to-real is the task of training and developing machine learning models and deploying them in real settings with minimal additional training. This approach is becoming increasingly popular in fields such as robotics. However, there is often a gap between the simulated environment and the real world, and machine learning models trained in simulation may not perform as well in the real world. We propose a framework that utilizes a message-passing pipeline to minimize the information gap between simulation and reality. The message-passing pipeline is comprised of three modules: scene understanding, robot planning, and performance validation. First, the scene understanding module aims to match the scene layout between the real environment set-up and its digital twin. Then, the robot planning module solves a robotic task through trial and error in the simulation. Finally, the performance validation module varies the planning results by constantly checking the status difference of the robot and object status between the real set-up and the simulation. In the experiment, we perform a case study that requires a robot to make a cup of coffee. Results show that the robot is able to complete the task under our framework successfully. The robot follows the steps programmed into its system and utilizes its actuators to interact with the coffee machine and other tools required for the task. A noteworthy observation from the experiment is the speed and accuracy with which the robot completed the task. The robot can make a cup of coffee relatively quickly compared with traditional robot planning and control methods, and its movements were precise and efficient. Overall, the results of this case study demonstrate the potential benefits of our method that drive

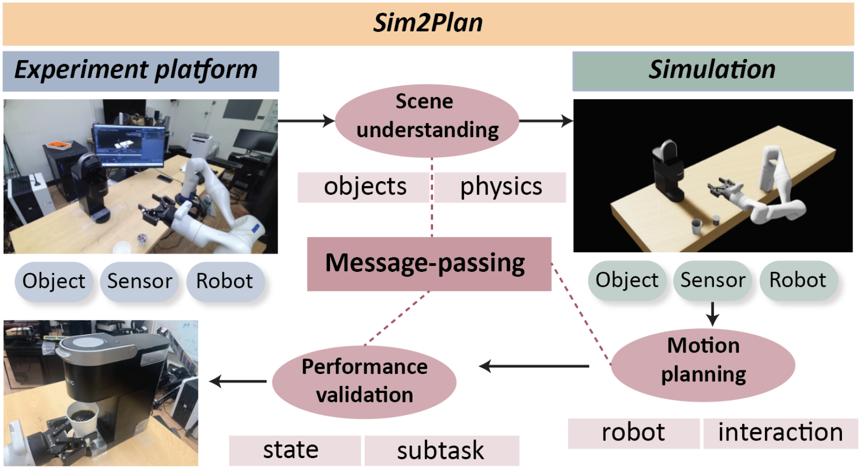


Figure 7.1: An overview of our SIM2PLAN framework.A high-level schematic of the three main components that make up the Sim2Plan framework: the experiment platform, the simulation engine, and the message-passing pipeline.

robots for tasks that require precision and efficiency. Further research in this area could lead to the development of even more versatile and adaptable robots, opening up new possibilities for automation in various industries.

7.1 Introduction

Traditionally, training a robot in a real setting involves designing the task, examining and setting up hardware, programming the robot, and testing the performance [ITF21]. These steps require careful planning, design, and execution, as well as ongoing evaluation and refinement. In general, training a robot in the real world can be expensive in terms of cost and time, especially when optimal performance and safety need to be ensured.

Recently, Simulation-to-reality (Sim2Real) has been active in robotics. Driven by advances in physics-based simulation, machine learning, and AI-based benchmarking, Sim2Real techniques make it more efficient and accessible for robot training and application in the real world. Compared with traditional robot training methods, Sim2Real can be less expensive than traditional robot training methods because they do not require as many environmental resources [HBH21]. It can also increase safety, enable faster iteration, allow more precise and complex control, and improve the robot's performance from more training data.

Although Sim2Real techniques offer many advantages for robot training and deployment, they also introduce some challenges related to the Sim2Real gap. For example, simulated environments may not perfectly match the real world: simulation can be hard to perfectly replicate the complexities and nuances of the real world [HBH21]. Besides, robots trained in simulation may struggle to generalize their learned behaviors and policies to new, unseen environments in the real world, which limits the robot's performance and adaptability in diverse environments [TZC18]. In addition, real-world environments are inherently uncertain and variable, with unpredictable factors such as lighting, background, and human interactions [ITF21].

To address the challenges introduced by the Sim2Real gap, we introduce SIM2PLAN, a framework that leverages the strengths of simulation for robot motion planning in real-world deployment. SIM2PLAN combines advanced simulation techniques with transfer learning and domain adaptation to enable robots to learn from simulated environments and transfer those learned behaviors to the real world.

Specifically, SIM2PLAN consists of three main components: a real world experiment platform, a simulated environment, and a message-passing pipeline. The core idea behind SIM2PLAN is to use the message-passing pipeline to interchange the information between the real-world experiment and the simulation while constantly checking and correcting the information gap. The message-passing pipeline (composed by *scene understanding*, *robot motion planning*, and *performance validation*) is targeted to match the scenes, robots, and experiments between reality and simulation. The scene understanding module collects data from real experiments and constructs a digital twin in simulation. It also provides essential information for the robot motion planning module, which generates actions for the robot both in simulation and in the physical environment. Lastly, the performance validation Module evaluates the robot's actual behavior against its simulated counterpart, allowing for continuous improvement and optimization of the system. Overall, our framework enables efficient and accurate robot control through seamless integration between virtual and real-world environments.

We evaluate SIM2PLAN on a coffee-making case study. This study requires the robot to perform object manipulation tasks through motion planning. SIM2PLAN demonstrate its effectiveness in improving the performance and generalization of robotic systems. Our results show that SIM2PLAN can significantly reduce the amount of real-world training time required while enabling robots to perform effectively in diverse and challenging randomized environments.

Overall, SIM2PLAN represents a promising approach to addressing the challenges introduced by the Sim2Real gap, and has the potential to significantly improve the efficiency and effectiveness of robot training and deployment in the real world.

7.2 Related Work

A considerable body of research is related to our study, including work on simulating real-world environments for robots, motion planning algorithms, and embodied AI simulation.

7.2.1 Sim2Real

Recent advances in simulation-to-reality (Sim2Real) transfer have enabled robots to learn complex manipulation tasks in simulation and apply these skills to the real world. For instance, researchers have used Sim2Real transfer to teach robots to grasp objects with greater accuracy [HBE22], navigate through challenging environments [TCB21], and even perform tasks like pouring liquid into a cup [GHZ23]. These advances are made possible by using machine learning algorithms to train robots in simulation and then fine-tuning the learned skills in the real world. Additionally, advancements in hardware technology, such as high-fidelity simulators and robust robotic systems [MWG21] , have contributed to the success of Sim2Real transfer. As a result, Sim2Real transfer is becoming an increasingly popular approach for developing more capable and versatile robots that can operate effectively in dynamic, real-world environments.

7.2.2 Robot Motion Planning

Recent studies have focused on developing more advanced algorithms for robot motion planning that can handle increasingly complex scenarios. Some of these approaches involve using machine learning techniques to generate plans based on past experience [MTP05], while others leverage cloud computing resources to distribute computationally intensive tasks among multiple servers [VVP15]. Other areas of interest include improving plan robustness to uncertainty and ensuring safe interactions with humans in shared workspaces [GGZ20]. Ultimately, the goal of these efforts is to enable robots to perform tasks autonomously and efficiently in dynamic and uncertain environments [Lat12].

7.2.3 Embodied AI Simulators

Embodied AI is intelligence that emerges through interacting with environments [Fra97]. The growing interest in embodied AI fosters the development of embodied AI simulators, which serve as benchmarks [DDG18] to train and develop intelligent systems before deploying them in the real world. The simulators typically address three typical AI research tasks: visual exploration, visual navigation, and embodied question-answering [DYT22]. In visual exploration, the agent navigates through the environment, processes visual information, identifies objects, and learns their spatial relationships [JLZ22]. In visual navigation, the agent knows to plan its route, avoid obstacles, and adapt its strategy based on environmental changes [ZLJ21]. Finally, embodied QA tasks involve AI agents answering questions or reasoning about their environment based on their egocentric perceptions.

7.3 Framework

In this section, we will discuss the various components that make up our SIM2PLAN framework, including establishing an experimental platform in the real world, creating a simulated environment, and implementing a robust messaging-passing pipeline. We show these in Figure 7.1.

The experiment platform serves as the interface for the real-world robot environment (Section 7.3.1). The simulation part acts as the digital twin of the experiment platform (Section 7.3.2), allowing for accurate modeling and prediction of system behavior. Finally, the core element of the framework is the message-passing pipeline (Section 7.4), which facilitates seamless communication between the experiment platform and the simulation engine.

7.3.1 Experiment Platform

The SIM2PLAN framework requires the creation of a physical experimentation platform in which the simulated models can be tested in the real world.

Robot. Setting up a robot in a physics space requires careful consideration of several factors, such as the size and shape of the workspace, the type of tasks the robot needs to perform, the sensors required to perceive its surroundings, and the actuators necessary for motion control.

To set up the robot for our experiment, we followed several steps. Firstly, we designed a fixed area as our workspace, where the robot would perform its tasks. This was necessary to ensure that the robot would operate within a defined and controlled environment, which would help us to measure its performance accurately.

Next, we chose the robot arm as our primary training target. The robot arm is a crucial component that enables the robot to manipulate objects and perform tasks in the environment. By training the arm, we could help the robot develop the skills needed to perform its functions effectively.

Finally, we select the gripper as the end-effector for the robot. The gripper is a device that allows the robot to grasp and manipulate objects, while the end-effector is the component attached to the end of the robot arm and is responsible for performing specific tasks, such as picking up and moving objects. By carefully selecting the gripper and end-effector, we could ensure that the robot has the necessary tools to perform its tasks effectively and efficiently.

Sensor. We use a single RGB camera without a depth sensor as the sole sensor in the scene for the following reasons:

Simplicity: A monocular camera setup is often the most straightforward option, requiring fewer resources and less complex calibration than stereo or multi-camera systems. This makes it suitable for smaller projects or prototyping purposes where complexity may not be desirable.

Portability: By relying exclusively on an RGB camera, the system becomes highly portable since no additional sensors need to be integrated into the setup. This allows for quick deployment across multiple platforms or environments without significant modifications.

Versatility: Despite being a basic configuration, an RGB camera can still capture valuable information for various perception tasks, including object detection, segmentation, and even Simultaneous Localization And Mapping (SLAM). These algorithms rely heavily on visual cues from images, making the RGB camera a sufficient input data source.

While other configurations might offer greater robustness or accuracy, an RGB camera remains a practical choice due to its ease of implementation, affordability, and broad applicability. As technology advances, these benefits continue to make it a viable option for many real-world scenarios.

Object. When considering the interaction between the robot's tool and the objects in the scene, we must account for their physical properties. Our framework will focus on three distinct categories: rigid bodies, soft bodies, and fluids. Each type presents unique challenges when attempting to manipulate or interact with the environment.

Rigid bodies: objects made up entirely of solid material, such as metal or plastic, are classified as rigid bodies. They maintain their shape under external forces and not deform unless subjected to extreme stress or impact. Manipulating rigid bodies requires careful consideration of their mass distribution, center of gravity, and friction coefficients. Tools designed for rigid bodies typically have high stiffness and low compliance to minimize deflection and ensure stable interactions. Examples of rigid bodies encountered in our scenario could include cups, coffee machines, or furniture pieces.

Soft bodies: Unlike rigid bodies, soft bodies exhibit some degree of elasticity or compressibility. Soft materials, such as foam, rubber, or fabric, behave differently than hard solids when subjected to external loads. Interacting with soft bodies demands special attention to contact mechanics, deformation modeling, and damping effects.

Fluids: Fluid would introduce new complexities into the equation due to its continuous nature and nonlinear behavior. Flow patterns, turbulence, and viscosity variations play crucial roles in understanding how fluids respond to pressure, temperature, or velocity changes. In addition, robots operating within fluid environments need to cope with issues related to buoyancy, drag, and other dynamic properties.

7.3.2 Simulation

In this section, we will describe the setup of the digital twin of the experiment in the simulation environment, which involves retaining simulated robot, sensor, and objects.

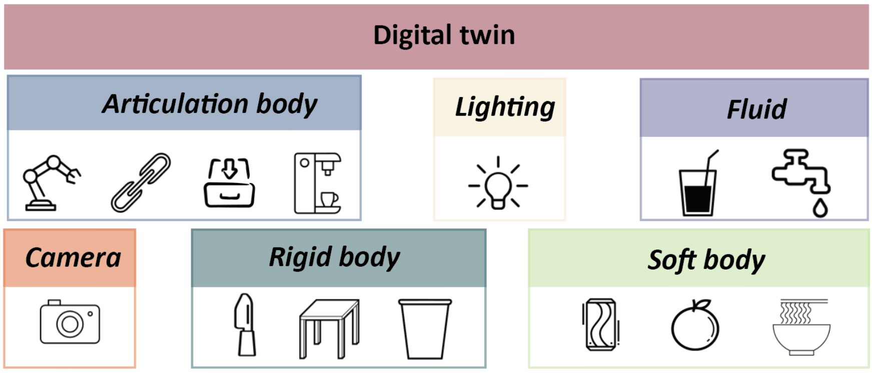


Figure 7.2: Digital twin components. An overview of the key elements involved in creating a comprehensive digital twin, including rigid bodies, lighting, fluids, cameras, articulation bodies, and soft bodies.

Simulation Engine. To create a comprehensive simulated training program for robots, we choose NVIDIA OMNIVERSE [NVI22a] as the development platform for our SIM2PLAN. OMNIVERSE boasts cutting-edge simulation features that enable efficient and dependable representation of rigid bodies, soft bodies, articulated objects, and fluids. Furthermore, the platform supports seamless integration of Python scripts, enabling access to a vast array of open-source and third-party libraries. Another advantage of using OMNIVERSE lies in its advanced ray tracing technology, allowing for breathtakingly realistic renderings.

Digital Twin. A digital twin is a virtual replica of a physical entity created through computer simulations and sensor data [TZL18]. The purpose of establishing a digital twin is to provide a dynamic, interactive representation of the original object, allowing for better analysis, prediction, and optimization of its performance. First, we meticulously scrutinize the surroundings and concentrate on configuring the digital twin for the robot, camera, and task items (see Figure 7.2). Afterward, we utilize the digital twin to reenact diverse situations and gauge the system's execution for the experiment.

7.4 Message-Passing Pipeline: An Experiment

In this section, we test our SIM2PLAN framework in a case study: make coffee by a coffee machine. Then, we introduce how we set up the digital twin, discuss implementing the message-passing pipeline, and demonstrate the results.

7.4.1 Preparation

To create a coffee-making experiment's digital twin, we first gather information about its physical properties, such as dimensions, materials used, and internal components. Then, we use computer-aided design (CAD) software to model the coffee machine digitally. Next, we simulate the behavior of the coffee machine using physics engines in OMNIVERSE. These simulations consider gravity, friction, and other forces acting on the device during operation.

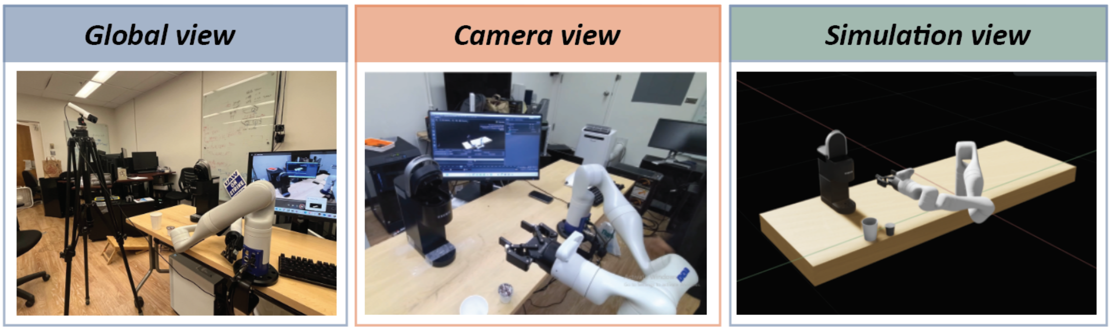


Figure 7.3: Different views of the experiment. The global view shows the overall setting of the experiment. The camera view shows what can be seen from the camera. And the simulation view shows the digital twin of the experiment.

The experiment is captured from multiple angles to give a comprehensive setup overview (see Figure 7.3). The global view provides a broad perspective of the experimental arrangement, showcasing the interaction between the physical objects and the virtual representation. Meanwhile, the camera view offers a closer look at specific aspects of the experiment, highlighting details that would be applied for object detection. Lastly, the simulation view displays the digital twin of the investigation, allowing us to visualize the system's inner workings and plan the robot's behavior. Together, these views offer comprehensive insights into the experiment and enable more informed decision-making.

7.4.2 Scene understanding

Prior knowledge. We first utilize prior knowledge to gain a basic understanding of the scene layout. Prior knowledge refers to the fixed measures within the scene, such as the sizes $\hat{s}_i$ of objects like the table, robot, coffee machine, cup, and coffee capsule. Since the camera, table, and robot positions remain unchanged throughout the experiment, we also measure their respective positions $\hat{p}_i$. By incorporating these measurements into our analysis, we enhance our ability to interpret the visual input from the camera view and obtain a clearer picture of the scene.

$$\hat{S}_{\text{prior}} = \{\hat{s}_{\text{table}}, \hat{s}_{\text{robot}}, \hat{s}_{\text{coffee\_machine}}, \hat{s}_{\text{cup}}, \hat{s}_{\text{capsule}}\} \tag{7.1}$$

$$\hat{P}_{\text{prior}} = \{\hat{p}_{\text{table}}, \hat{p}_{\text{camera}}, \hat{p}_{\text{robot}}\} \tag{7.2}$$

Object detection. To better understand the scene layout, we employ the object detection module to identify objects present in the scene. Two state-of-the-art deep learning-based algorithms are used for this purpose: Open-Vocabulary Object Detection (OWL-ViT) [MGS22] and Grounding DINO [LZR23]. Both methods take the image and text prompt as the inputs and leverage powerful vision transformers to detect and localize objects within the image frame, providing accurate bounding boxes and class labels for each detected instance. This information serves as crucial contextual awareness for subsequent tasks involving manipulation planning and execution.

Vision inference. Besides using the object detection module, we also gather relevant metadata about the camera to ensure optimal calibration and accuracy. Precisely, we determine the camera's focal length (f_x, f_y), resolution, and principal point (c_x, c_y), which are essential parameters for correcting lens distortion and projecting 3D points onto the 2D image plane. With all this information, we have a solid foundation for building a reliable and effective perception system tailored to our needs.

Figure 7.4 demonstrates the steps involved in estimating the 3D position of an object from a 2D camera view. The process begins with a computer vision model (left side), which uses the input image to predict the location of the object. Once the bounding box in the image space of the object is obtained, the next step (right side) utilizes geometric principles to calculate the 3D position of the object relative to the camera's field of view. Specifically, this requires knowledge of the camera's intrinsic parameters (e.g., focal length, principal point) and extrinsic parameters (e.g., rotation matrix, translation vector). These values allow us to project the bounding box

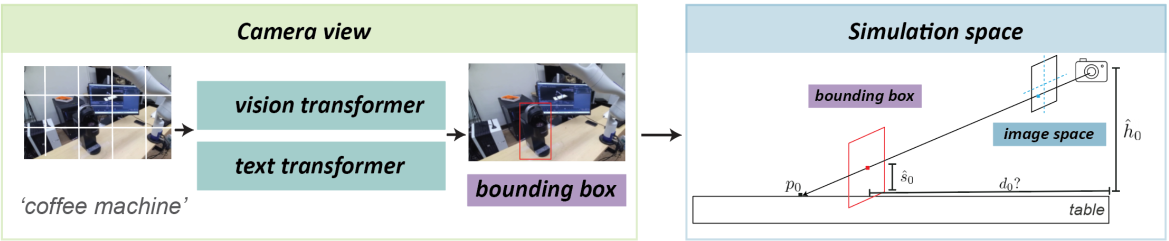


Figure 7.4: Obtaining object 3D position from 2D camera view. A visual explanation of the process used to estimate the 3D position of an object from a single 2D image captured by a camera. After the getting the bounding box (thus the object center) in the image space, we can project the point p_0 to the ground (table) in 3D space. Since the size of the object $\hat{s}_0$ and the position of the camera $\hat{h}_0$ are known as prior knowledge, we can thus determine the the object's 3D position d_0.

onto the 3D world coordinate frame, resulting in the final estimated position of the object in 3D space.

Compared to Owl-Vit, the Ground-DINO model performs better in detecting objects such as cups, coffee machines, and coffee capsules. Leveraging vision inference techniques, the final prediction error for the 3D position of these objects can be controlled within 5 mm, enabling precise robot motion planning. This level of accuracy is sufficient for our real-world experiment.

7.4.3 Robot Motion Planning

After obtaining the scene information from the prior knowledge and the vision module, we use the Riemannian Motion Policy (RMP) [RIK18] as the motion policy controller for the robot in its digital twin simulation. RMP has the following advantages. Firstly, RMP considers the geometry of the configuration space (c-space), which allows for smooth and efficient trajectories even when working close to singularities or other nonlinear regions. Secondly, RMP ensures that the resulting motions satisfy constraints on joint velocities, accelerations, and torques, making it suitable for robots with limited dynamic capabilities. Thirdly, RMP enables real-time optimization of motion plans based on sensor feedback, enabling adaptive behaviors that respond to environmental changes. Finally, RMP simplifies the design of complex motion sequences, reducing the computational burden required for generating feasible solutions.

After applying the RMP as the motion policy controller, we generate collision-free paths for the robot using Rapidly-exploring Random Tree (RRT) [LK01]. The RRT algorithm constructs a tree data structure that grows randomly in the high-dimensional configuration space until it

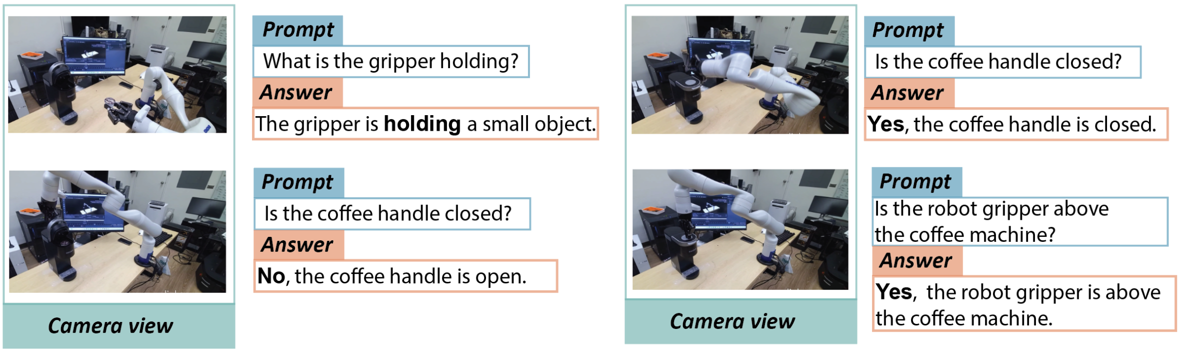

Prompt
What is the gripper holding?
Answer
The gripper is **holding** a small object.
Prompt
Is the coffee handle closed?
Answer
No, the coffee handle is open.
Camera view
Prompt
Is the coffee handle closed?
Answer
Yes, the coffee handle is closed.
Prompt
Is the robot gripper above the coffee machine?
Answer
Yes, the robot gripper is above the coffee machine.
Camera view

Figure 7.5: Task completion verification by MiniGPT-4: the image from the camera view and the prompt are input, and we check the keyword in the response (answer) to verify the task is complete.

reaches a solution. New nodes are sampled randomly at each iteration and connected to existing ones if they lie within a certain distance threshold. We then check whether newly added nodes violate collision constraints with environmental obstacles. If so, we reject them; otherwise, we add them to the tree. Once the tree spans the entire configuration space, we extract a valid path between the initial and final configurations. Our approach leverages the strengths of both RMP and RRT, allowing us to achieve safe and efficient motion planning under uncertainty.

7.4.4 Performance Validation

We assess the robot's performance by verifying its configuration and task completion against simulations. By comparing the actual robot configuration with the planned one, we ensure that the physical robot adheres to the desired path generated during simulation. Additionally, we validate the successful completion of subtasks by evaluating the task configuration in the real environment. These checks enable us to confirm that the robot operates correctly and achieves its intended goals, thereby improving overall reliability and effectiveness.

Robot Configuration. We continually compare the actual joint states j_i (and gripper state g_i) with those predicted by the simulation. By doing so, we can ensure that the physical robot follows the intended instructions and performs according to expectations. The deviations or errors identified during this process can be addressed and corrected. If a large deviation is detected, we immediately stop the task execution to prevent the failure case from causing any safety concerns. Through this validation step, we can improve the reliability, effectiveness, and safety of the robotic system.

Task Completion. In addition, we continually compare the robot's performance in the real world with simulation by verifying the completion of subtasks. To ensure that each subtask, such as *pick up the cup* or *place the cup*, is executed correctly, we apply visual question answering (VQA) techniques [AAL15] to verify the goal conditions from the camera's perspective. Specifically, we employ the newly released MiniGPT-4 [ZCS23] module to perform the VQA task in practice. This allows us to accurately assess the robot's ability to accomplish specific subtasks and identify discrepancies between the simulated and real-world environments from vision-based prediction.

Figure 7.5 illustrates verifying subtask completion using Visual Question Answering (VQA). To do this, we first compare the current robot configuration with the planned one after the planning and execution stages of a subtask. Next, we employ the MiniGPT-4 model to analyze the real-time visual input from the camera. Then, based on the specific context of each subtask, we formulate appropriate prompts for the VQA module and receive the corresponding answers from MiniGPT-4. Subsequently, we examine the keywords in these responses (such as *yes* or *no*) to confirm the successful completion of the subtask. This method provides an accurate and timely evaluation of the robot's progress, ensuring that it meets the requirements of each step along the way.

7.4.5 Results

Our experiments have demonstrated the effectiveness of our SIM2PLAN framework for zero-shot robot motion planning (without training the robot in real space). In the first set of trials, where the positions of the coffee machine, coffee capsule, and cup are fixed, our framework achieved a remarkable 90% success rate out of 20 attempts. When the positions of the capsule and cup are randomized, our framework could still successfully pick up the items 83.3% and 76.6% of the time, respectively. Despite the increased difficulty due to randomization, our framework managed to complete 75% of a total of 20 trials. Overall, these results highlight the robustness and adaptability of our SIM2PLAN framework in various environments and situations, making it a promising tool for robotic manipulation tasks.

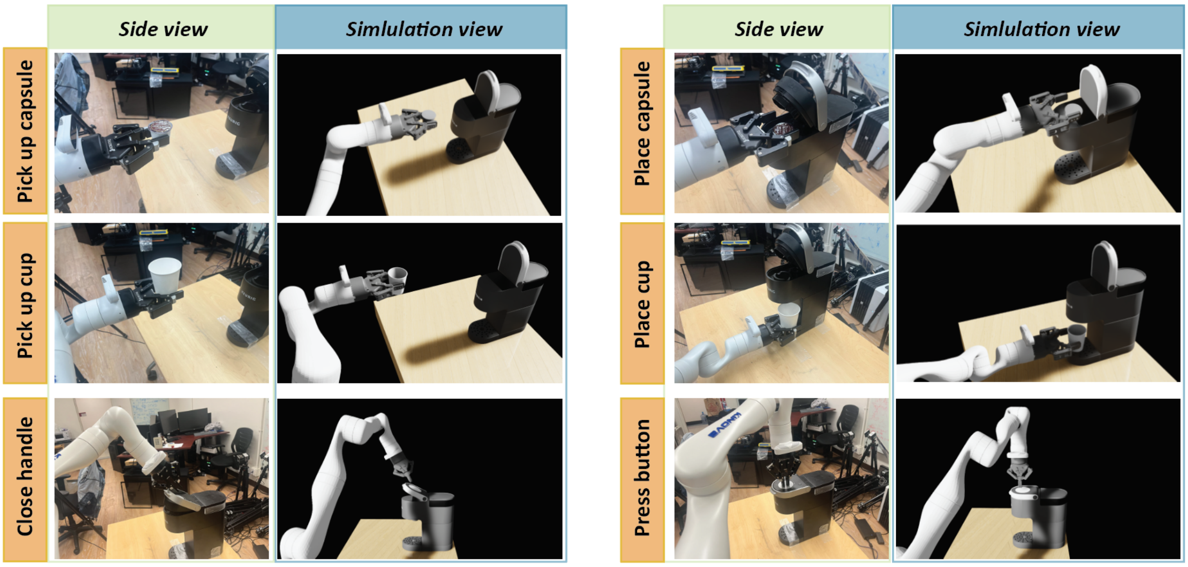


Figure 7.6: Real-world side camera view vs. simulated screenshot for subtask examples. Comparison between real-world images captured through a side camera and their simulated counterparts in the virtual environment.

Figure 7.6 presents several examples of subtasks performed by a robot from a side camera view in the real world and their corresponding screenshots taken from the simulation environment. Each block displays two images side by side, with the left one being the real-world snapshot and the right one showing the simulated scenario. These comparisons showcase how closely the simulation matches the real-world environment, demonstrating the validity of our proposed framework for zero-shot robot motion planning.

Furthermore, our proposed method also has practical benefits when applied to real-world settings. Since the robot was trained in a zero-shot manner during testing, it did not require any additional fine-tuning or retraining after being deployed to new environments. This means that the robot could quickly adapt to new situations without incurring significant delays or costs associated with retraining. By leveraging the motion planning from the digital twin, our method effectively reduces the amount of time required to train robots for specific tasks, making them more versatile and useful in a wide range of industries.

Chapter 8

Conclusion

In conclusion, this thesis delves into the study of simulation in the realm of artificial intelligence, specifically within the context of artificial general intelligence (AGI). The simulation environment serves as a crucial tool for training AI agents, enabling them to learn, adapt, and make decisions in a controlled and safe setting. The exploration of two fundamental challenges in AGI, efficient value-based training and bridging the simulation-to-reality gap, is at the heart of my research.

The research establishes the significance of both potential function learning (U) and value learning (V) in the path towards AGI. While potential function learning relies on data-driven methods, value learning allows AI agents to derive their goals, intents, and social values. By combining U and V learning, the agents' behavior becomes more aligned with human cognition and values.

The thesis unfolds in several distinct parts: the acquisition of practical knowledge and skills through potential function learning, the exploration of scenarios where utility-based learning approaches are limited, and the integration of U and V learning through simulation-based and data-driven methods. The study primarily assesses the effectiveness of U learning in simulated environments, progressing from controlled settings to more complex ones. Once compelling results are obtained within simulations, the focus shifts to transferring this knowledge to real-world scenarios. This transfer involves adapting the learned policies, strategies, and decision-making abilities to navigate the complexities of genuine environments.

The research outcomes emphasize the holistic approach required for AGI development, incorporating both U and V learning, leveraging simulations for training, and effectively transferring knowledge to reality. The contributions made in this thesis provide valuable insights and strategies for advancing the field of artificial intelligence, ultimately contributing to the realization of artificial general intelligence.

Bibliography

[AAA21] Anish Acharya, Suranjit Adhikari, Sanchit Agarwal, Vincent Auvray, Nehal Belgamwar, Arijit Biswas, Shubhra Chandra, Tagyoung Chung, Maryam Fazel-Zarandi, Raefer Gabriel, et al. "Alexa Conversations: An Extensible Data-driven Approach for Building Task-oriented Dialogue Systems." *arXiv preprint arXiv:2104.09088*, 2021.

[AAL15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. "Vqa: Visual question answering." In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

[ABB20] Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. "Task-Oriented Dialogue as Dataflow Synthesis." *Transactions of the Association for Computational Linguistics*, **8**:556–571, 2020.

[Abe09] Alex Abella. *Soldiers of reason: The RAND corporation and the rise of the American empire*. Houghton Mifflin Harcourt, 2009.

[Aga18] Abien Fred Agarap. “Deep learning using rectified linear units (relu).” *arXiv preprint arXiv:1803.08375*, 2018.

[AH19] Leonard Adolphs and Thomas Hofmann. “Ledeepchef: Deep reinforcement learning agent for families of text-based games.” *arXiv preprint arXiv:1909.01646*, 2019.

[AHK98] Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins SRI, Anthony Barrett, Dave Christianson, et al. “PDDL| The Planning Domain Definition Language.” Technical report, Technical Report, 1998.

[All02] Michael Allingham. *Choice theory: A very short introduction*. OUP Oxford, 2002.

[ALS19] Arjun R Akula, Changsong Liu, Sari Saba-Sadiya, Hongjing Lu, Sinisa Todorovic, Joyce Y Chai, and Song-Chun Zhu. “X-tom: Explaining with theory-of-mind for gaining justified human trust.” *arXiv preprint arXiv:1909.06907*, 2019.

[App10] Ian Apperly. *Mindreaders: the cognitive basis of" theory of mind"*. Psychology Press, 2010.

[Arr17] Rodrigo Torres Arrazate. “Development of a URDF file for simulation and programming of a delta robot using ROS.” *Santiago de Querétaro*, 2017.

[AUL20] Prithviraj Ammanabrolu, Jack Urbanek, Margaret Li, Arthur Szlam, Tim Rocktäschel, and Jason Weston. “How to Motivate Your Dragon: Teaching Goal-Driven Agents to Speak and Act in Fantasy Worlds.” *arXiv preprint arXiv:2010.00685*, 2020.

[AYC20] Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and Will Hamilton. “Learning dynamic belief graphs to generalize on text-based games.” *Advances in Neural Information Processing Systems*, **33**, 2020.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473*, 2014.

[BCC20] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. "Rearrangement: A challenge for embodied ai." *arXiv preprint arXiv:2011.01975*, 2020.

[BGB17] Tarek R Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C Lamb, Daniel Lowd, Priscila Machado Vieira Lima, et al. "Neural-symbolic learning and reasoning: A survey and interpretation." *arXiv preprint arXiv:1711.03902*, 2017.

[BMR20] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners." *arXiv preprint arXiv:2005.14165*, 2020.

[BPF21] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. "A Persistent Spatial Semantic Representation for High-level Natural Language Instruction Execution." *arXiv preprint arXiv:2107.05612*, 2021.

[Bro91] Rodney A Brooks. "Intelligence without representation." *Artificial intelligence*, **47**(1-3):139–159, 1991.

[BYM13] Fan Bao, Dong-Ming Yan, Niloy J Mitra, and Peter Wonka. "Generating and exploring good building layouts." *ACM Transactions on Graphics (TOG)*, **32**(4):1–10, 2013.

[CAT17] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. "EMNIST: Extending MNIST to handwritten letters." In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

[CCM18] Feiyang Chen, Nan Chen, Hanyang Mao, and Hanlin Hu. "Assessing four neural networks on handwritten digit recognition dataset (MNIST)." *arXiv preprint arXiv:1811.08278*, 2018.

[CD12] Jan Cieciuch and Eldad Davidov. "A comparison of the invariance properties of the PVQ-40 and the PVQ-21 to measure human values across German and Polish samples." In *Survey Research Methods*, volume 6, pp. 37–48, 2012.

[CGP20] Souradip Chakraborty, Aritra Roy Gosthipaty, and Sayak Paul. "G-SimCLR: Self-Supervised Contrastive Learning with Guided Projection via Pseudo Labelling." *arXiv preprint arXiv:2009.12007*, 2020.

[CKN20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations." *arXiv preprint arXiv:2002.05709*, 2020.

[CKY18] Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. "Textworld: A learning environment for text-based games." In *Workshop on Computer Games*, pp. 41–75. Springer, 2018.

[CM16] Kevin Clark and Christopher D. Manning. "Deep Reinforcement Learning for Mention-Ranking Coreference Models." In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2256–2262, Austin, Texas, November 2016. Association for Computational Linguistics.

[CRC20] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. "Generative pretraining from pixels." In *Proceedings of the 37th International Conference on Machine Learning*, volume 1, 2020.

[CSM14] Angel Chang, Manolis Savva, and Christopher D Manning. "Semantic parsing for text to 3d scene generation." In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pp. 17–21, 2014.

[CVB14] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the properties of neural machine translation: Encoder-decoder approaches." *arXiv preprint arXiv:1409.1259*, 2014.

[CWH71] Kenneth Mark Colby, Sylvia Weber, and Franklin Dennis Hilf. "Artificial paranoia." *Artificial Intelligence*, **2**(1):1–25, 1971.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*, 2018.

[DCL19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[DDG18] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. "Embodied question answering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–10, 2018.

[DDS09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database." In *CVPR09*, 2009.

[Den78] Daniel C Dennett. "Beliefs about beliefs [P&W, SR&B]." *Behavioral and Brain sciences*, **1**(4):568–570, 1978.

[Den12] Li Deng. "The mnist database of handwritten digit images for machine learning research [best of the web]." *IEEE Signal Processing Magazine*, **29**(6):141–142, 2012.

[DHH20] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew

Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. "RoboTHOR: An Open Simulation-to-Real Embodied AI Platform." In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[DKK14] Andreas Doumanoglou, Andreas Kargakos, Tae-Kyun Kim, and Sotiris Malassiotis. "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning." In *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 987–993. IEEE, 2014.

[DLM20] Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. "The second conversational intelligence challenge (convai2)." In *The NeurIPS'18 Competition*, pp. 187–208. Springer, 2020.

[DLT17] Yubin Deng, Chen Change Loy, and Xiaoou Tang. "Image aesthetic assessment: An experimental survey." *IEEE Signal Processing Magazine*, **34**(4):80–106, 2017.

[DRS19] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. "Wizard of Wikipedia: Knowledge-Powered Conversational Agents." In *International Conference on Learning Representations*, 2019.

[DSM19] Nishi Doshi, Gitam Shikkenawis, and Suman K Mitra. "Image Aesthetics Assessment Using Multi Channel Convolutional Neural Networks." In *International Conference on Computer Vision and Image Processing*, pp. 15–24. Springer, 2019.

[DYT22] 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*, 2022.

[EHH21] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. "Manipulathor: A framework for visual object manipulation." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4497–4506, 2021.

[EVW10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. “The pascal visual object classes (voc) challenge.” *International journal of computer vision*, **88**(2):303–338, 2010.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.” In *ICML*, 2017.

[FCG20] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Zengqi Xun, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, et al. “3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics.” *arXiv preprint arXiv:2011.09127*, 2020.

[FHS20] Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. “Social Chemistry 101: Learning to Reason about Social and Moral Norms.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 653–670, Online, November 2020. Association for Computational Linguistics.

[Fis70] Peter C Fishburn. “Utility theory for decision making.” Technical report, Research analysis corp McLean VA, 1970.

[Fle71] Joseph L Fleiss. “Measuring nominal scale agreement among many raters.” *Psychological bulletin*, **76**(5):378, 1971.

[FMS17] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm.” In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1615–1625, 2017.

[FN04] Jerome Feldman and Srinivas Narayanan. “Embodied meaning in a neural theory of language.” *Brain and language*, **89**(2):385–392, 2004.

[Fra97] Stan Franklin. “Autonomous agents as embodied AI.” *Cybernetics & Systems*, **28**(6):499–520, 1997.

[Fre07] Michael Freeman. *The complete guide to light & lighting in digital photography*. Sterling Publishing Company, Inc., 2007.

[FRS12] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. “Example-based synthesis of 3D object arrangements.” *ACM Transactions on Graphics (TOG)*, **31**(6):1–11, 2012.

[FSL15] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. “Activity-centric scene synthesis for functional 3D scene modeling.” *ACM Transactions on Graphics (TOG)*, **34**(6):1–13, 2015.

[FSR18] Denis Fedorenko, Nikita Smetanin, and Artem Rodichev. “Avoiding echo-responses in a retrieval-based conversation system.” In *Conference on Artificial Intelligence and Natural Language*, pp. 91–97. Springer, 2018.

[Gar14] Peter Gardenfors. *The geometry of meaning: Semantics based on conceptual spaces*. MIT press, 2014.

[GBS21] Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. “Paragraph-level commonsense transformers with recurrent memory.” In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[GGG22] Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. “Dialfred: Dialogue-enabled agents for embodied instruction following.” *arXiv preprint arXiv:2202.13330*, 2022.

[GGZ20] Xiaofeng Gao, Ran Gong, Yizhou Zhao, Shu Wang, Tianmin Shu, and Song-Chun Zhu. “Joint mind modeling for explanation generation in complex human-robot collaborative tasks.” In *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)*, pp. 1119–1126. IEEE, 2020.

[GH17] Andrew S Gordon and Jerry R Hobbs. *A formal theory of commonsense psychology: How people think people think.* Cambridge University Press, 2017.

[GHZ23] Ran Gong, Jiangyong Huang, Yizhou Zhao, Haoran Geng, Xiaofeng Gao, Qingyang Wu, Wensi Ai, Ziheng Zhou, Demetri Terzopoulos, Song-Chun Zhu, et al. "ARNOLD: A Benchmark for Language-Grounded Task Learning With Continuous States in Realistic 3D Scenes." *arXiv preprint arXiv:2304.04321*, 2023.

[GLC18] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. "Long text generation via adversarial training with leaked information." In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[GLG08] Artur SD'Avila Garcez, Luis C Lamb, and Dov M Gabbay. *Neural-symbolic cognitive reasoning*. Springer Science & Business Media, 2008.

[GM15] MY Ganaie and Hafiz Mudasir. "A study of social intelligence & academic achievement of college students of district Srinagar, J&K, India." *Journal of American Science*, **11**(3):23–27, 2015.

[GM16] Jon Gauthier and Igor Mordatch. "A paradigm for situated and goal-driven language learning." *arXiv preprint arXiv:1610.03585*, 2016.

[GMG20] Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. "COSMIC: COmmonSense knowledge for eMotion Identification in Conversations." In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2470–2481, Online, November 2020. Association for Computational Linguistics.

[Gri81] H Paul Grice. "Presupposition and conversational implicature." *Radical pragmatics*, **183**, 1981.

[Gri89] H Paul Grice. "Indicative conditionals." *Studies in the Way of Words*, pp. 58–85, 1989.

[GSA20] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, et al. "Threedworld: A platform for interactive multi-modal physical simulation." *arXiv preprint arXiv:2007.04954*, 2020.

[Haa14] John Haas. “A history of the unity game engine.” *Diss. WORCESTER POLYTECHNIC INSTITUTE*, 2014.

[HBB20] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. “Aligning ai with shared human values.” *arXiv preprint arXiv:2008.02275*, 2020.

[HBB21] Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. “Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs.” In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[HBE17] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. “Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings.” In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1766–1776, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[HBE22] Dániel Horváth, Kristóf Bocsi, Gábor Erdős, and Zoltán Istenes. “Sim2Real Grasp Pose Estimation for Adaptive Robotic Applications.” *arXiv preprint arXiv:2211.01048*, 2022.

[HBH21] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. “Sim2Real in robotics and automation: Applications and challenges.” *IEEE transactions on automation science and engineering*, **18**(2):398–400, 2021.

[HCK18] Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. “EmotionLines: An Emotion Corpus of Multi-Party Conversations.” In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[HD18] Dan Hendrycks and Thomas G Dietterich. "Benchmarking neural network robustness to common corruptions and surface variations." *arXiv preprint arXiv:1807.01697*, 2018.

[HE16] Jonathan Ho and Stefano Ermon. "Generative adversarial imitation learning." In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

[HGD19] Kaiming He, Ross Girshick, and Piotr Dollár. "Rethinking imagenet pre-training." In *Proceedings of the IEEE international conference on computer vision*, pp. 4918–4927, 2019.

[HGP20] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. "Leveraging the Feature Distribution in Transfer-based Few-Shot Learning." *arXiv preprint arXiv:2006.03806*, 2020.

[HJ15] Matthew Honnibal and Mark Johnson. "An Improved Non-monotonic Transition System for Dependency Parsing." In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1373–1378, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[HLM19] Patrik Haslum, Nir Lipovetzky, Daniele Magazzeni, and Christian Muise. "An introduction to the planning domain definition language." *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **13**(2):1–187, 2019.

[HM17] Matthew Honnibal and Ines Montani. "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing." To appear, 2017.

[HMV13] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. "Procedural content generation for games: A survey." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, **9**(1):1–22, 2013.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." *Neural computation*, **9**(8):1735–1780, 1997.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[ITF21] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. “How to train your robot with deep reinforcement learning: lessons we have learned.” *The International Journal of Robotics Research*, **40**(4-5):698–721, 2021.

[JAT20] Rishabh Jangir, Guillem Alenyà, and Carme Torras. “Dynamic Cloth Manipulation with Deep Reinforcement Learning.” In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4630–4636. IEEE, 2020.

[JDF11] Dhiraj Joshi, Ritendra Datta, Elena Fedorovskaya, Quang-Tuan Luong, James Z Wang, Jia Li, and Jiebo Luo. “Aesthetics and emotions in images.” *IEEE Signal Processing Magazine*, **28**(5):94–115, 2011.

[JGB17] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. “Bag of Tricks for Efficient Text Classification.” In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431. Association for Computational Linguistics, April 2017.

[JLZ22] Zhiwei Jia, Kaixiang Lin, Yizhou Zhao, Qiaozi Gao, Govind Thattai, and Gaurav S Sukhatme. “Learning to act with affordance-aware multimodal neural slam.” In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5877–5884. IEEE, 2022.

[JMA20] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. “Rlbench: The robot learning benchmark & learning environment.” *IEEE Robotics and Automation Letters*, **5**(2):3019–3026, 2020.

[Jon12] Jan de Jonge. “Rational and Moral Action.” In *Rethinking Rational Choice Theory*, pp. 199–206. Springer, 2012.

[JT20] Longlong Jing and Yingli Tian. "Self-supervised visual feature learning with deep neural networks: A survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[KBV16] Douwe Kiela, Luana Bulat, Anita L Vero, and Stephen Clark. "Virtual embodiment: A scalable long-term strategy for artificial intelligence research." *arXiv preprint arXiv:1610.07432*, 2016.

[KBY20] Takuro Karamatsu, Gibran Benitez-Garcia, Keiji Yanai, and Seiichi Uchida. "Iconify: Converting Photographs into Icons." In *Proceedings of the 2020 Joint Workshop on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia*, pp. 7–12, 2020.

[KMH17] 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.

[KO11] Bekir Karlik and A Vehbi Olgac. "Performance analysis of various activation functions in generalized MLP architectures of neural networks." *International Journal of Artificial Intelligence and Expert Systems*, **1**(4):111–122, 2011.

[KP19] Nikhil Krishnaswamy and James Pustejovsky. "Multimodal continuation-style architectures for human-robot interaction." *arXiv preprint arXiv:1909.08161*, 2019.

[KT15] Vikash Kumar and Emanuel Todorov. "Mujoco haptix: A virtual reality system for hand manipulation." In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 657–663. IEEE, 2015.

[KTJ06] Yan Ke, Xiaoou Tang, and Feng Jing. "The design of high-level features for photo quality assessment." In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pp. 419–426. IEEE, 2006.

[KWM11] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. "Twitter sentiment analysis: The good the bad and the omg!" In *Fifth International AAAI conference on weblogs and social media*, 2011.

[Lat12] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[LBC21] Nicholas Lourie, Ronan Le Bras, and Yejin Choi. "Scruples: A Corpus of Community Ethical Judgments on 32,000 Real-Life Anecdotes." In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[LBP19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks." *arXiv preprint arXiv:1908.02265*, 2019.

[LCC20] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. "You Impress Me: Dialogue Generation via Mutual Persona Perception." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1417–1427, Online, July 2020. Association for Computational Linguistics.

[LGG19] Manuel Lagunas, Elena Garces, and Diego Gutierrez. "Learning icons appearance similarity." *Multimedia Tools and Applications*, **78**(8):10733–10751, 2019.

[LHA21] Yuan Liang, Lei He, and Xiang Anthony'Chen. "Human-Centered AI for Medical Imaging." *Artificial Intelligence for Human Computer Interaction: A Modern Approach*, pp. 539–570, 2021.

[Lia16] Percy Liang. "Learning executable semantic parsers for natural language understanding." *Communications of the ACM*, **59**(9):68–76, 2016.

[LJ80] George Lakoff and Mark Johnson. "The metaphorical structure of the human conceptual system." *Cognitive science*, **4**(2):195–208, 1980.

[LK01] Steven M LaValle and James J Kuffner. "Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan." *Algorithmic and computational robotics*, pp. 303–307, 2001.

[LKT20] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems." *arXiv preprint arXiv:2005.01643*, 2020.

[LL21] Xiang Lisa Li and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." *arXiv preprint arXiv:2101.00190*, 2021.

[LLG19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461*, 2019.

[LLJ15] Xin Lu, Zhe Lin, Hailin Jin, Jianchao Yang, and James Z Wang. "Rating image aesthetics using deep learning." *IEEE Transactions on Multimedia*, **17**(11):2021–2034, 2015.

[LMB14] 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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

[LN12] Michael F Land and Dan-Eric Nilsson. *Animal eyes*. Oxford University Press, 2012.

[LOG19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692*, 2019.

[LR21] Teven Le Scao and Alexander M Rush. "How many data points is a prompt worth?" In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2627–2636, 2021.

[LS20] Corey Lynch and Pierre Sermanet. "Language conditioned imitation learning over unstructured data." *arXiv preprint arXiv:2005.07648*, 2020.

[LSS17] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. "DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset." In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 986–995, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[LST19] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. "The Omniglot challenge: a 3-year progress report." *Current Opinion in Behavioral Sciences*, **29**:97–104, 2019.

[LSY19] Hsu-Chao Lai, Hong-Han Shuai, De-Nian Yang, Jiun-Long Huang, Wang-Chien Lee, and Philip S Yu. "Social-aware VR configuration recommendation via multi-feedback coupled tensor factorization." In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1773–1782, 2019.

[LUT17] 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**, 2017.

[LWO20] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation." *arXiv preprint arXiv:2011.07215*, 2020.

[LWT11] Wei Luo, Xiaogang Wang, and Xiaoou Tang. "Content-based photo quality assessment." In *2011 International Conference on Computer Vision*, pp. 2206–2213. IEEE, 2011.

[LXM21] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. "Igibson 2.0: Object-centric simulation for robot learning of everyday household tasks." *arXiv preprint arXiv:2108.03272*, 2021.

[LZR23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. "Grounding dino: Marrying dino with grounded pre-training for open-set object detection." *arXiv preprint arXiv:2303.05499*, 2023.

[LZW20] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B Tenenbaum. "End-to-End Optimization of Scene Layout." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3754–3763, 2020.

[MBT18] Spandan Madan, Zoya Bylinskii, Matthew Tancik, Adrià Recasens, Kimberli Zhong, Sami Alsheikh, Hanspeter Pfister, Aude Oliva, and Fredo Durand. "Synthetically trained icon proposals for parsing and summarizing infographics." *arXiv preprint arXiv:1807.10441*, 2018.

[MGS22] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. "Simple open-vocabulary object detection with vision transformers." *arXiv preprint arXiv:2205.06230*, 2022.

[MHB22] Oier Mees, Lukas Hermann, and Wolfram Burgard. "What Matters in Language Conditioned Robotic Imitation Learning." *arXiv preprint arXiv:2204.06252*, 2022.

[MHR22] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. "CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks." *IEEE Robotics and Automation Letters*, 2022.

[MJB16] Tomas Mikolov, Armand Joulin, and Marco Baroni. "A roadmap towards machine intelligence." In *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 29–61. Springer, 2016.

[MLX21] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations." *arXiv preprint arXiv:2107.14483*, 2021.

[MPF18] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. "Language-driven synthesis of 3D scenes from scene databases." *ACM Transactions on Graphics (TOG)*, **37**(6):1–16, 2018.

[MTP05] Marco Morales, Lydia Tapia, Roger Pearce, Samuel Rodriguez, and Nancy M Amato. "A machine learning approach for feature-sensitive motion planning." *Algorithmic Foundations of Robotics VI*, **17**:361–376, 2005.

[MVC10] Gary McKeown, Michel F Valstar, Roderick Cowie, and Maja Pantic. "The SEMAINE corpus of emotionally coloured character interactions." In *2010 IEEE International Conference on Multimedia and Expo*, pp. 1079–1084. IEEE, 2010.

[MVL14] Chongyang Ma, Nicholas Vining, Sylvain Lefebvre, and Alla Sheffer. "Game level layout from design specification." In *Computer Graphics Forum*, volume 33, pp. 95–104. Wiley Online Library, 2014.

[MWG21] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. "Isaac gym: High performance gpu-based physics simulation for robot learning." *arXiv preprint arXiv:2108.10470*, 2021.

[MXW21] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. "What Matters in Learning from Offline Human Demonstrations for Robot Manipulation." In *arXiv preprint arXiv:2108.03298*, 2021.

[NCC20] Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. "House-GAN: Relational Generative Adversarial Networks for Graph-constrained House Layout Generation." *arXiv preprint arXiv:2003.06988*, 2020.

[Nil13] Dan-E Nilsson. "Eye evolution and its functional basis." *Visual neuroscience*, **30**(1-2):5–20, 2013.

[NKB15] Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. "Language understanding for text-based games using deep reinforcement learning." *arXiv preprint arXiv:1506.08941*, 2015.

[NMZ17] Zhangkai Ni, Lin Ma, Huanqiang Zeng, Jing Chen, Canhui Cai, and Kai-Kuang Ma. "ESIM: Edge similarity for screen content image quality assessment." *IEEE Transactions on Image Processing*, **26**(10):4818–4831, 2017.

[NOS11] Masashi Nishiyama, Takahiro Okabe, Imari Sato, and Yoichi Sato. "Aesthetic quality classification of photographs based on color harmony." In *CVPR 2011*, pp. 33–40. IEEE, 2011.

[NSA22] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, et al. "Factory: Fast Contact for Robotic Assembly." *arXiv preprint arXiv:2205.03532*, 2022.

[NVI22a] NVIDIA Corporation. "NVIDIA Omniverse." https://www.nvidia.com/en-us/omniverse/, 2022. Online.

[NVI22b] NVIDIA Corporation. "NVIDIA PhysX System Software." https://www.nvidia.com/en-us/drivers/physx/physx-9-19-0218-driver/, 2022. Online.

[NZ08] Maria-Elena Nilsback and Andrew Zisserman. "Automated flower classification over a large number of classes." In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

[OKB19] Marin Orsic, Ivan Kreso, Petra Bevandic, and Sinisa Segvic. "In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 12607–12616, 2019.

[OS18] Takuma Okuda and Sanae Shoda. "AI-based chatbot service for financial industry." *Fujitsu Scientific and Technical Journal*, **54**(2):4–8, 2018.

[PHM19] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. "MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 527–536, Florence, Italy, July 2019. Association for Computational Linguistics.

[PLL20] Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. "SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model." *arXiv preprint arXiv:2005.05298*, 2020.

[PRB18] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. "Virtualhome: Simulating household activities via programs." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8494–8502, 2018.

[PSM14a] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[PSM14b] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation." In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[PSS21] Alexander Pashevich, Cordelia Schmid, and Chen Sun. "Episodic Transformer for Vision-and-Language Navigation." *arXiv preprint arXiv:2105.06453*, 2021.

[PW78] David Premack and Guy Woodruff. "Does the chimpanzee have a theory of mind?" *Behavioral and brain sciences*, **1**(4):515–526, 1978.

[Pyl78] Zenon W Pylyshyn. "When is attribution of beliefs justified?[P&W]." *Behavioral and brain sciences*, **1**(4):592–593, 1978.

[QLZ21] Liang Qiu, Yuan Liang, Yizhou Zhao, Pan Lu, Baolin Peng, Zhou Yu, Ying Nian Wu, and Song-Chun Zhu. "SocAoG: Incremental Graph Parsing for Social Relation

Inference in Dialogues." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 658–670, Online, August 2021. Association for Computational Linguistics.

[QZH18] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. "Human-centric indoor scene synthesis using stochastic grammar." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5899–5908, 2018.

[QZL21] Liang Qiu, Yizhou Zhao, Yuan Liang, Pan Lu, Weiyan Shi, Zhou Yu, and Song-Chun Zhu. "Towards Socially Intelligent Agents with Mental State Transition and Human Utility." *arXiv preprint arXiv:2103.07011*, 2021.

[QZL22] Liang Qiu, Yizhou Zhao, Jinchao Li, Pan Lu, Baolin Peng, Jianfeng Gao, and Song-Chun Zhu. "ValueNet: A New Dataset for Human Value Driven Dialogue System." *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**(10):11183–11191, Jun. 2022.

[QZS20] Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-Chun Zhu. "Structured Attention for Unsupervised Dialogue Structure Induction." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1889–1899, Online, November 2020. Association for Computational Linguistics.

[RCV21] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. "3D Semantic Scene Completion: a Survey." *arXiv preprint arXiv:2103.07466*, 2021.

[RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*, **28**, 2015.

[RIK18] Nathan D Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. "Riemannian motion policies." *arXiv preprint arXiv:1801.02854*, 2018.

[RKH21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision." In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

[RRB19] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. "Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML." In *International Conference on Learning Representations*, 2019.

[RSL19] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. "Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5370–5381, Florence, Italy, July 2019. Association for Computational Linguistics.

[RSM86] David E Rumelhart, Paul Smolensky, James L McClelland, and G Hinton. "Sequential thought processes in PDP models." *Parallel distributed processing: explorations in the microstructures of cognition*, **2**:3–57, 1986.

[RWC19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners." *OpenAI blog*, **1**(8):9, 2019.

[RWL19] Daniel Ritchie, Kai Wang, and Yu-an Lin. "Fast and flexible indoor scene synthesis via deep convolutional generative models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6182–6190, 2019.

[SBK20] Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. "Moca: A modular object-centric approach for interactive instruction following." *arXiv preprint arXiv:2012.03208*, 2020.

[Sch92] Shalom H Schwartz. "Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries." In *Advances in experimental social psychology*, volume 25, pp. 1–65. Elsevier, 1992.

[Sch12] Shalom H Schwartz. “An overview of the Schwartz theory of basic values.” *Online readings in Psychology and Culture*, **2**(1):2307–0919, 2012.

[SCT20] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. “Metasdf: Meta-learning signed distance functions.” *Advances in Neural Information Processing Systems*, **33**:10136–10147, 2020.

[SCU21] 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*, **34**, 2021.

[SCV12] Shalom H Schwartz, Jan Cieciuch, Michele Vecchione, Eldad Davidov, Ronald Fischer, Constanze Beierlein, Alice Ramos, Markku Verkasalo, Jan-Erik Lönnqvist, Kursad Demirutku, et al. “Refining the theory of basic individual values.” *Journal of personality and social psychology*, **103**(4):663, 2012.

[SDC19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” *arXiv preprint arXiv:1910.01108*, 2019.

[SGT21] Alessandro Suglia, Qiaozi Gao, Jesse Thomason, Govind Thattai, and Gaurav Sukhatme. “Embodied bert: A transformer model for embodied, language-guided visual task completion.” *arXiv preprint arXiv:2108.04927*, 2021.

[SKB18] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. “Modeling relational data with graph convolutional networks.” In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

[SKF16] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. “Bidirectional attention flow for machine comprehension.” *arXiv preprint arXiv:1611.01603*, 2016.

[SKM19] 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*, pp. 9339–9347, 2019.

[SLC19] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. “Meta-transfer learning for few-shot learning.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 403–412, 2019.

[SLL22] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. “BEHAVIOR: Benchmark for Everyday Household Activities in Virtual, Interactive, and Ecological Environments.” In *Conference on Robot Learning*, pp. 477–490. PMLR, 2022.

[SMF22] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. “Cliport: What and where pathways for robotic manipulation.” In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022.

[SMZ19] Wei Sun, Xiongkuo Min, Guangtao Zhai, Ke Gu, Huiyu Duan, and Siwei Ma. “MC360IQA: A Multi-channel CNN for Blind 360-Degree Image Quality Assessment.” *IEEE Journal of Selected Topics in Signal Processing*, **14**(1):64–77, 2019.

[SRC19] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. “Social IQa: Commonsense Reasoning about Social Interactions.” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics.

[SS12] Jean-Philippe Saut and Daniel Sidobre. “Efficient models for grasp planning with a multi-fingered hand.” *Robotics and Autonomous Systems*, **60**(3):347–357, 2012.

[SSS19] Yosuke Shinya, Edgar Simo-Serra, and Taiji Suzuki. “Understanding the Effects of Pre-Training for Object Detectors via Eigenspectrum.” In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning." In *Advances in neural information processing systems*, pp. 4077–4087, 2017.

[Sta02] Robert Stalnaker. "Common ground." *Linguistics and philosophy*, **25**(5/6):701–721, 2002.

[STG20] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. "Alfred: A benchmark for interpreting grounded instructions for everyday tasks." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020.

[Sti10] Mike Stilman. "Global manipulation planning in robot joint space with task constraints." *IEEE Transactions on Robotics*, **26**(3):576–584, 2010.

[Sun94] Ron Sun. *Integrating rules and connectionism for robust commonsense reasoning*. John Wiley & Sons, Inc., 1994.

[SVT16] Arjen Stolk, Lennart Verhagen, and Ivan Toni. "Conceptual alignment: How brains achieve mutual understanding." *Trends in cognitive sciences*, **20**(3):180–191, 2016.

[SXL20] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Shyamal Buch, Claudia D'Arpino, Sanjana Srivastava, Lyne P Tchapmi, et al. "iGibson, a Simulation Environment for Interactive Tasks in Large RealisticScenes." *arXiv preprint arXiv:2012.02924*, 2020.

[SYZ17] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. "Semantic scene completion from a single depth image." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1746–1754, 2017.

[TCB21] Joanne Truong, Sonia Chernova, and Dhruv Batra. "Bi-directional domain adaptation for sim2real transfer of embodied navigation agents." *IEEE Robotics and Automation Letters*, **6**(2):2634–2641, 2021.

[TE21] Tuan Tran and Chinwe Ekenna. “Identifying Valid Robot Configurations via a Deep Learning Approach.” In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8973–8978. IEEE, 2021.

[TL19] Mingxing Tan and Quoc V Le. “Efficientnet: Rethinking model scaling for convolutional neural networks.” *arXiv preprint arXiv:1905.11946*, 2019.

[TSH20] Ronen Tamari, Chen Shani, Tom Hope, Miriam R L Petruck, Omri Abend, and Dafna Shahaf. “Language (Re)modelling: Towards Embodied Language Understanding.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6268–6281, Online, July 2020. Association for Computational Linguistics.

[TZC18] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. “Sim-to-real: Learning agile locomotion for quadruped robots.” *arXiv preprint arXiv:1804.10332*, 2018.

[TZL18] Fei Tao, He Zhang, Ang Liu, and Andrew YC Nee. “Digital twin in industry: State-of-the-art.” *IEEE Transactions on industrial informatics*, **15**(4):2405–2415, 2018.

[UFK19] Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. “Learning to Speak and Act in a Fantasy Text Adventure Game.” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 673–683, Hong Kong, China, November 2019. Association for Computational Linguistics.

[VBL16] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. “Matching networks for one shot learning.” In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention is all you need.” *arXiv preprint arXiv:1706.03762*, 2017.

[VVP15] Axel Vick, Vojtěch Vonásek, Robert Pěnička, and Jörg Krüger. “Robot control as a service—towards cloud-based motion planning and control for industrial robots.” In *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, pp. 33–39. IEEE, 2015.

[WDK21] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. “Visual Room Rearrangement.” In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[WDS20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. “Transformers: State-of-the-Art Natural Language Processing.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics.

[Wei66] Joseph Weizenbaum. “ELIZA—a computer program for the study of natural language communication between man and machine.” *Communications of the ACM*, **9**(1):36–45, 1966.

[Wil92] Ronald J Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning.” *Machine learning*, **8**(3):229–256, 1992.

[WLW19] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. “Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks.” *ACM Transactions on Graphics (TOG)*, **38**(4):1–15, 2019.

[WSC18] Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. “Deep convolutional priors for indoor scene synthesis.” *ACM Transactions on Graphics (TOG)*, **37**(4):1–14, 2018.

[WSK20] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. “Allenact: A framework for embodied ai research.” *arXiv preprint arXiv:2008.12760*, 2020.

[WYN20] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. “SceneFormer: Indoor Scene Generation with Transformers.” *arXiv preprint arXiv:2012.09793*, 2020.

[XQM20] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. “Sapien: A simulated part-based interactive environment.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11097–11107, 2020.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.” *arXiv preprint arXiv:1708.07747*, 2017.

[XSF02] Ken Xu, James Stewart, and Eugene Fiume. “Constraint-based automatic placement for scene composition.” In *Graphics Interface*, volume 2, pp. 25–34, 2002.

[XSX16] Caiming Xiong, Nishant Shukla, Wenlong Xiong, and Song-Chun Zhu. “Robot learning with a spatial, temporal, and causal and-or graph.” In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2144–2151. IEEE, 2016.

[YCS18] Xingdi Yuan, Marc-Alexandre Côté, Alessandro Sordoni, Romain Laroche, Remi Tachet des Combes, Matthew Hausknecht, and Adam Trischler. “Counting to explore and generalize in text-based games.” *arXiv preprint arXiv:1806.11525*, 2018.

[YDL18] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. “Qanet: Combining local convolution with global self-attention for reading comprehension.” *arXiv preprint arXiv:1804.09541*, 2018.

[YL17] Wang Yuan and Zhijun Li. "Development of a human-friendly robot for socially aware human-robot interaction." In *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 76–81. IEEE, 2017.

[YLF15] Yezhou Yang, Yi Li, Cornelia Fermuller, and Yiannis Aloimonos. "Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web." In *Twenty-ninth AAAI conference on artificial intelligence*. Citeseer, 2015.

[YM19] Xusen Yin and Jonathan May. "Comprehensible context-driven text game playing." In *2019 IEEE Conference on Games (CoG)*, pp. 1–8. IEEE, 2019.

[YMB18] Claudia Yan, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi. "Chalet: Cornell house agent learning environment." *arXiv preprint arXiv:1801.07357*, 2018.

[YSC20] Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. "Dialogue-Based Relation Extraction." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4927–4940, Online, July 2020. Association for Computational Linguistics.

[YWG18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding." *arXiv preprint arXiv:1810.02338*, 2018.

[YYT11] Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. "Make it home: automatic optimization of furniture arrangement." *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86*, **30**(4), 2011.

[ZC21] Yichi Zhang and Joyce Chai. "Hierarchical Task Learning from Language Instructions with Unified Transformers and Self-Monitoring." *arXiv preprint arXiv:2106.03427*, 2021.

[ZCJ22] Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Eric Wang. "VLMbench: A Compositional Benchmark for Vision-and-Language Manipulation." *arXiv preprint arXiv:2206.08522*, 2022.

[ZCS23] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. "MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models." *arXiv preprint arXiv:2304.10592*, 2023.

[ZDU18] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. "Personalizing Dialogue Agents: I have a dog, do you have pets too?" In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[ZGL20] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. "The Design and Implementation of XiaoIce, an Empathetic Social Chatbot." *Computational Linguistics*, **46**(1):53–93, March 2020.

[Zha19] Ran Zhao. *Socially-Aware Dialogue System*. PhD thesis, Carnegie Mellon University, 2019.

[ZLJ21] Yizhou Zhao, Kaixiang Lin, Zhiwei Jia, Qiaozi Gao, Govind Thattai, Jesse Thomason, and Gaurav S Sukhatme. "Luminous: Indoor scene generation for embodied ai challenges." *arXiv preprint arXiv:2111.05527*, 2021.

[ZM20] Guangtao Zhai and Xiongkuo Min. "Perceptual image quality assessment: a survey." *SCIENCE CHINA Information Sciences*, **63**(11):211301, 2020.

[ZMB08] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. "Maximum entropy inverse reinforcement learning." In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

[ZRR18] Ran Zhao, Oscar J Romero, and Alex Rudnicky. "SOGO: a social intelligent negotiation dialogue system." In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pp. 239–246, 2018.

[ZSG19] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. "Dialogpt: Large-scale generative pre-training for conversational response generation." *arXiv preprint arXiv:1911.00536*, 2019.

[ZSG20] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. "DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 270–278, Online, July 2020. Association for Computational Linguistics.

[ZWK19] Yang Zhou, Zachary While, and Evangelos Kalogerakis. "Scenegraphnet: Neural message passing for 3d indoor scene augmentation." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7384–7392, 2019.

[ZYM20] Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. "Deep generative modeling for scene synthesis via hybrid representations." *ACM Transactions on Graphics (TOG)*, **39**(2):1–21, 2020.