

Robot Learning with a Spatial, Temporal, and Causal And-Or Graph

Caiming Xiong*, Nishant Shukla*, Wenlong Xiong, and Song-Chun Zhu

Abstract—We propose a stochastic graph-based framework for a robot to understand tasks from human demonstrations and perform them with feedback control. It unifies both knowledge representation and action planning in the same hierarchical data structure, allowing a robot to expand its spatial, temporal, and causal knowledge at varying levels of abstraction. The learning system can watch human demonstrations, generalize learned concepts, and perform tasks in new environments, across different robotic platforms. We show the success of our system by having a robot perform a cloth-folding task after watching few human demonstrations. The robot can accurately reproduce the learned skill, as well as generalize the task to other articles of clothing.

I. INTRODUCTION

Writing automated software on robots is not nearly as robust as that on traditional computers. This is due to the heavy burden of matching software assumptions to physical reality. The complexities and surprises of the real world require robots to adapt to new environments and learn new skills to remain useful.

In robot automation, implicit motor control is widely used for learning from human demonstrations [1] [2] [3]. However, implicit motor control is insufficient for generalizing robot execution. For instance, a robot can imitate a human’s demonstration to open a door; yet, it cannot execute a similar motion trajectory such as opening a window without the explicit representation of the task. Intuition such as how to rotate the joints of an arm is not something easily expressible, but rather learned through experiences. Uniting explicit and implicit knowledge allows immediate communication through natural language, as well as clear grounding of abstract concepts into atomic actions.

In this paper, we propose a unified framework to bridge the implicit motor control with explicit high-level knowledge so the robot can understand human behavior, perform a task with feedback control, and reason in vastly different environments. As a proof of concept, we teach a robot how to fold a shirt through few human demonstrations, and have it infer how to fold various articles of clothing, such as pants or towels. The same causality-learning framework can be extrapolated to arbitrary tasks, not just cloth-folding. Moreover, since explicit knowledge is structured graphically, our framework naturally allows for the merging, trimming, and addition of knowledge from various human demonstrations, all with feedback control. The high-level concepts are

human-understandable, so both the human and robot can communicate through this intermediate language [7]. Thus, programming the robot becomes an act of merely modifying a graph-based data structure.

The contributions of this paper include the following:

- Proposes a cross-platform stochastic framework for robots to ground human demonstrations into hierarchical spatial, temporal, and causal knowledge.
- Demonstrates a robot capable of learning, correcting its mistakes, and generalizing in a cloth-folding task from human demonstrations.
- Establishes the first system to use a non-rigid physical simulation to model the robot’s environment to improve task execution.
- Provides experimental evidence of our framework to generalize a cloth-folding task across different clothes and different robot platforms.

II. RELATED WORKS

While precisely grounding a human demonstration to atomic robot actions has been done in various forms [6] [12] [25], we instead focus on the novel representation and generalizability of tasks. Beetz et al. integrate robot knowledge representation into the perception processes as well, but our framework allows alternative planning generated by probabilistic sampling to match observed expectations. For example, there are multiple ways to fold a t-shirt, and each of these ways has its own likelihood. Our probabilistic learning framework resembles closest to the human-inspired Bayesian model of imitation by Rao et al. [20]. However, we instead emphasize the hierarchical and ever-changing nature of spatial, temporal, and causal concepts in the real world.

Autonomously folding clothes has been demonstrated in various works. Wang et al. [28] were able to successfully design a perception-based system to manipulate socks for laundry. Miller et al. [10] have demonstrated sophisticated cloth-folding robots, and Doumanoglou et al. [27] have made substantial progress in autonomously unfolding clothes, but our focus is on understanding how to perform arbitrary tasks. There are other systems [6] that also learn concrete action commands from small video clips, but unlike those, our design allows a modifiable grammar and our performance is measured on multi-step long-term actions. Furthermore, our solution to knowledge representation is more powerful than commonsense reasoning employed by first-order logic [18], since it takes advantage of the probabilistic models under ambiguous real-world perception.

Our work is based on the knowledge representation system incorporated by Tu et al. [11], augmented heavily into the

* C. Xiong and N. Shukla contributed equally to this work

C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu are with the Center for Vision, Cognition, Learning, and Autonomy (VCLA), University of California, Los Angeles caimingxiong@ucla.edu, nxs@ucla.edu, wenlongx@gmail.com, sczhu@stat.ucla.edu

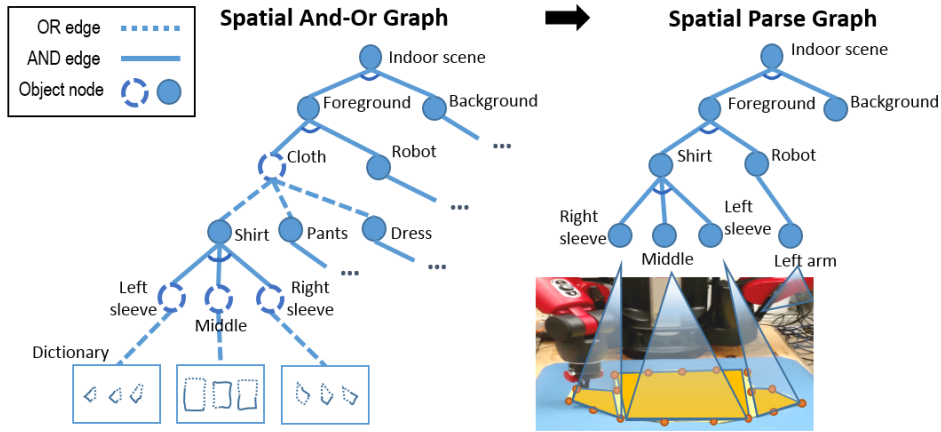


Fig. 1. The Spatial And-Or Graph on the left represents the ongoing perceptual knowledge of the world, i.e. a learned stochastic visual grammar. A specific instance of the And-Or graph is realized in the parse graph on the right.

robotics domain. We extend the learning of event And-Or grammars and semantics from video [4] to our real-time robotics framework. The And-Or graph encapsulates a conformant plan under partial observability, enabling an architecture that is cognitively penetrable since an updated belief of the world alters the robot’s behavior [13]. Unlike traditional graph planning [9], the hierarchical nature of the knowledge representation system enables a practical way of generating actions for a long-term goal.

III. METHOD

There is often a fine distinction between memorization and understanding, where the latter enables generalizing learned concepts. In order to understand a human task from demonstrations/videos such as cloth-folding, a knowledge representation system is necessary to ensure actions are not simply memorized. Four types of knowledge are important for understanding and generalizing:

- **Spatial knowledge** expresses the physical configuration of the environment when performing the task. For a cloth-folding task, a table, cloth, and each part of the cloth, such as the left and right sleeve of a shirt, needs to be detected.
- **Temporal knowledge** reveals the series of human actions in the process of the task. In cloth-folding, the hand motion, grip opening, and grip closing actions are essential. These actions combine together to form a fold action.
- **Casual knowledge** conveys the status change of an object in each dynamic human action. For example, a shirt may be folded in various ways, either by folding the left sleeve into the middle and then the right sleeve, or vice versa. Folding a cloth requires multiple hierarchical steps for reasoning.
- **The interplay between the spatial, temporal, and causal concepts** manifests a generalizable form of knowledge to be used in changing application domains. The robot must choose an action to achieve a state

change by using a causal reasoning concept. Each of the three must work together to express learned knowledge.

A. Mathematical Formulation for Human Task

Given a set of human task demonstrations $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ such as cloth-folding videos, the goal is to learn a joint model (G_{STC}) including Spatial, Temporal, and Causal concepts, that we formulate as

$$\begin{aligned}
 G_{STC}^* &= \underset{G_{STC}}{\operatorname{argmax}} P(G_{STC}|\mathcal{D}) \\
 &= P(G_S|\mathcal{D}) \cdot P(G_T|\mathcal{D}) \cdot P(G_C|\mathcal{D}) \\
 &\quad \cdot P(R(G_S, G_T, G_C)|\mathcal{D})
 \end{aligned} \tag{1}$$

where G_S is the model of spatial concepts, G_T is the model of temporal concepts, G_C is the model of casual concepts, and $R(G_S, G_T, G_C)$ is the relational/conditional model between spatial, temporal, casual concepts.

To implement this formulation, we need to define the concrete representation for each symbol in Eq. 1. Due to the structured and compositional nature of spatial, temporal, and causal concepts, we adopt the hierarchical stochastic grammar model, And-Or graph (AoG) [5], as the base of our model representation which is introduced below. To simplify the learning process, we marginalized the complex STC-AoG (G_{STC}) into the S-AoG (G_S), T-AoG (G_T) and C-AoG (G_C); thus, we can learn the G_S , G_T and G_C separately as the model’s initialization, then jointly learn the conditional model between them.

B. And-Or Graph Overview

The And-Or Graph is defined as a 3-tuple $\mathcal{G} = (V, R, P)$, where $V = V^{AND} \cup V^{OR} \cup V^T$ consists of a disjoint set of And-nodes, Or-nodes, and Terminal nodes respectively. R is a set of relations between Or-nodes or subgraphs, each of which represents a generating process from a parent node to its children nodes. $P(r)$ is an expansion probability for each relation.

Figure 1 shows an example of an And-Or graph. An And-node represents the decomposition of a graph into multiple

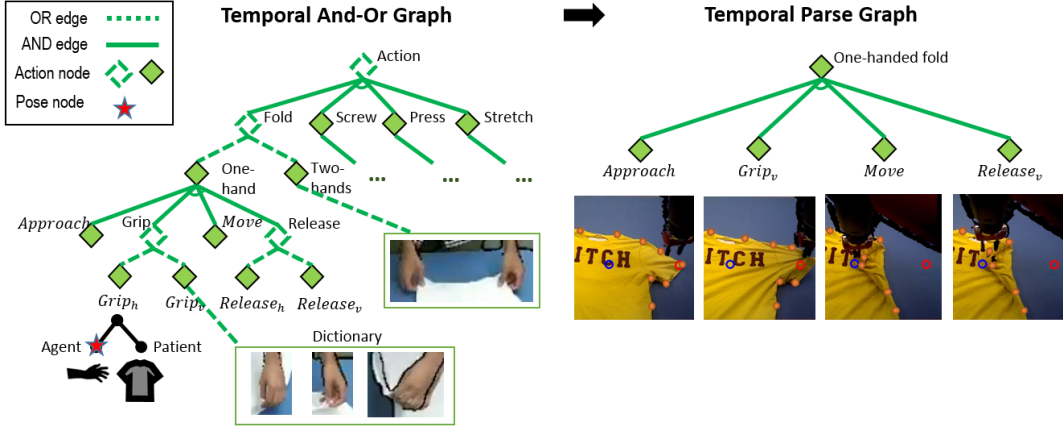


Fig. 2. The Temporal And-Or Graph on the left is a database of all actions currently known in the real world. Each action has an associated agent and patient. The realized parse graph on the right shows a generated sequence of actions directly executable by the robot.

sub-graphs. It is denoted by an opaque circle, and all the outgoing edges are opaque lines. An Or-node is a probabilistic switch deciding which of the sub-graphs to accept. It is denoted by an open circle with out-going edges drawn in dashed lines. The Terminal node represents grounded components, often referred to as a dictionary.

The nodes are structured into a hierarchical directed acyclic graph (DAG) structure. The AoG is a combination of a Markov tree and Markov random field, where an And-node corresponds to a graphic template model, and an Or-node corresponds to a switch in a Markov tree [16].

Given a set of human demonstrations \mathcal{D} , the graph \mathcal{G} is composed of an AoG graph structure $\hat{\mathcal{G}}$ and parameters θ . The nodes and rules/edges in the graph structure aim to maximize the objective function, denoted by the posterior probability:

$$P(\mathcal{G}|\mathcal{D}) = P(\hat{\mathcal{G}}, \theta|\mathcal{D}) \quad (2)$$

$$= P(\hat{\mathcal{G}}|\mathcal{D})P(\theta|\mathcal{D}, \hat{\mathcal{G}}) \quad (3)$$

The first term models the structure of an And-Or graph \mathcal{G} from a human demonstration \mathcal{D} . To solve the first term, we manually design the structure of the S-AoG, but we learn the T-AoG and C-AoG structure automatically [4] [24] [14].

The second term models the parameters θ in the graph, given the learned knowledge graph structure. It is reformulated as follows:

$$P(\theta|\mathcal{D}, \hat{\mathcal{G}}) \propto \prod_{D_i \in \mathcal{D}} P(D_i|\theta, \hat{\mathcal{G}}) \quad (4)$$

$$\approx \prod_{D_i \in \mathcal{D}} \max_{pg_i} P(D_i|pg_i, \theta, \hat{\mathcal{G}})P(pg_i|\theta, \hat{\mathcal{G}}) \quad (5)$$

where pg_i is the parse graph of D_i . A parse graph is an instance of \mathcal{G} where each Or-node decides one of its children. $P(pg_i|\theta, \hat{\mathcal{G}})$ is the prior probability distribution of parse graph pg_i given \mathcal{G} . To simplify the learning process, we set it as a uniform distribution. Thus,

$$P(\theta|\mathcal{D}, \hat{\mathcal{G}}) \propto \prod_{D_i \in \mathcal{D}} \max_{pg_i} P(D_i|pg_i, \theta, \hat{\mathcal{G}}) \quad (6)$$

And,

$$P(D_i|pg_i, \theta, \hat{\mathcal{G}}) = \prod_{v \in V^{AND}} P(Ch_v|v, \theta_v^{AND}) \quad (7)$$

$$\prod_{v \in V^{OR}} P(Ch_v|v, \theta_v^{OR}) \quad (8)$$

$$\prod_{v \in V^T} P(D_i|v) \quad (9)$$

where Ch_v denotes the child of a non-terminal node $v \in V^{AND} \cup V^{OR}$. The probability derivation represents a generating process from a parent node to its child node, and stops at the terminal nodes to generate the sample D_i . The parameters are learned in an iterative process through a Minimax Entropy algorithm explain in more detail later.

C. S-AoG: Spatial Concepts Model

A powerful way to capture perceptual information is through a visual grammar to produce the most probable interpretations of observed images. Therefore, we represent spatial concepts through a stochastic Spatial And-Or Graph (S-AoG) [5]. Nodes in the S-AoG represent visual information of varying levels of abstraction. The deeper a node lies in the graph, the more concrete of a concept it represents. An And-node signifies physical compositionality (i.e. a wheel is a part of a car) whereas an Or-node describes structural variation (i.e. a car is a type of vehicle).

As demonstrated in Figure 1, the root node of the S-AoG encompasses all possible spatial states a robot may perceive. Here, the ‘‘Indoor scene’’ is decomposed into ‘‘Foreground’’ and ‘‘Background,’’ which are then further decomposed. The nodes deeper in the tree represent finer and finer concepts until they end up the terminal nodes consisting of grounded perception units such as the sleeve of t-shirt.

D. T-AoG: Temporal Concepts Model

The action-space of the world is often an assortment of compositional and variational sub-actions. The hierarchical nature of actions leads us to represent actions by a stochastic

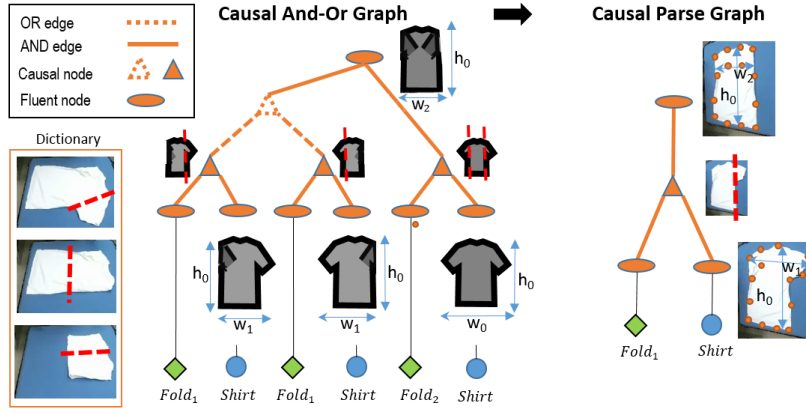


Fig. 3. The Causal And-Or Graph encapsulates the fluent changes per action. The parse graph on the right shows the reasoning system in action.

Temporal And-Or Graph (T-AoG) [4]. And-nodes correspond to a sequence of actions (i.e. close the door, then lock it), whereas Or-nodes correspond to alternate conflicting actions (i.e. close the door, or open the door). The leaf nodes of this graph are atomic action primitives that the robot can immediately perform. Different sequences of atomic actions produce different higher-level actions.

The T-AoG structure is learned automatically using techniques from Si et al. [4] establishing an initial knowledge base of actions. Our T-AoG does not learn new atomic actions, but may learn higher-level actions that are built from these atomic actions. By fixing the set of atomic actions, we ensure the grounding of higher-level actions to alleviate the correspondence problem. Our framework assumes detectors of such atomic action as input.

As shown in Figure 2, the root node of the T-AoG represents all possible actions. As we traverse the tree down, the actions become less and less abstract, until they can no longer be simplified. Therefore, the robot can unambiguously perform the atomic actions represented by the leaf nodes.

The T-AoG provides us a way to define the structure and sequence of actions, but how an action causes a change in state is incorporated in the causality data structure defined next.

E. C-AoG: Causal Concepts Model

Causality is defined as a fluent change due to a relevant action. We can think of fluents as functions on a situation $x_1(s), x_2(s), \dots$, such as the state of a car’s engine (on vs. off) or its current speed (5mph, 10mph, etc.). We use the Causal And-Or Graph (C-AoG) to encapsulate causality learned from human demonstration [14], as shown in Figure 3. Each causal node is a fluent change operator, transforming an input fluent to an output fluent by using an action from the T-AoG. As shown in the diagram, there are various ways to reach the same state. Or-nodes capture the various ways a fluent may change from one state to another.

From the point of view of automated planning, fluents are multi-variate observations of a state. The fluents that change due to a relevant action are vital for predicting future actions. If a fluent does not change from a change-inducing action,

then it is irrelevant with respect to the action. These time-invariant properties as defined as “attributes” of the node (i.e. color, weight). Additionally, fluents that change due to an inertial action (i.e. actions that are irrelevant to a fluent change) are noted inconsistent.

For example, given an cloth s , let fluent $x_1(s)$ represent high-level abstract information such as the shape of a cloth, whereas if the cloth is a shirt, fluent $x_2(s)$ represents specific keypoints for shirts. The C-AoG structure is learned through an information projection pursuit outlined by Fire et al [14]. The STC-AoG uses these relevant fluent changes to plan out tasks.

F. Relational Model between Spatial, Temporal, Causal And-Or Graph

Each of the three And-Or Graphs are unified into a common framework for a complete representation of the world [11]. This explicit knowledge is represented by a hierarchical graphical network specifying a stochastic context sensitive grammar [15], called the the Spatial, Temporal, and Causal And-Or Graph (STC-AoG) [11]. The cloth-folding task in our real-time robot framework is incorporated as described in Figure 4.

Formally, the fluent functions $\forall j x_i(s_j)$ partition the reals \mathbb{R} . Two fluents $x_i(s_a)$ and $x_i(s_b)$ are identical if they belong in the same partition. Each spatial or temporal situation s_i may have multiple fluents (x_1, x_2, \dots) .

$$x(s_i) = \begin{pmatrix} x_1(s_i) \\ x_2(s_i) \\ \dots \end{pmatrix} \quad (10)$$

The fluent change between two states s_j and s_k is formally defined as a binary vector:

$$\Delta x(s_j, s_k) = \begin{pmatrix} \Delta x_1(s_j, s_k) \\ \Delta x_2(s_j, s_k) \\ \dots \end{pmatrix} \quad (11)$$

$$\Delta x_i(s_j, s_k) = \begin{cases} 0 & \text{if } x_i(s_j) = x_i(s_k) \\ 1 & \text{otherwise} \end{cases}$$

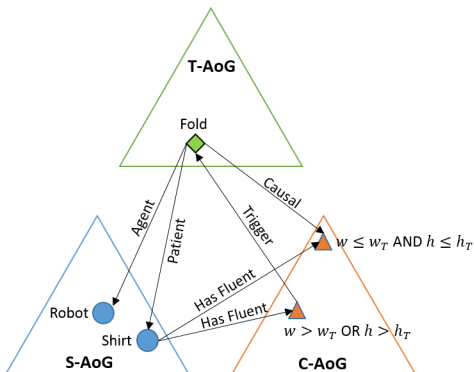


Fig. 4. For illustrative purposes, this diagram shows simple interactions between the spatial, temporal, and causal And-Or graphs. When the width w or height h of the shirt is larger than the target width w_T or height h_T , the C-AoG triggers a fold action in an attempt to reach a smaller folded shirt. The robot then folds the shirt to produce the desired width and height ($w \leq w_t$ AND $h \leq h_t$).

By accumulating human demonstrations of an action, we obtain a set of video clips $Q_a = \{q_1, q_2, \dots\}$ for a specific action a , where q_i is a video clip showing action a . The score $w_j(a)$ of an action to make a fluent change is defined as:

$$\forall j \ w_j(a) = P(\Delta x_j = 1 \mid Q) = \frac{\sum_i 1_{\Delta x_j=1|q_i}}{\|Q\|} \quad (12)$$

with the scores normalized by $\sqrt{\sum_j w_j(a)^2}$.

Fluents that represent specific properties, such as keypoints, tend to be heavier weighted than those that are broad high-level concepts, such as shape [17]. The fluents are typically hand-chosen, but we suggest automatically generating various abstractions of fluents by varying the dimensionality of autoencoders. Recent work on spatial semantics [26] can also initialize nodes with a set of useful fluents.

The STC-AoG is not just a knowledge representation system, but also a hierarchical planning graph. Folding a shirt using shirt fluents $x_1(s)$ and $x_2(s)$ has greater affordance than that from using just abstract shape information $x_1(s)$. That way, causal reasoning remains specific to the object, guaranteeing that when folding a shirt, there is less preference to use knowledge about how to fold pants if knowledge about how to fold shirts already exists. We define the affordance of transferring from state s_i to s_j using action a by $\mathbf{aff}(a, s_i, s_j) = w(a)^T \Delta x(s_i, s_j)$, suggesting that the automated planning and reasoning should only be based on the relevant features.

Unifying the three sub-graphs produces a closed-loop framework for robots learning from demonstrations. Moreover, graphs can store relationships in an intuitive and highly regular structure, allowing for algorithms that rely on simple graph manipulations. The real world is encoded through perception into the S-AoG to form a physical belief state of the world. The learning algorithm constructs a C-AoG to understand actions from human demonstrations. And lastly, inference combines the reasoning from the C-AoG and the actuators from the T-AoG to physically perform the task. The

energy of the joint parse graph [11] combines the energy terms of each:

$$E_{STC}(pg) = E_S(pg) + E_T(pg) + E_C(pg) + \sum_{r \in R_{pg}^*} E_R(r) \quad (13)$$

We use generative learning by the Minimax Entropy Principle [19] to learn the probability distribution of STC parse graphs $P(pg)$. Doing so assumes that the sample mean of statistics $\phi_j(pg)$ should approach the true expectation s_j from observations. The parameters are solved by minimizing the Kullback-Leibler divergence between the observed distribution and the candidate $KL(f||p) = E_f[\log f(pg)] - E_f[\log p(pg)]$. This simplifies to a maximum likelihood estimate, formulated by

$$p^* = \operatorname{argmax}_{p \in \Omega} E_f[\log p(pg)] = \operatorname{argmax}_{p \in \Omega} \sum_{i=1}^n \log p(pg_i) + \epsilon \quad (14)$$

Iteratively, we choose the statistics $F = \{\phi_1, \phi_2, \dots\}$ that minimize the entropy of the model, and the parameters β that yield maximum entropy.

$$p^* = \operatorname{argmin}_F \{ \max_{\beta} \text{entropy}(p(pg; \theta)) \} \quad (15)$$

Effectively, the robot “daydreams” possible probability distributions of parse graphs to converge with observations. During inference, it samples a parse graph to perform the action.

G. Learning Motor Control

The STC-AoG expresses explicit knowledge in a graphical structure easily understandable by humans, acting as a gateway for communication. However, the STC-AoG only defines discrete salient spatial, temporal, and causal concepts. The interpolation of how an individual action is performed requires a specification of the fine motor skills involved as well as an assignment of probability distribution parameters.

The explicit knowledge captured by a causal node represents a conformant plan learned by human demonstrations. The information stored in the STC-AoG only provides results from discrete time-steps, $t \in \mathbb{N}$. Its state-action table represents fluent changes by $x^{t+1}(s) = f(x^t(s), x^t(a))$. To shift paradigms from explicit to implicit knowledge, we relax the assumption of null run-time observability, and use a finer distinction in time, $x^{t+\delta t}(s) = f(x^t(s), x^t(a))$. By learning this continuous function f , the robot system is capable of verifying, correcting, and inferring causal relations to adapt to dynamic environments.

We make two assumptions to simplify the learning of f . First, we restrict the range of spatial and temporal changes to adhere to spatiotemporal continuity, rendering sudden changes impossible. Second, we use a physical simulator based on perception encoded by the STC parse graph (STC-pg) to compare with reality at practically infinitesimal time intervals. When a discrepancy is detected, we point fault at the robot’s actions. The feedback learning system uses a

simplified optimization process inspired by Atkeson et al [21] to update the control mechanics. Adjusting the parameters of the simulator to adhere to reality also reveals useful knowledge, but it is out of scope for this study.

H. Inference

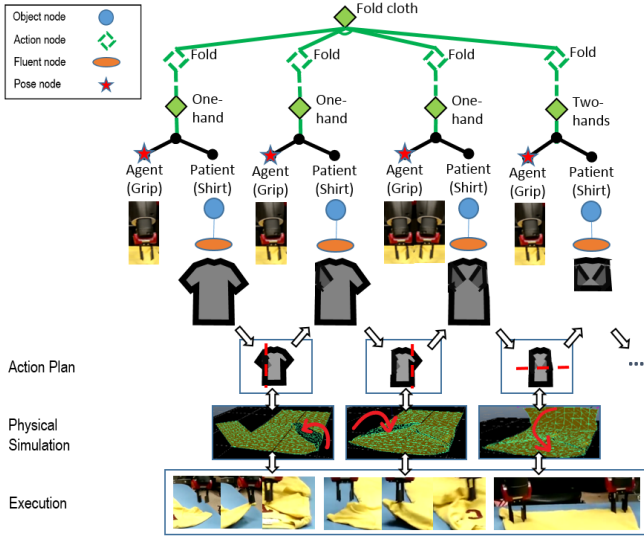


Fig. 5. The inference engine samples a parse graph to create a conformant action plan. There is feedback between the plan, its simulation, and the corresponding perceived execution.

Since the STC-AoG model is generatively learned, we infer a parse graph through a simple sampling process. As seen in Figure 5, the procedurally generated parse graph lays out a conformant action plan for the robot. It then creates a simulation of the action by converting the STC-pg into a motion plan and spatial objects into 3D meshes from point cloud.

The simulation plan is matched with reality at small interval steps to verify that the robot is at its corresponding simulated state. In case of substantial mismatch between expected and actual states, the robot understands the action did not complete, and that a new action plan must be generated based on the latest perception input. Concretely, the sampling procedure is encapsulated by the algorithm in Figure 6.

IV. EXPERIMENTS

We conduct our experiments on a cloth-folding task. The S-AoG models the physical status of the cloth, table, robot, human, and various decompositions of each. The T-AoG consists of three atomic actions to span the action-space for this simple task: *MoveArm(a)*, *Grab*, and *Release*. A *Fold* action in the T-AoG is a higher-level And-node consisting of four children: *MoveArm(a)*, *Grab*, *MoveArm(b)*, and *Release*, with the corresponding textual representation: $Fold(a, b) = MoveArm(a); Grab; MoveArm(b); Release$. And consequently, a specific instance of folding is a series of *Fold* actions: $FoldStyle1 = Fold(a, b); Fold(c, d); \dots; Fold(y, z)$.

- 1: **while** camera is producing image I **do**
- 2: $pg_S^t \leftarrow Interpret(G_S, I_t)$
- 3: $pg_T^t \leftarrow Sample(G_{STC}, pg_S^t)$
- 4: $pg_C^t \leftarrow Sample(G_{STC}, pg_S^t, pg_T^t)$
- 5: $pg_{STC} \leftarrow Merge(pg_S^t, pg_T^t, pg_C^t)$
- 6: PerformWithFeedback(pg_{STC})
- 7: **end while**

Fig. 6. The robot inference algorithm performs tasks on a learned STC-AoG. It interprets the sensory input as spatial, temporal, and causal parse graphs, which are merged to formed a joint representation that is sampled and acted on.

Lastly, the C-AoG nodes describe how to fold a shirt from one state to another, learned through human demonstrations.

We use Baxter, a two-armed industrial robot to perform our cloth-folding task. Each arm consists of 7 degrees of freedom that are adjusted through inverse kinematics relative to the robot’s frame of reference. The robot’s primary perception sensor is an Asus PrimeSense camera that provides an aligned RGB-D (Red, Green, Blue, and Depth) point cloud in real-time. In order to use localization results from perception, we compute the affine transformation matrix from the camera coordinate system to that of the robot. All components interact together through the Robot Operating System (ROS).

The STC-AoG is stored in the platform-independent Graphviz DOT language, and used by our platform written in C++. The hand-designed perception logic combines off-the-shelf graph-based [23] and foreground/background [22] segmentation to localize a cloth per frame. On top of that, we train a shirt detector model using a Support Vector Machine to facilitate narrowing down the search for an optimal S-AoG parse graph. Each cloth node has a fluent x_1 describing the low-level shape. If a cloth is a shirt, we represent the structure of its keypoints as another fluent x_2 . We simplify learning the probability distribution of parse graphs by limiting the number of statistics to $F = \{\phi_1\}$, where ϕ_1 is the affordance cost of the action sequence in a STC-pg.

Performance on a task is measured by the percent of successful actions throughout the task. The overall performance is the average of all task performances over multiple trials. An action is successful if performing the action satisfies the pre- and post-conditions of the causal relationship used.

A. Experiment Settings

In the first set of experiments, we measure the performance of representing learned knowledge from human demonstrations. After watching human demonstrations, the robot generates an action plan step by step. The human performs the action suggested by the robot, and at each step, the human qualitatively verifies whether the robot’s action was indeed the intended action as per the demonstration. If verification fails in either case, then the action is marked unsuccessful, and otherwise it is marked successful. This performance score on learning will set the baseline for the next set of experiments.

In the second series of experiments, we measure the quality of grounding the learned knowledge to the robot’s actions. This time we let the robot, instead of the human, perform the actions. We compare the performance of the robot folding clothes with the results from the first set of experiments to evaluate the success of grounding physical actions to see how well they match that of a human. The expected performance should be less than the ground truth established from the previous experiment.

In the third series of experiments, we measure the improvements from a feedback system compared to no feedback. We expect that the performance score calculated through this step should be higher than that from the previous experiment, but lower than the ground truth.

Finally, we are also curious how much we can stretch the generalizability of a learned task. After demonstrating how to fold a t-shirt, we ask the robot to infer how to fold different articles of clothing, such as full-sleeve shirts, towels, and pants. The criteria for generalizability of knowledge will follow the similar performance procedure as in the previous experiments.

B. RESULTS

On 10 trials per four sets of different t-shirt folding demonstrations D_1, D_2, D_3, D_4 , we measure the average performance of using our system to learn knowledge, ground robot actions, and control feedback.

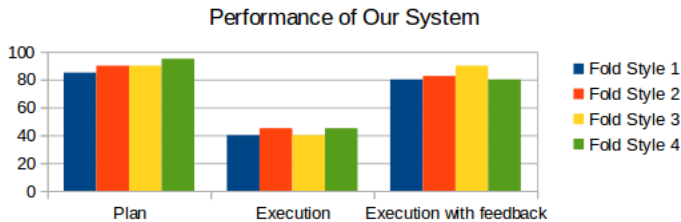


Fig. 7. Our learning system successfully understood the various folding techniques. It had some difficulty executing the task using simply a conformant plan, but with added feedback the execution was highly successful.

As seen in Figure 7, our knowledge representation system was able to characterize the cloth-folding task enough to faithfully communicate with a human, producing a learned representation with an average performance of 90%. This sets the upper bound for the next two inference experiments. As anticipated, our framework was able to ground the actions with a performance of 42.5%. The low score indicates that although the robot knows what to do, there is still a discrepancy between the human’s action and that generated by the STC-AoG. By adding feedback correction, the performance leaped to 83.125%, also matching our expectation.

The performance of generalizability was measured after training the robot on only t-shirt folding videos. The results are visualized in Figure 8. For example, since a full-sleeve shirt may have the same width and height fluents as that of a t-shirt, the inference plan for folding a full-sleeve shirt performed very well. Moreover, the robot was able to

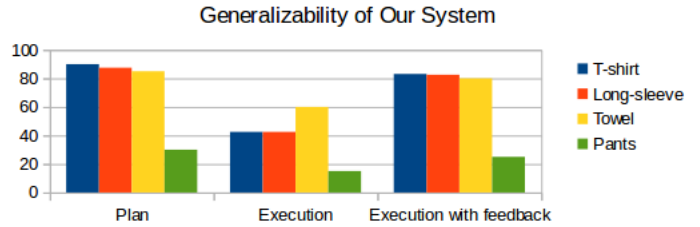


Fig. 8. Our knowledge framework correctly understood how to generalize a t-shirt folding instruction to long-sleeve shirts and towels; however, it expectedly had difficulty extrapolating its knowledge to fold pants.

generate reasonable action plans to fold a towel it has never seen, since a t-shirt with both its sleeves folded resembles the same rectangular shape of a towel. However, generating a reasonable inference result for folding pants was less successful due to the natural lack of knowledge transferred between a shirt folding and pant folding task. Figure 9 shows a few qualitative results of successful folding plans and executions.



Fig. 9. Some qualitative results on the robot execution after learning from human demonstrations.

V. DISCUSSION AND FUTURE WORK

The experiments show preliminary support for the expressive power of the robot learning and execution framework laid out in this paper. While we focus heavily in the cloth-folding domain, the framework may be used for training any goal-oriented task. In future work, we wish to continue improving the robustness of each spatial, temporal, and causal And-Or graph to optimize for speed and accuracy.

The STC-AoG acts as a language to ground knowledge and reasoning into robot actions. Since the knowledge representation and robot action planning systems share the same And-Or graph data structure, the graph acts as a programming language for the robot, and self-updating the graph is an act of metaprogramming.

Due to the hierarchical nature of the STC-AoG, the higher level nodes are readily articulated and understandable by humans. We are currently working on incorporating natural language statements, commands, and questions to more easily allow humans to manipulate the graph. To scale up the graph for life-long learning, we are investigating other practical storage solutions, including graph-based databases such as Neo4j. Since the graph is sufficient to transfer knowledge, we can upload different skills to a cloud platform and share knowledge between different robots.

Limits in physical reachability and dexterity of the robot arms played a crucial difficulty in mapping action plans to

motor control execution. If a grip location was unreachable, the conformant plan would fail to execute the action at all. Fortunately, by introducing the feedback control system, we were able to at least extend the reach as far as possible to grip a reasonable point.

Lastly, the performance of the causal learning system relies on successfully detecting fluent changes. This requires adjusting thresholds for fluent-change detectors until the results seem just right. We solved this problem by offline supervised learning for our chosen fluents, but we set aside the problem of learning these threshold parameters online to future work.

VI. CONCLUSIONS

The stochastic graph-based framework is capable of representing task-oriented knowledge for tractable inference and generalizability. It successfully unified theoretical foundations of And-Or perception grammars to a practical robotics platform. The experimental results support our claims for grounding learned knowledge to execute tasks accurately. We also express the generalizability of our framework by extrapolating from human demonstrations of folding a t-shirt to other articles of clothing. And lastly, our novel framework can make use of perceived discrepancies between high-level action plans and low-level motor control to verify and correct actions.

ACKNOWLEDGMENT

The authors would like to thank the support of DARPA SIMPLEX project N66001-15-C-4035 and DARPA MSEE project FA 8650-11-1-7149. In addition, we would like to thank SRI International and OSRF for their support.

REFERENCES

- [1] E. Theodorou, J. Buchli, and S. Schaal, "Reinforcement learning of motor skills in high dimensions: A path integral approach," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2397–2403.
- [2] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, p. 0278364911426178, 2011.
- [3] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, 2007.
- [4] Z. Si, M. Pei, B. Yao, and S.-C. Zhu, "Unsupervised learning of event and-or grammar and semantics from video," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 41–48.
- [5] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259–362, 2006.
- [6] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos, "Robot learning manipulation action plans by unconstrained videos from the world wide web," in *The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.
- [7] N. Shukla, C. Xiong, and S.-C. Zhu, "A unified framework for human-robot knowledge transfer," in *2015 AAAI Fall Symposium on AI for Human-Robot Interaction (AI-HRI 2015)*, 2015.
- [8] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang, "Error-driven incremental learning in deep convolutional neural network for large-scale image classification," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 177–186.
- [9] A. L. Blum and M. L. Furst, "Fast planning through planning graph analysis," *Artificial Intelligence*, vol. 90, no. 1, pp. 281–300, 1997.
- [10] S. Miller, J. van den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 2, pp. 249–267, 2012.
- [11] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S.-C. Zhu, "Joint video and text parsing for understanding events and answering queries," *Multimedia, IEEE*, vol. 21, no. 2, pp. 42–70, 2014.
- [12] J.-W. Ha, K.-M. Kim, and B.-T. Zhang, "Automated construction of visual-linguistic knowledge via concept learning from cartoon videos," *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, Austin, 2015.
- [13] G. W. Strong, "Z. w. pylyshyn: Computation and cognition: Toward a foundation for cognitive science. cambridge, massachusetts: The mit press, 1984, 320 pp." *Behavioral Science*, vol. 31, no. 4, pp. 286–289, 1986.
- [14] A. S. Fire and S. Zhu, "Learning perceptual causality from video," in *Learning Rich Representations from Low-Level Sensors, Papers from the 2013 AAAI Workshop, Bellevue, Washington, USA, July 15, 2013*, 2013.
- [15] J. Rekers and A. Schrr, "A parsing algorithm for context-sensitive graph grammars," Tech. Rep., 1995.
- [16] H. Chen, Z. J. Xu, Z. Q. Liu, and S. C. Zhu, "Composite templates for cloth modeling and sketching," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, June 2006, pp. 943–950.
- [17] C. Chao, M. Cakmak, and A. Thomaz, "Towards grounding concepts for cloth modeling from demonstration," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2, Aug 2011, pp. 1–6.
- [18] E. T. Mueller, in *Commonsense Reasoning*, E. T. Mueller, Ed. Morgan Kaufmann, 2006.
- [19] S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Computation*, vol. 9, no. 8, pp. 1627–1660, 1997.
- [20] R. P. N. Rao, A. P. Shon, and A. N. Meltzoff, "A bayesian model of imitation in infants and robots," in *In Imitation and Social Learning in Robots, Humans, and Animals*. Cambridge University Press, 2004, pp. 217–247.
- [21] C. Atkeson and S. Schaal, "Learning tasks from a single demonstration," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2, Apr 1997, pp. 1706–1712 vol.2.
- [22] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut -interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.
- [23] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, Sept. 2004.
- [24] K. Tu, M. Pavlovskaja, and S.-C. Zhu, "Unsupervised structure learning of stochastic and-or grammars," in *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 1322–1330.
- [25] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Motion planning for dynamic folding of a cloth with two high-speed robot hands and two high-speed sliders," in *ICRA. IEEE*, 2011, pp. 5486–5491.
- [26] K. Zampogiannis, Y. Yang, C. Fermuller, and Y. Aloimonos, "Learning the spatial semantics of manipulation actions through preposition grounding," in *ICRA. IEEE*, 2015, pp. 1389–1396.
- [27] A. Doumanoglou, A. Kargakos, T. Kim, and S. 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 2014, Hong Kong, China, May 31 - June 7, 2014*, 2014, pp. 987–993.
- [28] P. C. Wang, S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Perception for the manipulation of socks," in *IROS*, 2011, pp. 4877–4884.