Abductive Learning for Neuro-Symbolic Grounded Imitation

Jie-Jing Shao*
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing, China
shaojj@lamda.nju.edu.cn

Hao-Ran Hao*
National Key Laboratory for Novel Software Technology
School of Artificial Intelligence
Nanjing University, Nanjing, China
hrhao.ai@gmail.com

Xiao-Wen Yang
National Key Laboratory for Novel Software Technology
School of Artificial Intelligence
Nanjing University, Nanjing, China
yangxw@lamda.nju.edu.cn

Yu-Feng Li†
National Key Laboratory for Novel Software Technology
School of Artificial Intelligence
Nanjing University, Nanjing, China
liyf@nju.edu.cn

ABSTRACT

Recent learning-to-imitation methods have shown promising results in planning via imitating on the observation-action space. However, their ability in open environments is usually limited, especially in long-horizon tasks. In contrast, traditional symbolic planning excels at long-horizon tasks via logical reasoning on the human-defined symbolic spaces, but is weak at handling observations beyond symbolic states, like high-dimensional visual inputs in the real world. In this work, we borrow the idea of abductive learning and introduce a novel framework ABductive Imitation Learning (ABIL) that combines the benefits of data-driven learning and symbolic-based reasoning, enabling long-term planning. Specifically, we employ abductive reasoning to understand the demonstrations in symbolic space and design the principles of sequential consistency to resolve the conflict between perception and reasoning. It generates predicate candidates to promote the perception from raw observations to symbolic space without laborious predicate annotations, providing a groundwork for symbolic planning. With the symbolic understanding, we further build a policy ensemble whose base policies are built with different logical objectives and managed by symbolic reasoning. Experiments show that our proposal successfully understands the observations with the task-relevant symbolics to assist the imitating. Importantly, ABIL shows significantly improved performance in data efficiency and generalization settings within various long-horizon tasks.

CCS CONCEPTS

• **Computing methodologies → Learning from demonstrations**; *Knowledge representation and reasoning*.

*Both authors contributed equally to this research.
†Corresponding Author.

KEYWORDS

Imitation Learning, Abductive Learning, Neuro-Symbolic Learning

ACM Reference Format:
Jie-Jing Shao, Hao-Ran Hao, Xiao-Wen Yang, and Yu-Feng Li. 2025. Abductive Learning for Neuro-Symbolic Grounded Imitation. In *Proceedings of KDD '25: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '25)*. ACM, New York, NY, USA, 15 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

A long-standing goal in AI is to build agents that are flexible and general, able to accomplish a diverse set of tasks in open and novel environments, such as home robots for cooking meals or assembling furniture. These tasks generally require the agents to execute sequential decision-making, which is often formulated as a planning problem. Recently, the learning-based method, *Imitation Learning*, has achieved remarkable success via imitating expert demonstrations, in a variety of domains, such as robotic manipulation [9, 32], autonomous driving [3, 26] and language models [4, 34]. However, the theoretical studies [27, 36] reveal that imitation learning can suffer from serious performance degradation due to the covariate shift between limited expert demonstrations and the state distribution actually encountered by the agents, especially in long-horizon tasks. In traditional AI literature, symbolic planners effectively generalize in long-horizon decision-making, via logical reasoning on the human-defined symbolic spaces [10, 11, 13]. However, they typically abstract away perception with ground-truth symbols. Given observations and actions, without predicate-level supervision, pure logic-based methods struggle to map raw observations to the corresponding human-defined symbolic spaces.

To address these issues, efforts are underway to merge the advantages of learning-based and reasoning-based approaches into neuro-symbolic planning. Xu et al. [35] propose the regression planning network, learning to predict symbolic sub-goals that need to be achieved before the final goals, thereby generating a long-term symbolic plan conditioned on high-dimensional observations. Konidaris et al. [21] collect feasibility annotations and transition data under different symbolic operations to learn the symbolic representation of different observations. The learned

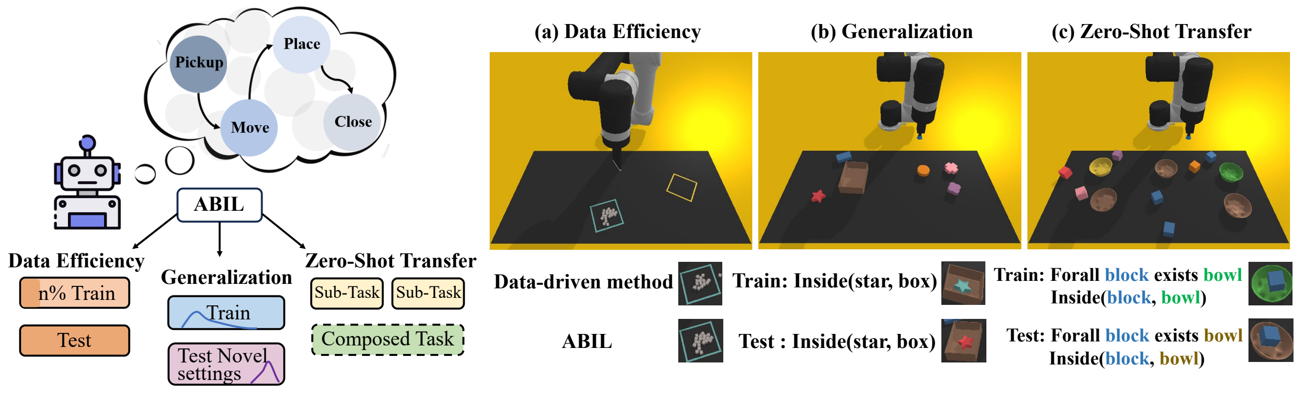


Figure 1: Our framework, ABductive Imitation Learning (ABIL), achieves neuro-symbolic grounding of imitation in complex scenes while showing state-of-the-art results in data efficiency, generalization, and zero-shot transfer.

representation enables traditional planning in the symbolic space and allows the acquisition of desired low-level controllers during inference. Silver et al. [31] formalizes operator learning for neuro-symbolic learning, viewing operators as an abstraction model of the raw transition and generating the high-level plan skeletons. However, most of these positive results typically assume that there are sufficient symbolic annotations to train the neural network to map high-dimensional observations to symbolic states for logic-based planning, or there are prior low-level controllers to achieve expected sub-goals perfectly. Compared to real-world applications, these approaches omit the learning from observations to imitate specific behaviors. The most relevant work to ours is PDSketch [24], which uses a neural network as the basic module of a human-specified programming structure and learns a transition model. This model supports generic network-based representations for predicates and action effects. Nevertheless, its model-based planning framework tends to accumulate errors which is not ideal for long-horizon decision-making tasks.

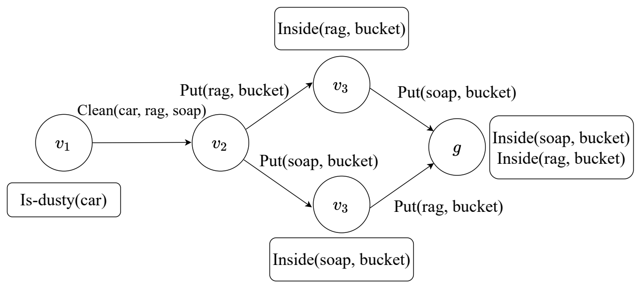
In this work, we borrow the idea of abductive learning and introduce a novel framework ABductive Imitation Learning (ABIL) that combines the benefits of data-driven learning and symbolic-based reasoning, enabling long-term planning with observations. Specifically, we employ abductive reasoning to understand the demonstrations in symbolic space and apply the principles of sequential consistency to resolve the conflict between perception and reasoning. It applies logical reasoning to generate predicate candidates that meet constraints, eliminating the need for laborious symbolic annotations. With the above symbolic understanding, we further build a policy ensemble whose base policies are built with different logical objectives and managed by symbolic reasoning. The learned policy imitates specific behaviors from observations, eliminating the need for prior low-level controllers used in earlier neuro-symbolic methods. Additionally, it makes decisions based on human-like cognition, enhancing its ability for generalization. Experiments show that our proposal successfully understands the observations with the task-relevant symbolics to assist the imitating. Importantly, ABIL shows significantly improved performance in data efficiency and generalization settings within a variety of long-horizon tasks.

2 RELATED WORK

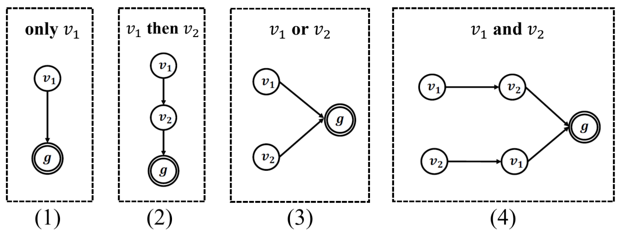
The preface work of this paper mainly includes *Imitation Learning*, *Neuro-Symbolic Planning* and *Abductive Learning*.

Imitation Learning learns the policies from expert demonstrations to address sequential decision-making [19]. There are many works that achieved successful results on varying domains, such as robotic manipulation [9, 32], autonomous driving [3, 26] and language models [4, 34]. However, learning theory reveals that the generalization of imitation learning is limited by the size of the expert dataset and degrades with the increase in decision-making horizons [27]. Especially in open environments, where home agents need to accomplish tasks in differently arranged rooms, the distribution shift, that is, the covariate shift between training observations and the scenarios the agent actually encounters, presents a greater challenge for imitation learning [20, 22].

Neuro-Symbolic Planning explores to combine traditional symbolic planning with learning that has strong generalization capabilities. To handle observations beyond symbolic states, previous studies typically involve training neural networks with task-relevant predicate annotations to transform raw observations into symbolic states for planning. For example, Xu et al. [35] propose the regression planning network, learning to predict sub-goals that need to be achieved before the final goals, enabling the traditional symbolic regression to complex high-dimensional inputs, like images. Konidaris et al. [21] collect feasibility annotations and transition data under different symbolic operations to learn the symbolic representation of different observations. Silver et al. [31] formalize operator learning for neuro-symbolic learning, viewing operators as an abstraction model of the raw transition and generating the high-level plan skeletons. Wang et al. [33] leverage a pre-trained vision-language model to provide the predicate-level annotations to help imitation learning. The most related work to ours is PDSketch [24], which utilizes the neural network as the basic module of human-specified programming structure and learns



(a) Graphical knowledge of Cleaning a Car.



(b) The typical graphical structures of the state machine

Figure 2: Illustration of the Knowledge Base.

an object-factored transition model that supports generic neural-network-based representations for predicates and action effects. However, we find that its planning, based on the raw-observation-action space, tends to accumulate errors and is not suitable for long-sequence decision-making tasks. In addition, its application of logical reasoning is rather limited and usually requires large amounts of training data to achieve neuro-symbolic grounding. In contrast, we develop sequential consistency for abductive reasoning which results in a more data-efficient grounding.

Abductive Learning provides a framework that integrates the machine learning and logical reasoning [7, 16, 18, 39]. It focuses on how to deal with the intermediate neuro-symbolic grounding which serves as pseudo-labels for learning and variables for abduction. Although there are some explorations to extend abductive learning to different applications, such as judicial sentencing [17] and historical document understanding [12], they mainly consider the traditional classification tasks. In this work, we focus on the planning problems, where long-horizon sequential decision-making tasks are mainly considered.

Generally speaking, it is still challenging for the above technologies to implement imitation in the real world. Considering that humans can relatively easily provide a knowledge base with high-level symbolic solutions for long-term tasks like robotic manipulation [21] or household tasks [20, 24, 35], this work follows the research line of neural-symbolic methods to use a knowledge base to help with imitation learning, while also avoiding the requirements on tedious predicate annotations.

3 THE PROPOSED FRAMEWORK

3.1 Problem Formulation

In this paper, we focus on the goal-based planning task. Following [24, 29, 30], the environment is formally defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{P}, \mathcal{OP}, \mathcal{S}^0, g\rangle$. Here, $\mathcal{S}$ represents the state space, and $\mathcal{A}$ represents the action space. The transitions between states and actions are governed by a deterministic transition function $\mathcal{T}$: $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$. $\mathcal{S}^0$ denotes the distribution of initial states. The set $\mathcal{O}$ consists of a finite number of task-related objects, where each object o possesses a unique name, such as *car* and *rag*. Additionally, $\mathcal{P}$ is a finite set of task-related predicate symbols, where each predicate symbol p has an arity that indicates the number of arguments it takes. For example, *Inside/2* has an arity of two, representing that one object is inside another. A ground atom $\overline{p}$ is a predicate that only contains concrete objects, such as *Inside(rag, bucket)*. If a state s satisfies $s \models \overline{p}$, it indicates that s semantically entails the interpretation of $\overline{p}$. The set g consists of ground atoms that represent the task's target. The task is to find an action sequence that generates a trajectory $(s^0, a^1, s^1, ..., a^T, s^T)$ satisfying $\forall \overline{p} \in g, s^T \models \overline{p}$. For simplification, we denote $s \models P$ (where P is a set of ground atoms) as $\forall \overline{p} \in P, s \models \overline{p}$. Moreover, $\mathcal{OP}$ is also a set of finite predicate symbols that represent logical operators, such as *clean/3* and *put/2*. We denote $obj(\overline{p})$ to retrieve the object(s) from a ground atom. For instance, $obj(Put(rag, bucket)) = \{rag, bucket\}$.

The symbolic knowledge base provided by experts could be formulated as a finite-state machine with a directed graph $G = \langle V, E\rangle$. Each node v in the vertex set V contains a set of ground atoms of $\mathcal{P}$, which can be viewed as the condition of a sub-task. Each edge is noted as a tuple $\langle \overline{op}, \text{EFF}^+, \text{EFF}^-\rangle$. $\overline{op}$ is a ground atom of $\mathcal{OP}$ representing the symbolic action, e.g., *Put(rag, bucket)*. EFF^+ is the add effect and EFF^- is the delete effect, each is a set of grounding atoms. A symbolic action op typically requires multiple actions $a \sim \mathcal{A}$ to achieve the desired logical sub-goal, corresponding to a segment of the complete trajectory. For the sake of simplicity, the notation $\overline{\text{op}}$ is utilized to denote the edge.

For each node $u, v \in V$, if there is a directed edge pointing from u to v, then $v = (u - \text{EFF}^-) \cup \text{EFF}^+$. We define a trajectory $z = (s^0, a^1, s^1, ..., a^T, s^T)$ satisfying the knowledge base G denoted as $z \models G$ if and only if for every adjacent states pair (s^t, s^{t+1}), there exists $u \in V, s^t \models u \wedge s^{t+1} \models u$ or $s^t \models u$, $s^{t+1} \models v, \exists u, v \in V, (u, v) \in E$. This indicates that the expert trajectory satisfies the corresponding symbolic knowledge base, such as first using a rag and soap to clean a dusty car, and then putting them into a bucket. Figure 2a demonstrates an example of the knowledge base formalized as a state machine with a directed graph.

The state machine contains multiple basic structures as illustrated in Figure 2b. In the event that a singular node, denoted as v_1, directs towards the goal, it signifies the necessity to address a corresponding sub-task v_1. For instance, the action *Clean(car, rag, soap)* is imperative whenever the objective is to clean a car. Conversely, should there be a directed edge from v_1 to v_2, with v_2 subsequently pointing towards the goal, it implies that the sub-task v_1 must be solved before sub-task v_2. In scenarios where both v_1 and v_2 have directed edges towards the goal, either sub-task v_1 or v_2 is

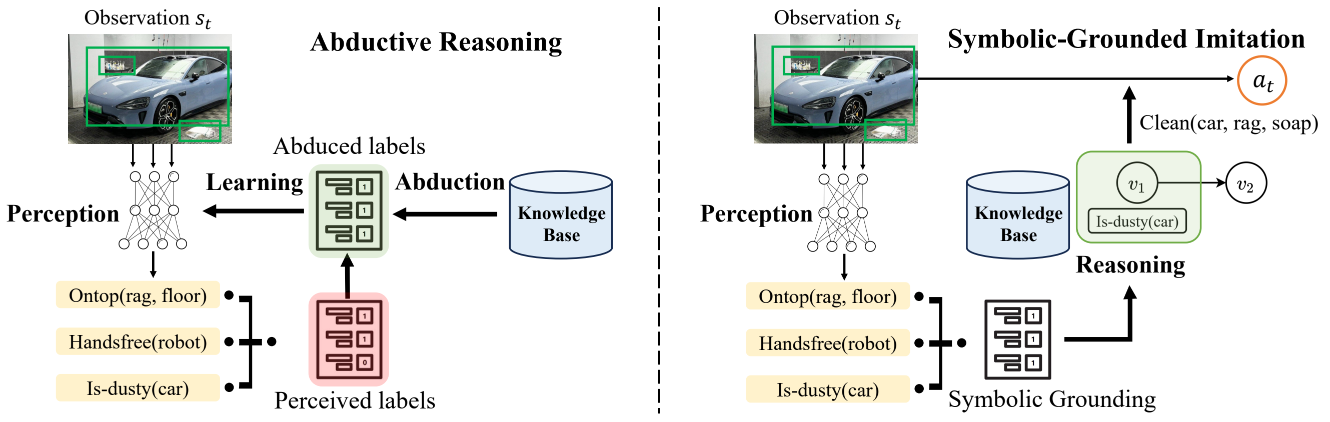


Figure 3: The Framework of our Abductive Imitation Learning.

required to be solved. The final configuration arises when there is bidirectional pointing between v_1 and v_2, indicating that both subtasks are mandatory to be solved. Utilizing the state machine, a symbolic planning solution can be derived via algorithms such as A^* search or dynamic programming, represented as a sequence of states and transitions: $\{(v_0, \overline{op}_0), (v_1, \overline{op}_1) \ldots (v_K, \overline{op}_K)\}$.

3.2 Abductive Reasoning

The ABIL framework can be roughly divided into abductive learning with the state machine and imitation with symbolic reasoning. The overall framework is illustrated and summarized in Figure 3 and Algorithm 1, respectively.

Given the state machine and expert demonstrations, the challenge lies in establishing the perception function f from observation to symbolic grounding when symbolic supervision of $\{s_t\}$ is not available. To address this challenge, we introduce abductive reasoning to provide pseudo labels derived from the state machine's knowledge, which could be taken to optimize the perception function f.

$$\min_f \sum_{s_i \in D} \sum_{t=1}^{T} \mathcal{L}(f(s_i^t), \widehat{z_i^t}),$$
$$\{\widehat{z_i^t}\}_{t=1}^T = \arg\min_{\{z_i^t\}_{t=1}^T} \sum \|z_i^t - f(s_i^t)\|^2, \quad \text{s.t.} \{z_i^t\}_{t=1}^T \models G \tag{1}$$

Specifically, for the different structures in the state machine, we could derive the sequential abduction: $\{z_i^t\}_{t=1}^T \models G$ as:

- The task has been completed at the end of the expert demonstration sequence. The symbolic state z_i^T satisfies the final goal g, that is, $z_i^T \models g$.
- Task a should be accomplished before b. The symbolic sequence $\{z_i^t\}_{t=1}^T$ will satisfy $\{z_i^t\}_{t=1}^j \models a, \{z_i^t\}_{t=j+1}^T \models b, \exists j \in (1, T)$ where the demonstrations could be divided into the two sequential part of accomplishing a and b.
- The agent should either complete a or b. The symbolic sequence $\{z_i^t\}_{t=1}^T$ will satisfy $\{z_i^t\}_{t=1}^T \models a$ or $\{z_i^t\}_{t=1}^T \models b$.
- The agent should complete both t1 and t2, but in any order. The symbolic sequence $\{z_i^t\}_{t=1}^T$ will satisfy $\{z_i^t\}_{t=1}^j \models a, \{z_i^t\}_{t=j+1}^T \models b, \exists j \in (1, T)$ or $\{z_i^t\}_{t=1}^j \models b, \{z_i^t\}_{t=j+1}^T \models a, \exists j \in (1, T)$.

Algorithm 1 Abductive Imitation Learning

```
Require: Demonstration dataset $D$, symbolic knowledge $G$.
    Number of learning rounds $N_R$ and $N_I$.
 1: for $t = 1$ to $N_R$ do
 2:     Get the perceived labels via $f(s)$
 3:     Get the abduced labels via Eq. 1.
 4:     Update the perception network $f$.
 5: end for
 6: for $t = 1$ to $N_I$ do
 7:     Get the symbolic states via $f(s)$
 8:     Get the logical operator $\bar{op}$ via Eq. 2.
 9:     Update the behavior network $h_{\bar{op}}$ via Eq. 4.
10: end for
11: return Perception $f$ and behavior $\{h_{\bar{op}}\}, \bar{op} \in \mathcal{OP}$.
```

With the sequential abduction, the pseudo label $\hat{z}$ in Equation 1 could be obtained and the perception module f could be optimized. Nevertheless, the perception module f plays an important role in symbolic grounding, which is crucial for the subsequent reasoning. Following [15, 24, 29, 30], we train the perception module f at the object level. For each predicate p, we have a predicate model f_p with the object-level features o as input, which could be obtained via an object detection model in practice. A ground atom $p(o1, o2)$ could be inferred by: $prob(p(o_1, o_2)|s) = f_p([o_1, o_2])$.

As summarized in the left part of Figure 3, our perception module f is optimized via the abductive reasoning. Equation 1 generates the pseudo labels based on the sequential consistency between the perception output and the solution of z satisfying the knowledge base. Unlike previous works [33, 35], it does not rely on the symbolic-level annotations of each observation in demonstrations, which are usually costly and difficult to obtain.

3.3 Symbolic-grounded Imitation

As a human being, one often consciously knows what he or she is doing, such as making the action of turning left because they realize the destination is on the left. In this part, we thus incorporate the reasoning of high-level operators into the original imitation learning process, regarding the symbolic operators as an assistance signals. Specifically, we first build the behavioral actor for each logical operator h_{op}, e.g. h_{clean} and h_{put}. Then we derive the desired behavior module by the symbolic states output of perception f and the corresponding abstract logical operator. Given the solution of symbolic planning: $\{(v_0, \overline{op}_0), (v_1, \overline{op}_1) \dots (v_K, \overline{op}_K)\}$.

$$\overline{op}^t = \overline{op}_k, \text{s.t.} f(s^t) \models v_k, \exists k \in [0, K] \quad (2)$$

The desired parameter of the operator $\overline{op}^t$, e.g., which object will be picked, could be reasoning as:

$$o^t = obj(\overline{op}^t) \quad (3)$$

Then the learning of behavior actors could be formulated as:

$$\min_h \sum_{s_i, a_i \in D} \sum_{t=1}^{T} \mathcal{L}(h_{\overline{op}_i^t}(s_i^t, o^t), a_i^t) \quad (4)$$

As summarized in the right part of Figure 3, our behavioral actors, referring to the human model of cognition before decision-making, embed high-level logical reasoning into the imitation learning process. The behavior ensemble $h_{\bar{op}}$ learns from experience through imitation, without relying on a pre-existing perfect controller to reach each sub-goal. Importantly, by utilizing the generalization capability of symbolic planning, the proposed actors can decompose diverse observations into symbolic states, facilitating reliable decision-making.

4 EMPIRICAL STUDY

We evaluate our proposal in three environments, including two neuro-symbolic benchmarks: BabyAI [6], Mini-BEHAVIOR [20], and a robotic manipulation benchmark, CLIPort [28]. We compare our method with three baselines: Behavior Cloning (**BC**) [2], Decision Transformer (**DT**) [5] and **PDSketch** [24]. For a fair comparison, all of these methods use the same network architecture which is based on the Neural Logic Machine (NLM) [8]. Specifically, we first encode the state with a two-layer NLM. For BC, we use a single linear layer, taking the state embedding as the input and output actions. For DT, we build a single transformer layer following the two-layer encoder, with the causal mask to generate future action with past states and actions. For PDSketch, we choose the full mode in the original paper [24], which provides sufficient prior knowledge of the symbolic transition, keeping consistency with our symbolic state machine. Following [24, 28], we report the percentage of successful planning for the desired goals, which are averaged over 100 evaluations under three random seeds.

4.1 Evaluation on BabyAI

BabyAI provides a benchmark for grounding logical instructions where an agent needs to follow instructions for a series of tasks like picking up objects, or unlocking doors. Following [24, 35], we consider 5 tasks: $\{goto, pickup, open, put, unlock\}$, and conduct the generalization evaluation with different numbers of objects in the testing environments. For example, the training environments contain 4 objects or 4 doors, and testing environments contain 8 objects or 8 doors. It simulates the challenges to the generalization of imitation learning, where household robots in open environments need to complete tasks in differently arranged rooms.

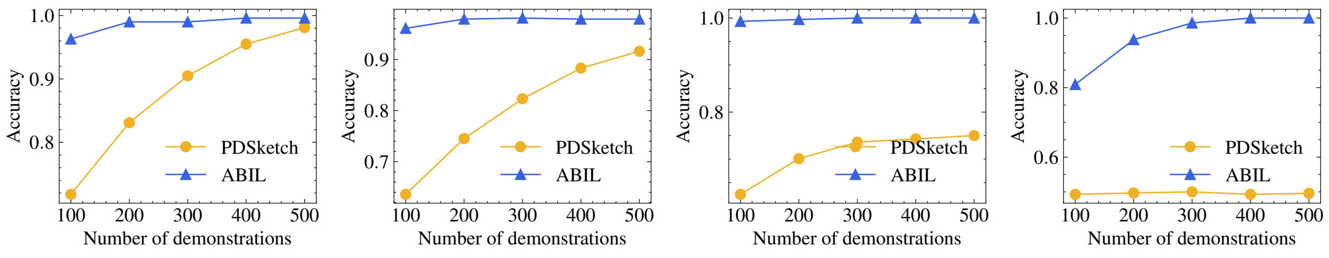
Results and Analysis. From Table 1, we could find that BC baseline is efficient in the simple *GotoSingle* task but significantly deteriorates on complex tasks and their generalization evaluation. PDSketch exhibits favorable performance in tasks that require few actions to complete and generalizes well in the case of increasing object number. Nevertheless, it struggles to solve long-horizon tasks like *Put* and *Unlock*, where experts' demonstrations require 9 or 10 actions to accomplish. One plausible reason is that, as a model-based method, PDSketch faces the accumulation of search errors, thus fails in long-horizon tasks. DT excels in handling tasks with a sequential nature (e.g. *Put* and *Unlock*). However, in short-horizon tasks, such as *Goto* and *Pickup*, DT performs weaker compared to BC. The possible reason could be its relatively high model complexity contradicts with limited data, making it difficult to learn efficiently from short-horizon demonstrations. In contrast, ABIL achieves competitive performance compared to PDSketch and has made significant improvements in long-horizon tasks. It is worth noting that ABIL exhibits stable generalization in novel environments, which supports our intention of using symbolic grounding to assist the generalization of imitation learning.

Comparison of Neuro-Symbolic Grounding. Like previous neuro-symbolic methods, the efficiency of neuro-symbolic grounding is the key to determining whether successful or not. In Figure 4, we compare the accuracy of the predicates learned by ABIL and PDSketch, under varying demonstration budgets. We found that in the relatively simple task, PDSketch can achieve over 90% predicate accuracy with 500 demonstrations. However, in more challenging tasks, such as open and unlock, its neuro-symbolic grounding ability is quite poor, which also leads to unsatisfactory performance in Table 1. In contrast, our ABIL not only achieves reliable neuro-symbolic grounding on all tasks (with nearly perfect predicate accuracy), but it's also more data efficient than PDSketch, requiring less than 20% of their data to achieve superior neuro-symbolic grounding results. It clearly indicates the advantages of our abductive reasoning on neuro-symbolic understanding.

Comparison of Efficiency. Learning-based methods BC and DT could promptly provide responses in the inference phase because they do not give adequate attention to the subsequent considerations. The model-based planning method PDSketch needs to search for a whole sequence of actions that can achieve the goal, which can be time-consuming, especially in cases with longer sequences and a multitude of available actions. Our approach integrates higher-level reasoning into the foundation of lower-level perception. By integrating symbolic-based planning, our approach enhances planning effectiveness, significantly reducing time consumption compared to PDSketch, which requires searching in the original observation-action space. While gaining advantages of logical reasoning in long-horizon goals, ABIL maintains the inference efficiency of the learning-based method.

Table 1: Success rates@100 in the BabyAI benchmark. The best result in each setting is bold and the second is underlined.

Task (Averaged Length)		Eval	BC	DT	PDSketch	ABIL-BC	ABIL-DT
GotoSingle	(3)	Basic	**1.00**	0.893 ± 0.049	**1.00**	**1.00**	0.963 ± 0.047
Goto	(3)	Basic	0.843 ± 0.006	0.720 ± 0.044	**1.00**	0.900 ± 0.046	0.900 ± 0.020
		Gen	0.743 ± 0.045	0.583 ± 0.049	**1.00**	0.777 ± 0.032	0.793 ± 0.029
Pickup	(4)	Basic	0.723 ± 0.031	0.490 ± 0.040	**0.990 ± 0.010**	0.847 ± 0.025	0.845 ± 0.035
		Gen	0.533 ± 0.031	0.320 ± 0.070	**0.973 ± 0.012**	0.730 ± 0.010	0.763 ± 0.051
Open	(6)	Basic	0.933 ± 0.025	0.493 ± 0.059	**1.00**	0.963 ± 0.021	0.923 ± 0.031
		Gen	0.877 ± 0.015	0.440 ± 0.078	**1.00**	0.927 ± 0.032	0.857 ± 0.055
Put	(9)	Basic	0.950 ± 0.044	0.910 ± 0.036	0.697 ± 0.021	0.940 ± 0.026	**0.953 ± 0.006**
		Gen	0.260 ± 0.036	0.380 ± 0.026	0.417 ± 0.025	**0.637 ± 0.064**	0.593 ± 0.084
Unlock	(10)	Basic	0.957 ± 0.012	0.990 ± 0.010	0.293 ± 0.051	0.967 ± 0.023	**0.993 ± 0.012**
		Gen	0.910 ± 0.030	0.990 ± 0.010	0.247 ± 0.051	0.963 ± 0.006	**0.993 ± 0.012**
Averaged time per evaluation			0.174 seconds	0.260 seconds	8.170 seconds	0.320 seconds	0.354 seconds



(a) Goto (b) Pickup (c) Open (d) Unlock

Figure 4: Accuracy of neuro-symbolic grounding under varying data budgets.

4.2 Evaluation on Mini-BEHAVIOR

Mini-BEHAVIOR is a recently proposed benchmark for embodied AI. It contains varying 3D household tasks chosen from the BEHAVIOR benchmark, including *Sorting Books*, *Making Tea*, *Cleaning A Car*, and so on. Most tasks are long-horizon and heterogeneous, some of which require more than one hundred decision-making steps to be completed. There are hundreds of different types and plenty of predicates which is challenging for neuro-symbolic grounding. In this domain, our state machine is mainly composed of several typical categories. For tasks mainly about tidying up the room, we split the primitive actions into op_{pick} and op_{place}, which is required to perform an action sequence to finish picking or placing sub-tasks. Combined with our symbolic-grounding model f, the agent will be able to distinguish when and where to pick and place. For tasks mainly about cleaning, we split the primitive actions into op_{clean} and op_{put}, which is required to finish washing or putting sub-tasks. In the generalization evaluation, we challenge the agents in environments with distractor objects that are unseen at the training phase.

Results and Analysis. The results on Mini-BEHAVIOR are provided in Table 2. In some simple short-horizon tasks, such as *Installing a printer* and *Opening packages*, BC shows satisfactory performance. Nevertheless, as the desired decision sequence grows, errors made by BC gradually accumulate, leading to an increasing deviation from the correct solution. This becomes particularly critical in the presence of disturbances or interferences, leading to a significant degradation on the generalization evaluation. Compared to BC, DT has achieved better performance in most tasks, however, it still performs poorly in generalization test, pointing out its vulnerability to environmental changes. In this benchmark, PDSketch failed to finish most tasks in the given time budget, due to the significant search depth required to achieve the goal. This highlights the limitations of model-based planning methods in long-horizon scenarios. Ensuring the learned transition remains accurate after numerous decision steps is challenging, making it difficult to provide a successful termination signal for the search. In contrast, ABIL performs reasoning at the symbolic level. Even if cleaning a car requires about 45 decision steps to complete, from the perspective of abstract operations, we can understand that we need to put the rag and soap back into the buckets after using them. This is similar to human's behavior, where we first recognize what the logical goal to be completed is, and then achieve it step by step through actions, rather than considering the impact of each limb movement, which would make the entire reasoning planning path too long. In this way, our neuro-symbolic method ABIL successfully incorporates logic-based reasoning into imitation learning, achieving competitive results and showing good generalization under environmental change.

Further Analysis with Varying Horizons. As we discussed above, the generalization of imitation learning methods is closely

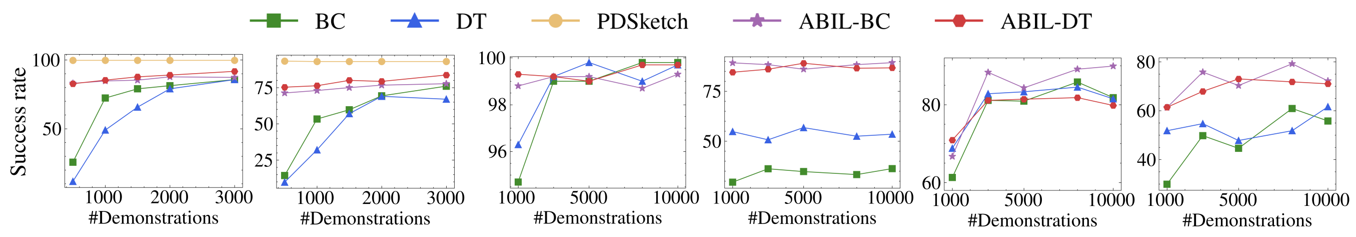
Table 2: Success rates@100 in the Mini-BEHAVIOR benchmark. The best result in each setting is bold.

Task	(Averaged Length)	Eval	BC	DT	PDSketch	ABIL-BC	ABIL-DT
Installing a printer	(10)	Basic	0.903 ± 0.023	0.927 ± 0.021	0.343 ± 0.032	0.920 ± 0.035	**0.947 ± 0.012**
		Gen	0.003 ± 0.006	0.300 ± 0.147	0.310 ± 0.046	0.577 ± 0.167	**0.760 ± 0.010**
Opening packages	(19)	Basic	0.947 ± 0.045	0.963 ± 0.034	0.020 ± 0.010	0.988 ± 0.016	**0.993 ± 0.008**
		Gen	0.295 ± 0.180	0.548 ± 0.065	0.020 ± 0.010	**0.892 ± 0.043**	0.845 ± 0.071
Making tea	(36)	Basic	0.607 ± 0.015	0.583 ± 0.105	> 5 minutes	0.613 ± 0.032	**0.623 ± 0.012**
		Gen	0.070 ± 0.078	0.113 ± 0.105		0.074 ± 0.053	**0.487 ± 0.049**
Moving boxes to storage	(38)	Basic	0.783 ± 0.061	0.787 ± 0.060	> 5 minutes	0.803 ± 0.031	**0.807 ± 0.038**
		Gen	0.433 ± 0.470	0.613 ± 0.047		0.717 ± 0.049	**0.753 ± 0.038**
Cleaning A Car	(45)	Basic	0.417 ± 0.047	0.313 ± 0.091	> 5 minutes	**0.420 ± 0.036**	0.340 ± 0.090
		Gen	0.170 ± 0.036	0.147 ± 0.083		0.183 ± 0.104	**0.220 ± 0.017**
Throwing away leftovers	(46)	Basic	0.833 ± 0.080	0.890 ± 0.029	> 5 minutes	**0.941 ± 0.030**	0.855 ± 0.059
		Gen	0.222 ± 0.167	0.653 ± 0.039		0.488 ± 0.147	**0.680 ± 0.039**
Putting away dishes	(65)	Basic	0.811 ± 0.031	0.828 ± 0.052	> 5 minutes	**0.872 ± 0.024**	0.811 ± 0.018
		Gen	0.141 ± 0.111	0.547 ± 0.296		**0.758 ± 0.118**	0.679 ± 0.108
Sorting books	(66)	Basic	0.601 ± 0.032	0.543 ± 0.053	> 5 minutes	**0.672 ± 0.084**	0.575 ± 0.011
		Gen	0.131 ± 0.047	0.220 ± 0.010		**0.360 ± 0.037**	0.333 ± 0.048
Laying wood floors	(68)	Basic	0.616 ± 0.062	0.638 ± 0.027	> 5 minutes	0.641 ± 0.054	**0.643 ± 0.036**
		Gen	0.068 ± 0.018	0.366 ± 0.041		0.225 ± 0.134	**0.435 ± 0.067**
Watering houseplants	(68)	Basic	0.814 ± 0.034	0.806 ± 0.020	> 5 minutes	**0.824 ± 0.023**	0.812 ± 0.034
		Gen	0.002 ± 0.004	0.187 ± 0.113		0.197 ± 0.095	**0.409 ± 0.151**
Cleaning shoes	(78)	Basic	0.482 ± 0.086	0.427 ± 0.042	> 5 minutes	**0.623 ± 0.033**	0.505 ± 0.075
		Gen	0.030 ± 0.005	0.053 ± 0.046		**0.215 ± 0.106**	0.193 ± 0.120
Collect misplaced items	(86)	Basic	0.460 ± 0.030	0.299 ± 0.015	> 5 minutes	**0.577 ± 0.053**	0.421 ± 0.042
		Gen	**0.325 ± 0.074**	0.261 ± 0.023		0.270 ± 0.062	0.279 ± 0.020
Organizing file cabinet	(106)	Basic	0.156 ± 0.047	0.522 ± 0.067	> 5 minutes	0.166 ± 0.028	**0.596 ± 0.058**
		Gen	0.083 ± 0.012	0.382 ± 0.112		0.109 ± 0.032	**0.490 ± 0.078**
Averaged time per evaluation			1.48 seconds	2.09 seconds	> 5 minutes	2.88 seconds	2.98 seconds

Table 3: Success rates@100 in the CLIPort benchmark. The best result in each setting is bold.

Task	Packing-5shapes	Packing-20shapes	Placing-red-in-green	Putting-blocks-in-bowls	Separating-20piles	Assembling-kits
BC	0.883 ± 0.025	0.207 ± 0.006	0.840 ± 0.031	0.507 ± 0.030	0.226 ± 0.017	0.187 ± 0.015
DT	0.913 ± 0.046	0.180 ± 0.026	0.846 ± 0.024	0.539 ± 0.068	0.250 ± 0.048	0.177 ± 0.023
ABIL-BC	**0.983 ± 0.015**	**0.940 ± 0.030**	0.988 ± 0.014	**0.962 ± 0.012**	0.305 ± 0.011	**0.829 ± 0.008**
ABIL-DT	0.977 ± 0.021	0.857 ± 0.025	**0.989 ± 0.017**	0.917 ± 0.033	**0.382 ± 0.029**	0.809 ± 0.008

related to the length of the task's horizon, especially in long-horizon tasks, where performance degradation is prone to occur. Mini-BEHAVIOR, which contains tasks that require different decision steps to complete, provides an appropriate observation window from this perspective. As shown in Table 2, the number of expert demonstrations required for these tasks ranges from 10 to 106 steps. On the one hand, we find that the increase in decision-making length required by the task indeed makes it more challenging, leading to a decline in the performance of almost all methods. On the other hand, we observe that the increase in decision-making length also amplifies performance difference in the generalization evaluation of the baseline method compared to the basic evaluation. Our method demonstrates good generalization performance, aligning with the basic evaluation, and showcasing the potential of the ABIL in open environments.



(a) Pickup (Basic/Gen) **(b) Opening Packages(Basic/Gen)** **(c) Putting dishes(Basic/Gen)**

Figure 5: Results under varying data budgets.

Table 4: Results on Zero-Shot Generalization tasks.

Domain	BabyAI			Mini-BEHAVIOR			Robotic Manipulation	
	Train		Eval	Train	Eval		Train	Eval
Task	Pickup	Open	Unlock	Throw 1	Throw 2	Throw 3	Putting-blocks	Novel combination
BC	0.760 ± 0.056	0.983 ± 0.021	0.120 ± 0.010	0.703 ± 0.085	0.117 ± 0.070	0.053 ± 0.045	0.597 ± 0.032	0.537 ± 0.095
DT	0.783 ± 0.031	0.957 ± 0.031	0.057 ± 0.051	0.770 ± 0.026	0.182 ± 0.008	0.056 ± 0.003	0.550 ± 0.019	0.424 ± 0.008
PDSketch	**0.970** ± **0.010**	0.990 ± 0.010	0.127 ± 0.021	0.013 ± 0.006	> 5 minutes	> 5 minutes	-	-
ABIL-BC	0.937 ± 0.021	**1.00**	0.980 ± 0.026	0.717 ± 0.055	0.590 ± 0.013	0.485 ± 0.054	**0.860** ± **0.027**	**0.857** ± **0.044**
ABIL-DT	0.925 ± 0.007	**1.00**	**0.993** ± **0.012**	**0.783** ± **0.031**	**0.702** ± **0.034**	**0.585** ± **0.120**	0.822 ± 0.026	0.823 ± 0.050

4.3 Evaluation on Robotic Manipulation

We further evaluate the proposal on the CLIPort [28] with 3D robotic manipulation tasks. In this environment, an agent needs to learn how to transport some objects and solve complex manipulation tasks based on visual observation. Every manipulation task needs to be achieved via a two-step primitive where each action involves a start and end-effector pose. Following [14, 24], we represent each object with its image cropped from the observation with its pose, which could be completed by an external detection module. All demonstrations are collected using handcrafted oracle policies following CLIPort [28], containing only successful trajectories. Since PDSketch mainly targets discrete actions, in this environment, we compared ABIL with BC and DT baselines. The results are provided in Table 3. In the *Packing-5shapes* task, BC and DT show satisfactory performance. However, it is evident that as the number of objects needed for operation increases, or when the task evolves from simple pick-and-place to more complex activities like sorting and assembling kits, the performance of pure-learning-based methods declines noticeably. The results of ABIL indicate it is capable of reasoning and planning how to appropriately allocate blocks to different containers to achieve the goal. Experimental results further support the effectiveness of ABIL. Due to the limited space, more discussion is provided in Appendix B.2.

4.4 Data Efficiency

Data efficiency is important for imitation learning, especially in robotic tasks where expert demonstrations are usually expensive and scarce. In this subsection, we conduct experiments across varying sizes of expert demonstrations to evaluate the data efficiency.

The results are provided in Figure 5. First, we found that PDSketch, the model-based planning method, achieved the best results on the *pickup* task, even with a limited 500 demonstrations. However, as shown in Table 2, PDSketch fails in the complex long-horizon tasks. Second, we could find that in the simple task, *pickup*, the generalization of different methods is consistently improved when the data volume increases. However, in the *Opening packages* and *Putting away dishes* tasks from Mini-BEHAVIOR, although the results in basic evaluation improve with the increase of data volume, their performance in the corresponding generalization tests no longer grows. This also reflects the weakness of pure-learning-based methods, that is, they easily overfit to the specific training observations, and their performance in out-of-distribution observation is fragile. Third, we found that our abductive imitation learning framework has clearly improved the data efficiency of the BC and DT baselines, especially achieved significant generalization improvement in the out-of-distribution evaluation.

4.5 Zero-Shot Generalization

Symbolic reasoning excels at generalization, especially ensuring the correctness of reasoning for any combination of logical clauses. In this subsection, we evaluate the zero-shot generalization performance in the composed tasks. In the BabyAI domain, we train the policies on the *pickup* and *open* task, then test them on the composed task *unlock*. During training, the demonstrations from two tasks are mixed and learning in a multi-task scheme. In the Mini-BEHAVIOR domain, we primarily concentrate on generalization with the longer series of events, which demands the agent to make use of learned techniques for repeatedly completing a single task to achieve the desired goal. Take the *Throwing away leftovers* task as an example, we train every model in the environment with 1 leftover hamburger to throw, while in the test environment, the agent is required to throw 2 or 3 hamburgers. In robotic manipulation, we primarily focus on compositional generalization with the novel combination of goals, which demands the agent to re-combine learned concepts to achieve, as shown in Figure 1(c).

The results are provided in Table 4. Although all baselines achieve satisfactory performance in training tasks (*pickup* and *open*), they degrade on the simple combined task (*unlock*). The pure-learning-based methods directly learn the action corresponding to the observations, lacking reasoning ability, thus unable to realize the need to first pick up a key that can open the target door, resulting in failure. PDSketch has reasoning ability, but its model-based planning solution accumulates errors with the increasing length of the sequence, resulting in poor performance and high computational overhead. In tasks with longer sequences, such as throwing away leftovers, solutions cannot be found even after running out of time. Our ABIL not only performs high-level reasoning to know that sub-goals should be sequentially completed but also can zero-shot achieve the composed tasks.

5 CONCLUSION AND DISCUSSION

In this work, we present a novel framework ABductive Imitation Learning, which combines the benefits of data-driven learning and symbolic-based reasoning. Like most neuro-symbolic and abductive learning research, ABIL primarily focuses on scenarios where a symbolic solution is available and typically assumes the knowledge base is accurate and sufficient [7, 24]. We will try our best to explore the situation of incomplete or ambiguous in future work. Another limitation is the assumption that environments are deterministic and fully observable, which follows the existing research [24, 29, 30]. We will attempt to draw advanced POMDP techniques for stochastic and partially observed settings in the future. Despite its limitations, ABIL provides a timely solution to enhance long-horizon imitation learning. This is particularly valuable considering that existing imitation learning often struggles with long-horizon tasks and is vulnerable to environmental changes. It effectively incorporates symbolic knowledge, which is widely present in tasks like household duties, to aid in imitation. Experiments demonstrate that ABIL significantly enhances data efficiency and generalization across a series of tasks, indicating its potential as a promising solution for neuro-symbolic imitation.

REFERENCES

[1] Samy Badreddine, Artur S. d'Avila Garcez, Luciano Serafini, and Michael Spranger. 2022. Logic Tensor Networks. *Artif. Intell.* 303 (2022), 103649.

[2] Michael Bain and Claude Sammut. 1995. A Framework for Behavioural Cloning. In *Machine Intelligence.*

[3] Raunak P. Bhattacharyya, Blake Wulfe, Derek J. Phillips, Alex Kuefler, Jeremy Morton, Ransalu Senanayake, and Mykel J. Kochenderfer. 2023. Modeling Human Driving Behavior Through Generative Adversarial Imitation Learning. *IEEE Transactions on Intelligent Transportation Systems* 24, 3 (2023), 2874–2887.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020.* Virtual Event.

[5] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems.* 15084–15097.

[6] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2019. BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning. In *7th International Conference on Learning Representations.* New Orleans, LA.

[7] Wang-Zhou Dai, Qiu-Ling Xu, Yang Yu, and Zhi-Hua Zhou. 2019. Bridging Machine Learning and Logical Reasoning by Abductive Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019.* Vancouver, Canada, 2811–2822.

[8] Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. 2019. Neural Logic Machines. In *7th International Conference on Learning Representations.* New Orleans, LA.

[9] Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. 2019. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications* 3, 4 (2019), 362–369.

[10] Richard Fikes and Nils J. Nilsson. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2, 3/4 (1971), 189–208.

[11] Maria Fox and Derek Long. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research* 20 (2003), 61–124.

[12] En-Hao Gao, Yu-Xuan Huang, Wen-Chao Hu, Xin-Hao Zhu, and Wang-Zhou Dai. 2024. Knowledge-Enhanced Historical Document Segmentation and Recognition. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence.*

[13] Alfonso Emilio Gerevini. 2020. An Introduction to the Planning Domain Definition Language (PDDL): Book review. *Artificial Intelligence* 280 (2020), 103221.

[14] Joy Hsu, Jiayuan Mao, Joshua B. Tenenbaum, and Jiajun Wu. 2023. What's Left? Concept Grounding with Logic-Enhanced Foundation Models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023.* New Orleans, LA.

[15] De-An Huang, Danfei Xu, Yuke Zhu, Animesh Garg, Silvio Savarese, Li Fei-Fei, and Juan Carlos Niebles. 2019. Continuous Relaxation of Symbolic Planner for One-Shot Imitation Learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Macau, China, 2635–2642.

[16] Yu-Xuan Huang, Wang-Zhou Dai, Le-Wen Cai, Stephen H. Muggleton, and Yuan Jiang. 2021. Fast Abductive Learning by Similarity-based Consistency Optimization. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021.* Virtual Event, 26574–26584.

[17] Yu-Xuan Huang, Wang-Zhou Dai, Jian Yang, Le-Wen Cai, Shaofen Cheng, Ruizhang Huang, Yu-Feng Li, and Zhi-Hua Zhou. 2020. Semi-Supervised Abductive Learning and Its Application to Theft Judicial Sentencing. In *20th IEEE International Conference on Data Mining.* Sorrento, Italy, 1070–1075.

[18] Yu-Xuan Huang, Zequn Sun, Guangyao Li, Xiaobin Tian, Wang-Zhou Dai, Wei Hu, Yuan Jiang, and Zhi-Hua Zhou. 2023. Enabling Abductive Learning to Exploit Knowledge Graph. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence.* Macao, China, 3839–3847.

[19] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *Comput. Surveys* 50, 2 (2017).

[20] Emily Jin, Jiaheng Hu, Zhuoyi Huang, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, and Roberto Martín-Martín. 2023. Mini-BEHAVIOR: A Procedurally Generated Benchmark for Long-horizon Decision-Making in Embodied AI. *CoRR* abs/2310.01824 (2023).

[21] George Dimitri Konidaris, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. 2018. From Skills to Symbols: Learning Symbolic Representations for Abstract High-Level Planning. *Journal of Artificial Intelligence Research* 61 (2018), 215–289.

[22] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. 2022. BEHAVIOR-1K: A Benchmark for Embodied AI with 1, 000 Everyday Activities and Realistic Simulation. In *Conference on Robot Learning.* Auckland, New Zealand, 80–93.

[23] Zenan Li, Zehua Liu, Yuan Yao, Jingwei Xu, Taolue Chen, Xiaoxing Ma, and Jian Lü. 2023. Learning with Logical Constraints but without Shortcut Satisfaction. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.*

[24] Jiayuan Mao, Tomás Lozano-Pérez, Josh Tenenbaum, and Leslie Pack Kaelbling. 2022. PDSketch: Integrated Domain Programming, Learning, and Planning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022.* New Orleans, LA.

[25] Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. 2023. Not All Neuro-Symbolic Concepts Are Created Equal: Analysis and Mitigation of Reasoning Shortcuts. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.*

[26] Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. 2022. A Survey on Imitation Learning Techniques for End-to-End Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2022), 14128–14147.

[27] Nived Rajaraman, Lin F. Yang, Jiantao Jiao, and Kannan Ramchandran. 2020. Toward the Fundamental Limits of Imitation Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information*

Processing Systems 2020. Virtual Event.

[28] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2021. CLIPort: What and Where Pathways for Robotic Manipulation. In *Conference on Robot Learning*. London, UK, 894–906.

[29] Tom Silver, Ashay Athalye, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. 2022. Learning Neuro-Symbolic Skills for Bilevel Planning. In *Conference on Robot Learning*. Auckland, New Zealand, 701–714.

[30] Tom Silver, Rohan Chitnis, Nishanth Kumar, Willie McClinton, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Joshua B. Tenenbaum. 2023. Predicate Invention for Bilevel Planning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*. Washington, DC, 12120–12129.

[31] Tom Silver, Rohan Chitnis, Joshua B. Tenenbaum, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. 2021. Learning Symbolic Operators for Task and Motion Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Prague, Czech Republic, 3182–3189.

[32] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. 2023. MimicPlay: Long-Horizon Imitation Learning by Watching Human Play. In *7th Annual Conference on Robot Learning*. Atlanta, GA.

[33] Renhao Wang, Jiayuan Mao, Joy Hsu, Hang Zhao, Jiajun Wu, and Yang Gao. 2023. Programmatically Grounded, Compositionally Generalizable Robotic Manipulation. In *The Eleventh International Conference on Learning Representations*. Kigali, Rwanda.

[34] Grover J Whitehurst and Ross Vasta. 1975. Is language acquired through imitation? *Journal of Psycholinguistic Research* 4 (1975), 37–59.

[35] Danfei Xu, Roberto Martín-Martín, De-An Huang, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. 2019. Regression Planning Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*. Vancouver, Canada, 1317–1327.

[36] Tian Xu, Ziniu Li, and Yang Yu. 2022. Error Bounds of Imitating Policies and Environments for Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 10 (2022), 6968–6980.

[37] Xiaowen Yang, Jie-Jing Shao, Wei-Wei Tu, Yufeng Li, Wang-Zhou Dai, and Zhi-Hua Zhou. 2024. Safe Abductive Learning in the Presence of Inaccurate Rules. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*. Vancouver, Canada, 16361–16369.

[38] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 1039–1050.

[39] Zhi-Hua Zhou. 2019. Abductive learning: towards bridging machine learning and logical reasoning. *SCIENCE CHINA Information Science* 62, 7 (2019), 76101:1–76101:3.

A SUMMARY OF NOTATIONS

Table 5 summarizes the symbols used in this paper.

Table 5: Summary of Notations.

Notation	Meaning
$\mathcal{S}$	State space
$\mathcal{A}$	Action space
$\mathcal{T}$	Deterministic transition function
$\mathcal{O}$	Set of objects in the environment
$\mathcal{P}$	Set of predicate symbols
$\mathcal{OP}$	Set of operation predicate symbols
s_0	Initial state
g	Set of ground atoms representing the goal
$\overline{p}$	Ground atom
$obj(\overline{p})$	Function to retrieve objects from a ground atom
$s \models \overline{p}$	State s semantically entails the ground atom $\overline{p}$
$s \models P$	State s satisfies all ground atoms in set P
G	Directed graph representing the knowledge base
V	Set of vertices in the knowledge base graph
E	Set of edges in the knowledge base graph
$\overline{op}$	A ground atom representing a logical action
EFF^+	Add effect of a logical action
EFF^-	Delete effect of a logical action
$z \models G$	A trajectory z satisfies the knowledge base G

B EXPERIMENTAL DETAILS

B.1 BabyAI and Mini-BEHAVIOR

In these two environments, each object feature is represented by its state and position in the room, and the robot feature is represented by its position and direction. The state representation is composed of features of all objects and the robot. The action spaces are both discrete. All expert demonstrations are generated by scripts based on A^* search. In BabyAI, 1000 demonstrations were used for training per task and to obtain Table 1. In Mini-BEHAVIOR, *install-a-printer*, *opening packages*, and *moving boxes to storage* used 1000, while other tasks used 3000 expert demonstrations for training.
Model Architecture. For each predicate (e.g. is-dusty), we build a binary classifier, which takes a single object o as argument, and returns a scalar value from 0 to 1, indicating the classification score. All methods use the same network architecture which is based on the Neural Logic Machine [8]. Specifically, we first encode the state with a two-layer NLM. For BC, we use a single linear layer, taking the state embedding as the input and output actions. For DT, we build a single transformer layer following the two-layer encoder, with the causal mask to generate future action with past states and actions. For PDSketch, we choose the full mode in the original paper [24]. For ABIL-BC, we implement the behavior modules using BC model, and for ABIL-DT, we implement the behavior modules using DT model.

B.2 Robotic Manipulation

In this environment, each object is represented as a tuple of a 3D xyz location, and an image crop. Following [24], we first compute the 2D bounding box of the object in the camera plane, then crop the image patch and resize it to 24 by 24 to obtain the image crop. The action space is continuous, each action involves a start and end-effector pose. Table 6 provides the average length of expert demonstrations. This benchmark involves the agent manipulating objects of various colors and shapes, reflecting the requirements in the open world, providing a greater challenge for imitation learning. For each task, 1000 expert demonstrations were used for training and to obtain Table 3. All of these demonstrations are collected using oracle policies following CLIPort [28], containing only successful trajectories.

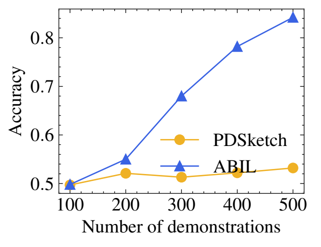
Model Architecture. Image feature of each object is a 64-dimensional embedding obtained via an image encoder, which is a 3-layer convolutional neural network followed by a linear transformation layer. For each predicate (e.g. is-red), we build a binary classifier, which takes the image feature of an object, and returns a scalar value from 0 to 1, indicating the classification score. The model implementation is same as in BabyAI and Mini-BEHAVIOR, except output continuous value as action.

Table 6: Details of CLIPort benchmark.

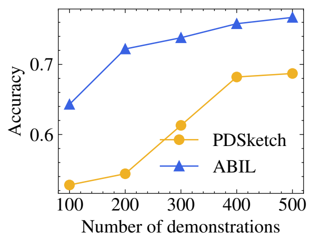
Task	Ave. Length	Evaluation
Packing-5shapes	1	4/7 unseen/total colors
Packing-20shapes	1	4/7 unseen/total colors
Placing-red-in-green	2	11 total colors
Putting-blocks-in-bowls	2	7 total colors
Assembling-kits	5	10/5 total shapes/colors
Separating-20piles	7	7 total colors

Further Analysis and Discussion The full experimental results are provided in Table 3. Although the execution length for robotic manipulation is shorter compared to household tasks in Mini-BEHAVIOR, the objects it needs to manipulate are more complex. As illustrated in Figure 1(b), in the packing-shapes task, the agent need to manipulate objects of the same shape but unseen colors during testing. In packing-20shapes, which experts can complete in one step, pure learning-based BC and DT only achieved a 20% success rate. However, our ABIL-BC achieved a satisfactory 94% success rate through neuro-symbolic grounding to recognize the shape of corresponding objects. These results highlight the vulnerability of pure-learning-based methods in open-world scenarios and demonstrate the necessity of introducing neuro-symbolic reasoning in our ABIL, which may provide a promising solution for household agents.

Comparison of Neuro-Symbolic Grounding. In robotic manipulation, the search based planning policy of PDSketch only applies to discrete symbolic operators, and therefore cannot be compared in our main experiments with continuous action space. We can only compare ABIL with PDSketch from the perspective of neuro-symbolic grounding. The results are provided in Figure 6. Achieving precise grounding in this environment is more challenging. As the amount of demonstration increases, the accuracy of PDSketch rises slowly and erratically. In contrast, our method shows a rapid improvement, further demonstrating the advantage of our ABIL in terms of data efficiency.



(a) Packing-20shapes



(b) Putting-blocks-in-bowls

Figure 6: Accuracy of neuro-symbolic grounding under varying data budgets in Robotic Manipulation tasks.

C SUPPLEMENTAL RESULTS

C.1 Study on Performance with Imperfect Symbolic Grounding

To evaluate the influences of neuro-symbolic errors upon ABIL, we further conduct experiments on the *Pickup* and *Putting-blocks-in-bowls* task . Experimental results are provided in Table 7.

Like human reasoning, incorrect logical objectives may lead to the failure of sequential decision-making. Neuro-symbolic errors indeed lower the performance of ABIL. Nevertheless, ABIL integrates data-driven imitation and logical objectives in learning, it has a tolerance for neuro-symbolic errors. Even under 75% grounding accuracy, it can still achieve improvements compared to purely data-driven methods.

C.2 Additional Results on zero-shot generalization

In Table 8, we provide additional evaluation results on zero-shot generalization task in the Mini-Behavior benchmark. In *Opening Packages* task, we train every model in the environment with 1 package to open, while in the testing environment, the agent is required to open 2 or 3 packages.

D RELIANCE ON KNOWLEDGE BASE

In this work, we mainly consider the case where a symbolic solution is available, which is consistent with neuro-symbolic learning and abductive learning, typically assuming the knowledge base is accurate [1, 7, 35, 38]. Please note that existing imitation learning struggles in handling long-horizon tasks. However, long-horizon tasks, such as household tasks in mini-behavior, are usually easier to define in symbolic space [20, 22]. In Appendix F, we further provide the details of the knowledge base used in the experiments, which supports the high-level symbolic solutions that are usually not complex and humans could relatively easily provide.

We also note that the quality and completeness of the symbolic knowledge base are crucial for the performance of ABIL. This is also a common feature of neuro-symbolic methods, that is, it may lead to neuro-symbolic reasoning shortcuts [23, 25]. To refine the incomplete symbolic knowledge base, several advanced strategies can be employed. For example, decomposing and transforming natural language instructions [33], using the prior knowledge of LLM to provide symbolic solutions [14], or adaptively using a noisy

Table 7: Results under varying gorunding accuracy.

Grounding accuracy			100%	90%	75%	50%	Data-driven
ABIL-BC	Pickup	Basic	0.847 ± 0.025 ↑	0.787 ± 0.015 ↑	0.763 ± 0.023 ↑	0.717 ± 0.021	0.723 ± 0.031 (BC)
		Gen	0.730 ± 0.010 ↑	0.667 ± 0.057 ↑	0.653 ± 0.023 ↑	0.560 ± 0.017 ↑	0.533 ± 0.031 (BC)
	Putting-blocks-in-bowls		-	0.962 ± 0.012 ↑	0.599 ± 0.036 ↑	0.512 ± 0.022 ↑	0.507 ± 0.030 (BC)
ABIL-DT	Pickup	Basic	0.845 ± 0.035 ↑	0.860 ± 0.017 ↑	0.800 ± 0.036 ↑	0.733 ± 0.025 ↑	0.490 ± 0.040 (DT)
		Gen	0.763 ± 0.051 ↑	0.740 ± 0.056 ↑	0.597 ± 0.042 ↑	0.563 ± 0.029 ↑	0.320 ± 0.070 (DT)
	Putting-blocks-in-bowls		-	0.917 ± 0.033 ↑	0.576 ± 0.041 ↑	0.462 ± 0.047	0.539 ± 0.068 (DT)

Table 8: Additional Results on Zero-Shot Generalization tasks.

Domain		Task	BC	DT	PDSketch	ABIL-BC	ABIL-DT
Mini-BEHAVIOR	Train	Opening 1 package	0.950 ± 0.087	**1.00**	0.467 ± 0.057	0.997 ± 0.006	**1.00**
	Eval	Opening 2 packages	0.012 ± 0.010	0.037 ± 0.025	0.020 ± 0.010	0.818 ± 0.014	**0.840 ± 0.035**
		Opening 3 packages	0.002 ± 0.004	0.024 ± 0.008	> 5 minutes	0.551 ± 0.032	**0.631 ± 0.041**

knowledge base [37]. This limitation may be alleviated through a combination of these approaches. We will take this as our future direction.

E REPRODUCIBILITY

To promote reproducibility, we have released the code in an anonymous repository[1]. This may also assist future research.

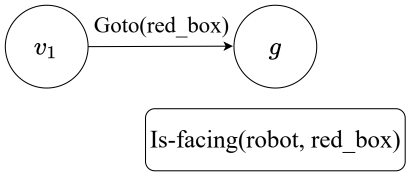
F DETAILS OF KNOWLEDGE BASE

In this section, we provide a detailed illustration of our knowledge base to help readers understand and reproduce.

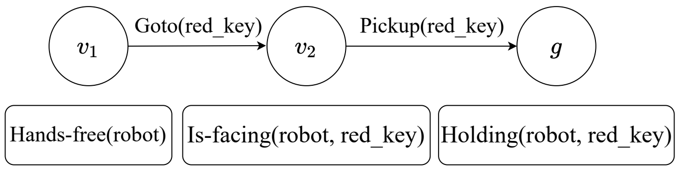
[1]https://anonymous.4open.science/r/ABIL_submission_KDD2025

F.1 BabyAI

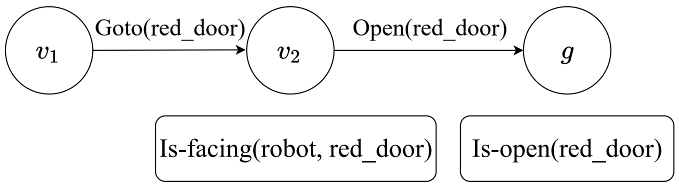
- Goto a red box



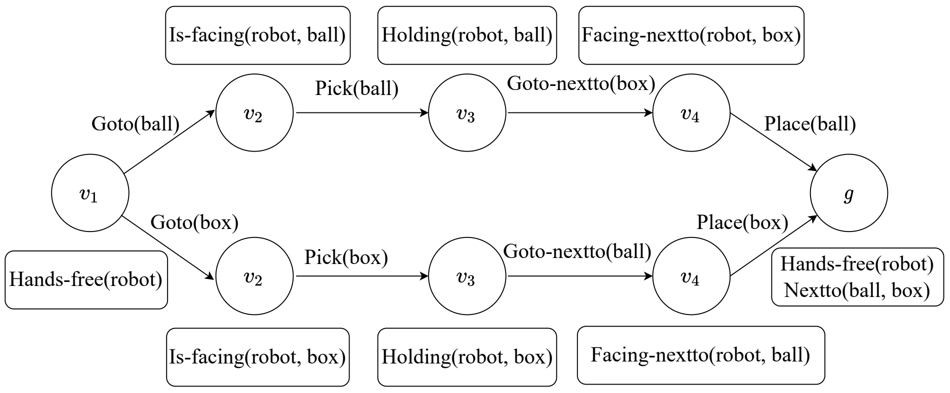
- Pickup a red key



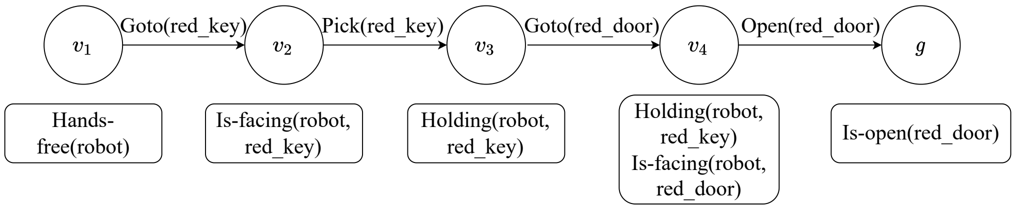
- Open a red door



- Put the ball next to the box



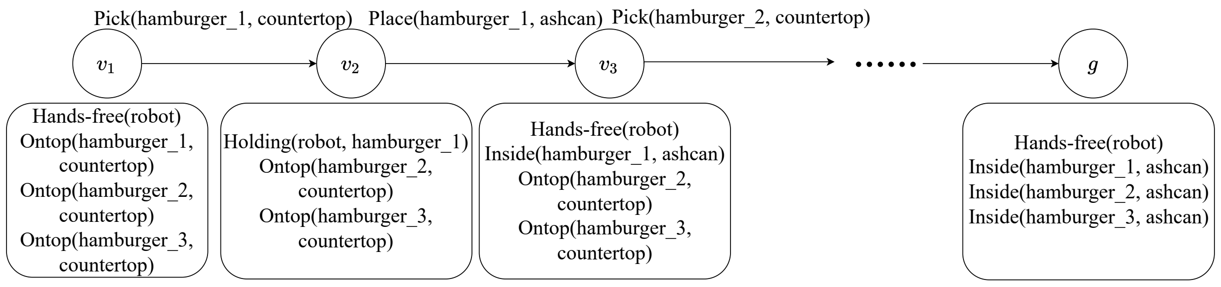
- Unlock a red door



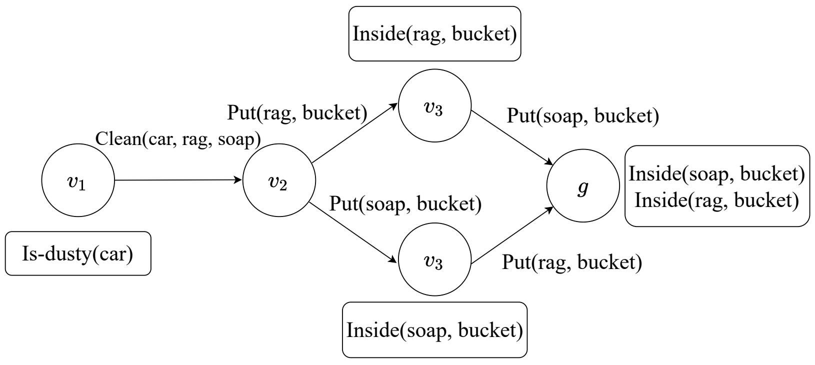
F.2 Mini-BEHAVIOR

In this domain, our state machine is mainly composed of several typical categories. For tasks mainly about tidying up the room, e.g. *Throwing away leftovers*, we split the primitive actions into op_{pick} and op_{place}, which is required to perform an action sequence to finish pickup or place subtask. Combined with our symbolic-grounding f, the agent will be able to distinguish when and where to pick and place. For tasks mainly about cleaning, e.g. *Cleaning a car*, we split the primitive actions into op_{clean} and op_{put}, which is required to finish washing or putting subtask. In addition, some tasks involve more operators, such as *install a printer*. We provide detailed illustrations of these representative state machine models.

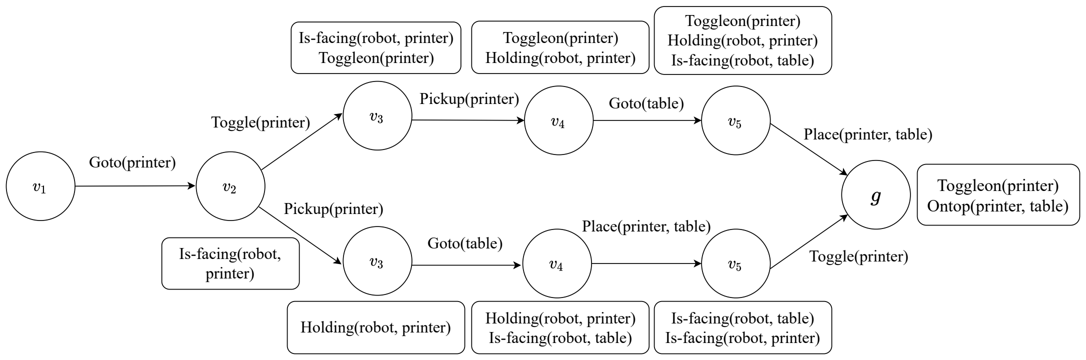
- Throwing away leftovers
  In this task, there are 3 hamburgers on plates, which are on a countertop in the kitchen. The agent must throw all of the hamburgers into the ashcan.


Pick(hamburger_1, countertop)
Place(hamburger_1, ashcan)
Pick(hamburger_2, countertop)
v_1
v_2
v_3
g
Hands-free(robot) Ontop(hamburger_1, countertop) Ontop(hamburger_2, countertop) Ontop(hamburger_3, countertop)
Holding(robot, hamburger_1) Ontop(hamburger_2, countertop) Ontop(hamburger_3, countertop)
Hands-free(robot) Inside(hamburger_1, ashcan) Ontop(hamburger_2, countertop) Ontop(hamburger_3, countertop)
Hands-free(robot) Inside(hamburger_1, ashcan) Inside(hamburger_2, ashcan) Inside(hamburger_3, ashcan)

- Cleaning a car
  In this task, initially there is a dusty car, a soap and unsoaked rag, the agent need to use the rag and soap to clean the car. Finally the agent should place the rag and soap in a bucket.

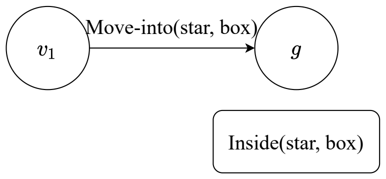


- Installing a printer
  In this task, initially there is a printer on the floor, and the agent must place it on the table and toggle it on.

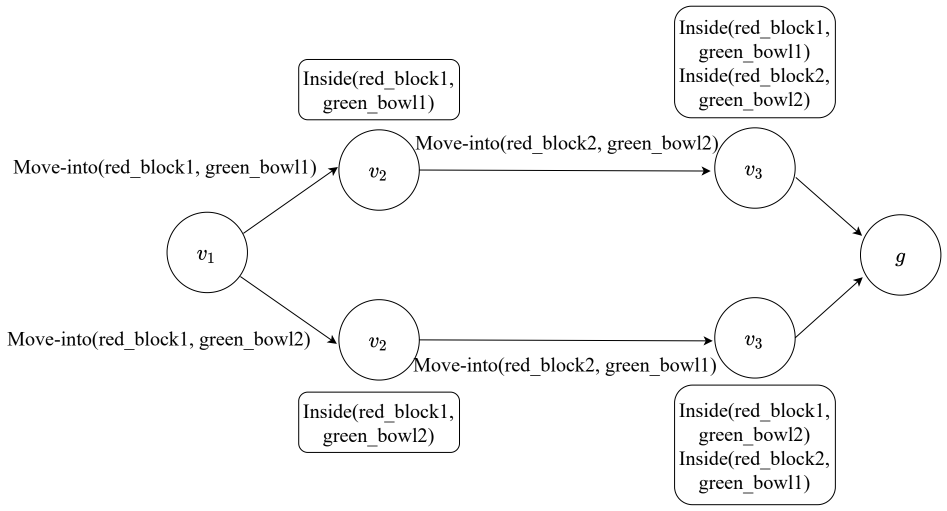

Is-facing(robot, printer) Toggleon(printer)
Toggleon(printer) Holding(robot, printer)
Toggleon(printer) Holding(robot, printer) Is-facing(robot, table)
Toggle(printer)
Pickup(printer)
Goto(table)
Place(printer, table)
Goto(printer)
v_1
v_2
v_3
v_4
v_5
g
Toggleon(printer) Ontop(printer, table)
Pickup(printer)
Is-facing(robot, printer)
Goto(table)
Place(printer, table)
Toggle(printer)
Holding(robot, printer)
Holding(robot, printer) Is-facing(robot, table)
Is-facing(robot, table) Is-facing(robot, printer)

F.3 Robotic Munipulation

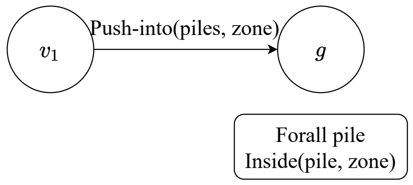
- Packing star into the box



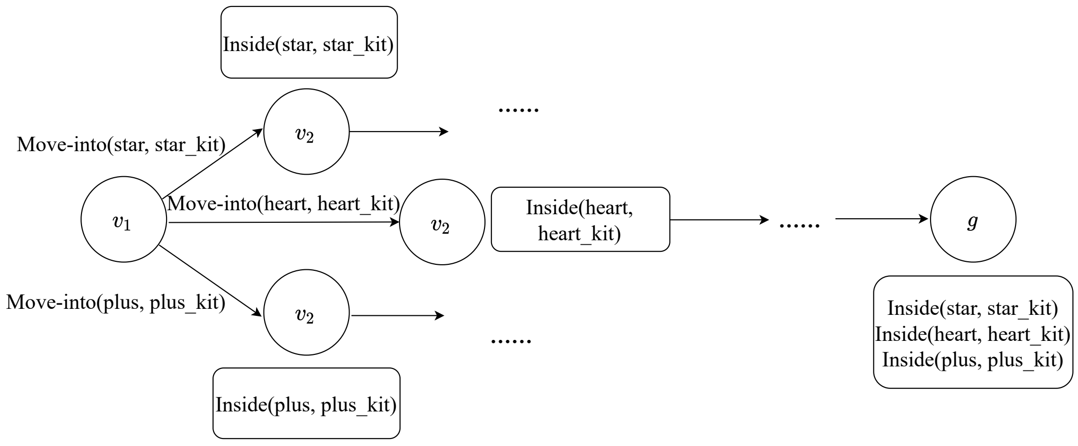
- Putting-red plocks-in-green bowls


Inside(red_block1, green_bowl1)
Inside(red_block1, green_bowl1)
Inside(red_block2, green_bowl2)
Move-into(red_block1, green_bowl1)
v_2
Move-into(red_block2, green_bowl2)
v_3
v_1
g
Move-into(red_block1, green_bowl2)
v_2
v_3
Move-into(red_block2, green_bowl1)
Inside(red_block1, green_bowl2)
Inside(red_block1, green_bowl2)
Inside(red_block2, green_bowl1)

- Separating-piles



- Assembling-kits


Inside(star, star_kit)
v_2
......
Move-into(star, star_kit)
v_1
Move-into(heart, heart_kit)
v_2
Inside(heart, heart_kit)
......
g
Move-into(plus, plus_kit)
v_2
......
Inside(star, star_kit)
Inside(heart, heart_kit)
Inside(plus, plus_kit)
Inside(plus, plus_kit)