TANGO: Commonsense Generalization in Predicting Tool Interactions for Mobile Manipulators

Shreshth Tuli[*1,2] , **Rajas Bansal**[*1] , **Rohan Paul**[1] and **Mausam**[1]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India
[2]Department of Computing, Imperial College London, UK

Abstract

Robots assisting us in factories or homes must learn to make use of objects as tools to perform tasks, e.g., a tray for carrying objects. We consider the problem of learning commonsense knowledge of when a tool may be useful and how its use may be composed with other tools to accomplish a high-level task instructed by a human. We introduce a novel neural model, termed TANGO, for predicting task-specific tool interactions, trained using demonstrations from human teachers instructing a virtual robot in a physics simulator. TANGO encodes the world state, comprising objects and symbolic relationships between them, using a graph neural network. The model learns to attend over the scene using knowledge of the goal and the action history, finally decoding the symbolic action to execute. Crucially, we address generalization to unseen environments where some known tools are missing, but alternative unseen tools are present. We show that by augmenting the representation of the environment with pre-trained embeddings derived from a knowledge-base, the model can generalize effectively to novel environments. Experimental results show a 60.5-78.9% absolute improvement over the baseline in predicting successful symbolic plans in unseen settings for a simulated mobile manipulator.

1 Introduction

Advances in autonomy have enabled robots to enter human-centric domains such as homes and factories where we envision them performing general purpose tasks such as transport, assembly, and clearing. Such tasks require a robot to interact with objects, often using them as *tools*. For example, a robot asked to "take fruits to the kitchen", can use a *tray* for carrying items, a *stick* to fetch objects beyond physical reach and may use a *ramp* to reach elevated platforms. We consider the problem of predicting *which* objects could be used as tools and *how* their use can be composed for a task.

Learning to predict task-directed tool interactions poses several challenges. First, real environments (a household or factory-like domain) are typically large where an expansive number of tool interactions may be possible (e.g., objects supporting others while transporting). Acquiring data for all feasible tool objects or exploring the space of tool interactions is challenging for any learner. Second, the robot may encounter new environments populated with novel objects not encountered during training. Hence, the agent's model must be able to *generalize* by reasoning about interactions with novel objects unseen during training. Humans possess innate commonsense knowledge about contextual use of tools for an intended goal [Allen *et al.*, 2019]. For example, a human actor when asked to move objects is likely to use trays, boxes, or even improvise with a new object with a flat surface. We aim at providing this commonsense to a robotic agent, so that it can generalize its knowledge to unseen tools, based on shared context and attributes of seen tools (see Figure 1).

We acquire a data set of robot plans, where a human teacher guides a simulated mobile manipulator to perform tasks involving multi-step use of objects as tools. The model encodes the world state using a graph neural network and learns to attend over the scene using knowledge of the goal and the action history, finally decoding the symbolic action to execute. The model learns a dense representation of the object-centric graph of the environment which is augmented with word embeddings from a knowledge base, facilitating generalization to novel environments. The action predictions are interleaved with physics simulation (or execution) steps which ameliorates the need for modeling the complex effects of actions inherent in tool interactions. We term the model, Tool Interaction Prediction Network for Generalized Object environments (TANGO). Experimental evaluation with a simulated mobile manipulator demonstrate (a) accurate prediction of a tool interaction sequences with high executability/goal-attainment likelihood, (b) common sense generalization to novel scenes with unseen object instances, and (c) robustness to unexpected errors during execution.

2 Related Work

Learning control policies for manipulating tools has received recent attention in robotics. [Finn *et al.*, 2017] learn tool manipulation policies from human demonstrations. [Holladay *et al.*, 2019] learn physics-based models and *effects* enabling

[*]Indicates equal contribution.
Project Page: https://github.com/reail-iitd/tango.

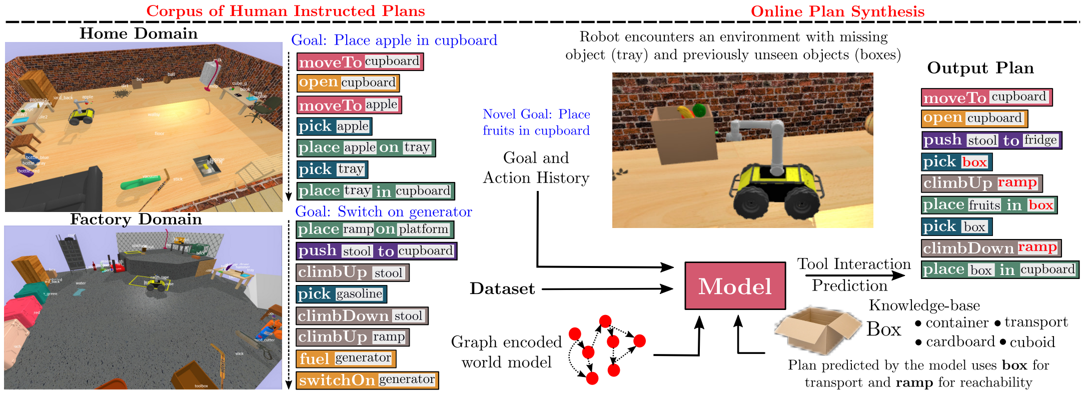


Figure 1: TANGO acquires commonsense knowledge from human demonstrations leveraging graph-structured world representation, knowledge-base corpora and goal-conditioned attention to perform semantic tasks. Our aim is to acquire commonsense knowledge to develop a generalized goal-conditioned policy for a robot.

compositional tool use. [Toussaint *et al.*, 2018] present a planner to compose physics tool interactions using a logic-based symbolic planner. The aforementioned works focus on learning *how* to manipulate a tool. In contrast, we extend our prior work on predicting *which* objects may serve as tools for a given task [Bansal *et al.*, 2020] to generate plans.

Others address the problem of acquiring knowledge for completing high-level task specifications. [Puig *et al.*, 2018, Liao *et al.*, 2019] create a knowledge base of task decompositions as *action sketches* and learn to translate sketches to executable plans. These efforts rely on the causal knowledge of sequences on sub-steps required to achieve an activity which are then contextually grounded. Instead, this work learns compositional tool use required to achieve the task without any causal sequence as input. [Huang *et al.*, 2019] learn task decompositions from human demonstration videos. However, the work does not explicitly model the physical constraints of the robot and does not generalize to new environments. [Boteanu *et al.*, 2015] present a symbolic system where a robot imitates a demonstrations from a single teacher. In new environments, it adapts the plan by performing object replacements using ConceptNet relation edges. In contrast, this paper proposes a neural model trained using a corpus of multiple and varied demonstrations provided by several teachers. Our model uses a dense embedding of semantic concepts, enabling generalization beyond relationships explicitly stored in ConceptNet.

In the context of robot instruction following, [Nyga *et al.*, 2018] and [Kho *et al.*, 2014] use curated knowledge bases to infer missing portions in instructions. Other such as [Jain *et al.*, 2015] learn motion preferences implicit in commands. [Bisk *et al.*, 2020] learn *physical* common sense knowledge for NLP tasks such as QA, analogy reasoning etc. The aforementioned approaches predict latent constraints or affordances for a specified task. This work, additionally predicts the *sequence* of tool interactions implicitly learning the causal relationships between tools use and effects. [Misra *et al.*, 2016] ground instructions for recipe preparation tasks. Their model can generalize to new recipes, but only in environments with previously *seen* objects. In contrast, our model generalizes to worlds with previously *unseen* tools.

3 Problem Formulation

Robot and Environment Model. We consider a mobile manipulator operating in a known environment populated with objects. An object is associated with a pose, a geometric model and symbolic states such as $\mathrm{Open}/\mathrm{Closed}$, On/Off etc. We consider object interactions such as (i) *support* e.g., a block supported on a tray, (ii) *containment*: items placed inside a box/carton and (iii) *attachment*: a nail attached to a wall, and (iv) *contact*: a robot grasping an object. Let s denote the world state that maintains (i) metric information: object poses, and (ii) symbolic information: object states, class type and object relations as OnTop, Near, Inside and $\mathrm{ConnectedTo}$. Let s_0 denote the initial world state and $\mathcal{O}(\cdot)$ denote a map from world state s to the set of object instances $O = \mathcal{O}(s)$ populating state s.

Let A denote the robot's symbolic action space. An action $a \in A$ is abstracted as $I(o^1, o^2)$, with an action type predicate $I \in \mathcal{I}$ that affects the states of objects $o^1 \in O$ and $o^2 \in O$, for instance, $\mathrm{Move}(\mathrm{fruit}_0, \mathrm{tray}_0)$. We shall also use the notion of a timestamp as a subscript to indicate prediction for each state in the execution sequence. The space of robot interactions include grasping, releasing, pushing, moving an object to another location or inducing discrete state changes (e.g.. opening/closing an object, operating a switch or using a mop). We assume the presence of an underlying low-level metric planner, encapsulated as a robot *skill*, which realizes each symbolic action or returns if the action is infeasible. Robot actions are stochastic, modeling execution errors (unexpected collisions) and unanticipated outcomes (objects falling, changing the symbolic state). Let $\mathcal{T}(\cdot)$ denote the transition function. The successor state s_{t+1} upon taking the action a_t in state s_t is sampled from a physics simulator. Let $\eta_t = \{a_0, a_1, \ldots, a_{t-1}\}$ denote the *action history* till time t.

The robot is instructed by providing a *declarative* goal g expressing the symbolic constraint between world objects. For example, the *declarative* goal, "place milk in fridge" can be expressed as a constraint $\mathrm{Inside}(\mathrm{milk}_0, \mathrm{fridge}_0)$ between specific object instances. Finally, the robot must synthesize a plan to satisfy the goal constraints. Goal-reaching plans may require using some objects as tools, for instance, using a container for moving items, or a ramp negotiate an elevation.

Predicting Tool Interactions. Our goal is to aim at learning common sense knowledge about *when* an object can be used as a tool and *how* their use can be sequenced for goal-reaching plans. We aim at learning a policy π that estimates the next action a_t conditioned on the the goal g and the initial state s (including the action history η_t, such that the robot's *goal-reaching* likelihood is maximized. We adopt the MAXPROB-MDP [Mausam and Kolobov, 2012] formalism and estimate a policy that maximizes the goal-reaching likelihood from the given state [1]. Formally, let $P^{\pi}(s,g)$ denote the *goal-probability* function that represents the likelihood of reaching the goal g from a state s on following π. Let $S_t^{\pi_s}$ be a random variable denoting the state resulting from executing the policy π from state s for t time steps. Let $\mathcal{G}(s,g)$ denote the Boolean *goal check* function that determines if the intended goal g is entailed by a world state s as $\mathcal{G}(s,g) \in \{\mathrm{True(T)}, \mathrm{False(F)}\}$. The policy learning objective is formulated as maximizing the likelihood of reaching a goal-satisfying state g from an initial state s_0, denoted as $\max_{\pi} P^{\pi}(s_0,g) =$

$$\max_{\pi} \sum_{t=1}^{\infty} P\bigg(\mathcal{G}(S_t^{\pi_{s_0}}, g) = \mathrm{T} : \mathcal{G}(S_{t'}^{\pi_{s_0}}, g) = \mathrm{F},\ \forall t' \in [1,t)\bigg).$$

The policy is modeled as a function $f_\theta(.)$ parameterized by parameters θ that determines the next action for a given world state, the robot's action history and the goal as $a_t = f_\theta(s_t, g, \eta_t)$. We adopt imitation learning approach and learn the function $f_\theta(.)$ from demonstrations by human teachers. Let $\mathcal{D}_{\mathrm{Train}}$ denote the corpus of N goal-reaching plans,

$$\mathcal{D}_{\mathrm{Train}} = \{(s_0^i, g^i, \{s_j^i, a_j^i\}) \mid i \in \{1,N\}, j \in \{0, t_i - 1\}\},$$

where the i^{th} datum consists of the initial state s_0^i, the goal g^i and a state-action sequence $\{(s_0^i, a_0^i), \ldots, (s_{t-1}^i, a_{t-1}^i)\}$ of length t_i. The set of human demonstrations elucidate common sense knowledge about *when* and *how* tools can be used for attaining provided goals. The data set $\mathcal{D}_{\mathrm{Train}}$ supervises an imitation loss between the human demonstrations and the model predictions, resulting in learned parameters θ^*. Online, the robot uses the learned model to sequentially predict actions and execute in the simulation environment till the goal state is attained. We also consider the *open-world* case where the robot may encounter instances of *novel* object categories *unseen* in training, necessitating a *zero-shot* generalization.

4 Technical Approach

TANGO learns to predict the next robot action a_t, given the world state s_t, the goal g and the action history η_t. TANGO is realized as a neural network model f_θ as follows:

$$a_t = f_\theta(s_t, g, \eta_t) = f_\theta^{act}\left(f_\theta^{goal}\left(f_\theta^{state}(s_t), g, f_\theta^{hist}(\eta_t)\right)\right)$$

It adopts an object-centric graph representation, learning a state encoding that fuses metric-semantic information about objects in the environment via function $f_\theta^{state}(\cdot)$. The function $f_\theta^{hist}(\cdot)$ encodes the action history. The model learns to attend over the world state conditioned on the declarative goal and the history of past actions through $f_\theta^{goal}(\cdot)$. Finally, the learned encodings are decoded as the next action for the robot to execute via $f_\theta^{act}(\cdot)$. Crucially, the predicted action is grounded over an *a-priori* unknown state and type of objects in the environment. The predicted action is executed in the environment and the updated state action history is used for estimation at the next time step. The constituent model components are detailed next.

Graph Structured World Representation. TANGO encodes a robot's current world state s_t as an object-centric graph $G_t = (O, R)$. Each node in the graph represents an object instance $o \in O = \mathcal{O}(s_t)$. The edge set consists of binary relationships OnTop, $\mathrm{ConnectedTo}$, Near and Inside between objects $R \subseteq O \times O$. Let $l_o \in \{0,1\}^p$ represents discrete object states for the object o (e.g. $\mathrm{Open/Closed}$, $\mathrm{On/Off}$). Next, it incorporates a pre-trained function $\mathcal{C}(\cdot)$ that embeds a word (like token of an object class or a relation) to a dense distributed representation, such that semantically close tokens appear close in the learned space [Mikolov *et al.*, 2018]. The use of such embeddings enables generalization, which we discuss subsequently.

Let $e_o = \mathcal{C}(o) \in \mathcal{R}^q$ denote the q-dimensional embedding for an object instance o. The embeddings l_o and e_o model object attributes that initialize the state of each object node in the graph. The local context for each o is incorporated via a Gated Graph Convolution Network (GGCN) [Liao *et al.*, 2019], which performs message passing between 1-hop vertices on the graph. Following [Puig *et al.*, 2018], the gating stage is realized as a Gated Recurrent Unit (GRU) resulting in *graph-to-graph* updates as:

$$\begin{aligned} r_o^0 &= \tanh\left(W_r\left[l_o; e_o\right] + b^r\right), \\ x_o^k &= \sum_{j \in R} \sum_{o' \in N^j(o)} W_j^k r_{o'}^{k-1}, \\ r_o^k &= \mathrm{GRU}\left(r_o^{k-1}, x_o^k\right). \end{aligned}$$

Here, the messages for object o are aggregated over neighbors $N^j(o)$ connected by relation j ($\forall j \in R$) during n convolutions, resulting in an embedding r_o^n for each object instance in the environment.

Fusing Metric Information. Next, TANGO incorporates the metric information associated with objects. Let $pose_o$ and $size_o$ represent the pose and size/extent (along xyz axes) for each object instance. The properties are encoded using a d-layer Fully Connected Network (FCN) with a Parameterized ReLU (PReLU) [He *et al.*, 2015] activation as:

$$\begin{aligned} m_o^0 &= \mathrm{PReLU}\left(W_{mtr}^0[pose_o; size_o] + b_{mtr}^0\right) \\ m_o^k &= \mathrm{PReLU}\left(W_{mtr}^k m_o^{k-1} + b_{mtr}^k\right), \end{aligned}$$

resulting in the metric encoding m_o^d for each object in the scene. A world state encoding (for s_t) is obtained by fusing the semantic and metric embeddings as $f_\theta^{state}(s_t) = \{\tilde{s}_t^o = [r_o^n; m_o^d] | \ \forall o \in \mathcal{O}(s_t)\}$. *Late fusion* of the two encodings allows downstream predictors to exploit them independently.

Encoding the Action History. The task of deciding the next action is informed by the agent's action history in two

[1] MAXPROB-MDP can be equivalently viewed as an infinite horizon, un-discounted MDP with a zero reward for non-goal states and a positive reward for goal states [Kolobov *et al.*, 2011].

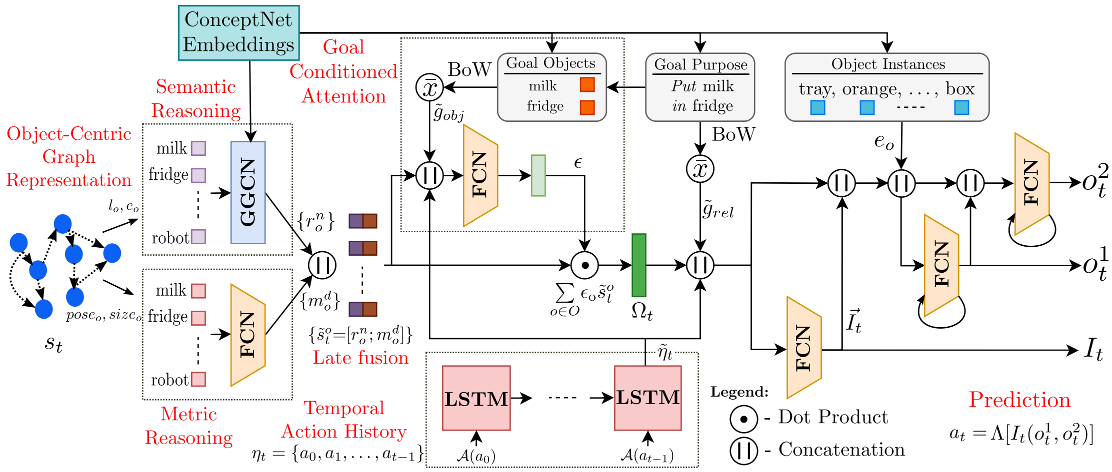


Figure 2: TANGO neural model encodes the metric-semantic world state using graph convolution (GGCN) and fully connected (FCN) layers. The model uses goal information and the robot's action history to attend over a task-specific context, finally decoding the next symbolic action for the robot to execute. A graph structured representation and inclusion of pre-trained word embeddings (from a knowledge base) facilitate generalization in predicting interactions in novel contexts with new objects unseen in training.

ways. First, sequential actions are often temporally correlated. E.g., a placing task often involves moving close to the box, opening it and then placing an item inside it. Hence, maintaining the action history can help in prediction of the next action. Second, the set of actions the robot executed in the past provides a local context indicating the objects the robot may affect in future. Formally, TANGO encodes the temporal action history η_t using an LSTM resulting in embedding vector $f_\theta^{hist}(\eta_t) = \tilde{\eta}_t$. We define action encoding $\mathcal{A}(a_{t-1})$ of $a_{t-1} = I_t(o_{t-1}^1, o_{t-1}^2)$ independent of the object set, as $\mathcal{A}(a_{t-1}) = [\vec{I}_{t-1}; \mathcal{C}(o_{t-1}^1); \mathcal{C}(o_{t-1}^2)]$, where $\vec{I}_{t-1}$ is a one-hot vector over possible interaction types $\mathcal{I}$, and $\mathcal{C}(o_{t-1}^1)$ and $\mathcal{C}(o_{t-1}^2)$ represent the word embeddings of the object instances o_{t-1}^1 and o_{t-1}^2. At each time step t, the LSTM encoder takes in the encoding of previous action, $\mathcal{A}(a_{t-1})$ and outputs the updated encoding $\tilde{\eta}_t$, given as $\tilde{\eta}_t = \mathrm{LSTM}(\mathcal{A}(a_{t-1}), \tilde{\eta}_{t-1})$.

Goal-conditioned Attention. The goal g consists of symbolic relations (e.g. Inside, OnTop etc.) between object instances (e.g., carton and cupboard) that must be true at the end of the robot's plan execution. The declarative goal input to the model is partitioned as relations g_{rel} and the object instances specified in the goal g_{obj}. The resulting encondings are denoted as $\tilde{g}_{rel}$ and $\tilde{g}_{obj}$:

$$\tilde{g}_{rel} = \frac{1}{|g_{rel}|} \sum_{j \in g_{rel}} \mathcal{C}(j) \ \text{ and } \ \tilde{g}_{obj} = \frac{1}{|g_{obj}|} \sum_{o \in g_{obj}} \mathcal{C}(o).$$

Next, the goal encoding and the action history encoding $\tilde{\eta}_t$ is used to learn attention weights over objects in the environment $\epsilon_o = f_\theta^{goal}(\tilde{s}_t^o, \tilde{g}_{obj}, \tilde{\eta}_t)$ [Bahdanau *et al.*, 2014]. This results in the attended scene encoding Ω_t as:

$$\Omega_t = \sum_{o \in O} \epsilon_o \tilde{s}_t^o \text{ where, } \epsilon_o = \mathrm{softmax}\left(W_g[\tilde{s}_t^o; \tilde{g}_{obj}; \tilde{\eta}_t] + b_g\right).$$

The attention mechanism aligns the goal information with the scene learning a task-relevant context, relieving the model from reasoning about objects in the environment unrelated to the task, which may be numerous in realistic environments.

Robot Action Prediction. TANGO takes the encoded information about the world state, goal and action history to decode the next symbolic action $a_t = I_t(o_t^1, o_t^2)$. The three components I_t, o_t^1 and o_t^2 are predicted auto-regressively, where the prediction of the interaction, I_t is used for the prediction of the first object, o_t^1 and both their predictions are used for the second object prediction, o_t^2. The prediction is made using the encoding of the state, i.e the attended scene embedding Ω_t, the relational description of the goal $\tilde{g}_{rel}$ and the action history encoding $\tilde{\eta}_t$. For the object predictors o_t^1 and o_t^2, instead of predicting over a predefined set of objects, TANGO predicts a likelihood score of each object $o \in O$ based on its object embedding $\tilde{s}_t^o$, and selects the object with highest likelihood score. The resulting factored likelihood allows the model to generalize to an *a-priori* unknown number and types of object instances:

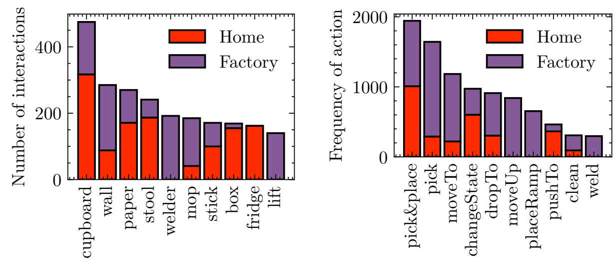
$$\begin{aligned} I_t &= \mathrm{argmax}_{I \in \mathcal{I}} \left(\mathrm{softmax}(W_I[\Omega_t; \tilde{g}_{rel}; \tilde{\eta}_t] + b_I)\right), \\ o_t^1 &= \mathrm{argmax}_{o \in O} \alpha_t^o \\ &= \mathrm{argmax}_{o \in O} \left(\sigma(W_\alpha[\Omega_t; \tilde{g}_{rel}; \tilde{\eta}_t; e_o; \vec{I}_t] + b_\alpha)\right), \\ o_t^2 &= \mathrm{argmax}_{o \in O} \left(\sigma(W_\beta[\Omega_t; \tilde{g}_{rel}; \tilde{\eta}_t; e_o; \vec{I}_t; \alpha_t^o] + b_\beta)\right). \end{aligned}$$

Here α_t^o denotes the likelihood prediction of the first object. The model is trained using a Binary Cross-Entropy loss, with the loss for the three predictors being added independently. Finally, we impose grammar constraints (denoted as Λ) at inference time based on the number of arguments that the predicted interaction I_t accepts. If I_t accepts only one argument only o_t^1 is selected, otherwise both are used. Thus, predicted action, $a_t = f_\theta^{act}(\Omega_t, \tilde{g}_{rel}, \tilde{\eta}_t) = \Lambda[I_t(o_t^1, o_t^2)]$, is then executed by the robot in simulation. The executed action and resulting world state is provided as input to the model for predicting the action at the next time step.

Word Embeddings Informed by a Knowledge Base TANGO uses word embedding function $\mathcal{C}(\cdot)$ that provides

Robot Actions
Push, Climb up/down, Open/Close, Switch on/off, Drop, Pick, Move to, Operate device, Clean, Release material on surface, Push until force
Object Attributes
Grabbed/Free, Outside/Inside, On/Off, Open/Close, Sticky/Not Sticky, Dirty/Clean, Welded/Not Welded, Drilled/Not Drilled, Driven/Not Driven, Cut/Not Cut, Painted/Not Painted
Semantic Relations
On top, Inside, Connected to, Near
Metric Properties
Position, Orientation, Size

Table 1: Domain Representation. Robot symbolic actions, semantic attributes, relations to describe the world state and objects populating the scene in Home and Factory Domains.



(a) Object interactions (b) Symbolic actions

Figure 3: Data set Characteristics. Distribution of plans with plan length for home and factory domains. Frequency of interaction of top 10 objects and frequency of actions for top 10 actions. The collected data set contains diverse interactions in complex spaces.

a dense vector representation for word tokens associated with object class and relation types. Contemporary models use word embeddings acquired from language modeling tasks [Mikolov *et al.*, 2018]. We adopt embeddings that are additionally informed by an existing knowledge graph ConceptNet [Speer *et al.*, 2017] that provides a sufficiently large knowledge graph connecting words with edges expressing relationships such as SimilarTo, IsA, UsedFor, PartOf and CapableOf. Word embeddings [Mikolov *et al.*, 2018] can be *retro-fitted* such that words related using knowledge graph embeddings are also close in the embedding space [Speer *et al.*, 2019]. Using such (pre-trained) embeddings incorporates *general purpose* relational knowledge to facilitate richer generalization for downstream policy learning. The complete sequence of steps is summarized in Figure 2.

5 Data Collection Platform and Annotation

Data Collection Environment. We use PyBullet, a physics simulator [Coumans and Bai, 2016], to generate home and factory-like environments populated with a virtual mobile manipulator (a Universal Robotics (UR5) arm mounted on a Clearpath Husky mobile base). The robot is tasked with goals that involve multiple interactions with objects derived from standardized data sets [Calli *et al.*, 2017]. These goals include: (a) *transporting* objects from one region to another (including space on top of or inside other objects), (b) *fetching* objects, which the robot must reach, grasp and return with, and (c) *inducing state changes* such as illuminating the room or removing dirt from floor. More details in [Bansal *et al.*, 2020]. The set of possible actions and the range of interactions are listed in Table 1. The effects of actions such as pushing or moving are simulated via a motion planner and propagated to the next time step. Abstract actions such as attachment, operating a tool or grasping/releasing objects are encoded symbolically as the establishment or release constraints. The simulation for these actions is coarse and considers their symbolic effects forgoing the exact motion/skills needed to implement them. We assume that the robot can realize abstract actions through low-level routines.

Annotated Corpus. To curate an imitation learning dataset, we recruit human instructors and provide them with goals. They instruct the robot by specifying a sequence of symbolic actions (one at a time) to achieve each goal. Each action is simulated so that they can observe its effects and the new world state. We encourage the instructors to to complete the task as quickly as possible, making use of available tools in the environment. To familiarize them with the simulation platform, we conduct tutorial sessions before data collection. Our resulting dataset consists of diverse plans with different action sets and object interactions. We collected plan traces from 12 human subjects using domain randomization with 10 scenes and 8 semantic goals resulting in a corpus of 708 and 784 plans for the home and factory domains. Figure 3a and 3b show number of interactions with 10 most interacted objects and frequency of 10 most frequent actions respectively. The complete set of objects and goals is given in Table 2. We also perform data augmentation by perturbing the metric states in the human plan trace, performing random replacements of scene objects and validating plan feasibility in simulation. The process results in 3540 and 3920 plans, respectively. Variation was observed in tool use for an instructor for different goals, and within similar goals based on context. The annotated corpus was split as $(75\% : 25\%)$ forming the Training data set and the Test data set to evaluate model accuracy. A 10% fraction of the training data was used as the Validation set for hyper-parameter search. No data augmentation was performed on the Test set.

Generalization Test Set. In order to asses the model's capacity to generalize to unseen worlds, we curate a second test set environments populated by instances of novel object types placed at randomized locations. The following sampling strategies were used: (i) *Position*: perturbing and exchanging object positions in a scene. (ii) *Alternate*: removal of the most frequently used tool in demonstrated plans evaluating the ability to predict the alternative, next best tool to use. (iii) *Unseen*: replacing an object with a similar object, which is not present in training. (iv) *Random*: replacing a tool with a randomly picked object which is *unrelated* to the task. (v) *Goal*: replacing the goals objects. This process resulted in a Generalization Test set with 7460 (goal, plan) pairs.

6 Experiments

Our experiments use the following accuracy metrics for model evaluation: (i) *Action prediction accuracy*: the fraction of tool interactions predicted by the model that matched the human demonstrated action a_t for a given state s_t, and (ii) *Plan execution accuracy*: the fraction of estimated plans that

Domain	Plan lengths	Objects interacted with in a plan	Tools used in a plan	Sample objects	Sample goal specifications
Home	23.25±12.65	4.12±1.97	0.93±0.70	floor[1], wall, fridge[123], cupboard[123], tables[1], couch[1], **big-tray**[1], **tray**[1], **book**[1], paper, cubes, light switch[4], bottle, **box**[2], fruits, **chair**[15], **stick**, dumpster[2], milk carton, shelf[1], **glue**[6], **tape**[6], **stool**[15], **mop**[8], **sponge**[8], **vacuum**[8], dirt[7], door[2]	1. Place milk in fridge, 2. Place fruits in cupboard, 3. Remove dirt from floor, 4. Stick paper to wall, 5. Put cubes in box, 6. Place bottles in dumpster, 7. Place a weight on paper, 8. Illuminate the room.
Factory	38.77±23.17	4.38±1.85	1.44±0.97	floor[1], wall, **ramp**, worktable[1], **tray**[1], **box**[2], crates[1], **stick**, long-shelf[1], **lift**[1], cupboard[123], **drill**[4], **hammer**[49], **ladder**[5], **trolley**[2], **brick**, **blow dryer**[48], **spraypaint**[4], **welder**[4], generator[4], **gasoline**, **coal**, **toolbox**[2], **wood cutter**[4], **3D printer**[4], screw[9], nail[9], **screwdriver**[49], wood, platform[1], oil[7], water[7], board, **mop**[8], **glue**[6], **tape**[6], **stool**[15]	1. Stack crated on platform, 2. Stick paper to wall, 3. Fix board on wall, 4. Turn on the generator, 5. Assemble and paint parts, 6. Move tools to workbench, 7. Clean spilled water, 8. Clean spilled oil.

Table 2: Dataset characteristics. The average plan lengths, number of objects interacted in plan and number of tools used in plans with object and goal sets for Home and Factory domains. Object positions were sampled using Gaussian distribution. Objects in bold can be used as tools. Legend:- [1]: surface, [2]: can open/close, [3]: container, [4]: can operate, [5]: can climb, [6]: can apply, [7]: can be cleaned, [8]: cleaning agent, [9]: can 3D print. Objects in bold can be used as tools. Stool/ladder are objects used to represent a tool for raising the height of the robot.

are successful, i.e., can be executed by the robot in simulation and attain the intended goal (in max. 50 steps). The implementation and data sets used in experiments are available at https://github.com/reail-iitd/tango.

6.1 Comparison with Baselines

We compare to the following three baseline models.

- *ResActGraph* model [Liao *et al.*, 2019], augmented with $\mathrm{FastText}$ embeddings [Mikolov *et al.*, 2018].
- *Affordance-only* baseline inspired from [Hermans *et al.*, 2011] that learns a co-association between tasks and tools, implemented by excluding the graph convolutions and attention from TANGO.
- *Vanilla Deep Q-Learning (DQN)* approach [Bae *et al.*, 2019] that learns purely by interactions with a simulator, receiving positive reward for reaching a goal state.

Table 3 (top half) compares the TANGO model performance with the baseline models. The TANGO model shows a $14-23$ point increase in *Action prediction accuracy* and a $66-71$ points increase in the *Plan execution accuracy* when compared to the *ResActGraph* baseline. Note that the *ResActGraph* model learns a scene representation assuming a fixed and known set of object types and hence can only generalize to new randomized scenes of known objects. In contrast, the TANGO model can not only generalize to randomized scenes with known object types (sharing the GGCN backbone with *ResActGraph*) but can to novel scenes new object types (relying on dense semantic embeddings) and an a-priori unknown number of instances (enabled by a factored likelihood).

The *Affordance-only* baseline model is confined to learning the possible association between a tool object type and the task specified by the human (largely ignoring the environment context). This approach addresses only a part of our problem as it ignores the sequential decision making aspect, where tools may need to be used in sequence to attain a goal.

Finally, the vanilla DQN baseline achieves less than 20% policy accuracy (even after a week of training). In contrast, the TANGO model shows accurate results after training on imitation data for $12-18$ hours. The challenges in scaling can be attributed to the problem size (≈1000 actions), long plans, sparse and delayed rewards (no reward until goal attainment).

Next, we assess the zero-shot transfer setting, i.e., whether the model can perform common sense generalization in worlds with new objects unseen in training. The same table shows that the plans predicted by TANGO lead to an increase of up to 56 points in plan execution accuracy on Generalization Test set over the best-performing baseline model. This demonstrates accurate prediction and use of unseen tool objects for a given goal. Specifically, in the home domain, if the *stool* is not present in the scene, the model is able to use a *stick* instead to fetch far-away objects. Similarly, if the robot can predict the use of a box for transporting objects even if it has only seen the use of a tray for moving objects during training. The *ResActGraph* model is unable to adapt to novel worlds and obtains zero points in several generalization tests.

The poorer performance of the *Affordance-only* model can again be attributed to the fact that planning tool interactions involves sequential decision-making. Even if the robot can use affordance similarity to replace a *tray* object with a *box*, it still needs to predict the opening of the *box* before placing an item in its plan for a successful execution. This is corroborated by the drop in performance for the *Unseen* generalization tests for this model by 52.3 points. Finally, the vanilla DQN model lacks a clear mechanism for transferring to novel settings, hence shows poor generalization in our experiments.

6.2 Ablation Analysis of Model Components

We analyze the importance of each component of the proposed model by performing an ablation study. Table 3 (lower half) presents the results. For a fair comparison, the model capacities remain the same during the ablation experiments.

The model builds on the *GGCN* environment representation encoding the inter-object and agent-object relational properties. The ablation of the GGCN component results in a reduction of 22% in the generalization accuracy in the factory domain (where tools may be placed at multiple levels in the factory). The inclusion of this component allows the robot to leverage relational properties such as OnTop to predict the use of tools such as a ramp to negotiate an elevation or a stick to fetch an object immediately beyond the manipulator's reach.

The *Metric* component encodes the metric properties of objects in the scene such as positions, size etc. Experiments demonstrate its effectiveness in prediction tool interactions

Model	Action Prediction		Plan Execution		Generalization Plan Execution Accuracy						
	Home	Factory	Home	Factory	Home (Avg)	Factory (Avg)	Position	Alternate	Unseen	Random	Goal
Baseline (ResActGraph)	27.67	45.81	26.15	0.00	12.38	0.00	0.00	0.00	0.00	25.10	9.12
Affordance Only	46.22	52.71	52.12	20.39	44.10	4.82	17.84	47.33	29.31	29.57	34.85
DQN	-	-	24.82	17.77	15.26	2.23	0.00	0.00	12.75	9.67	4.21
TANGO	59.43	60.22	92.31	71.42	**91.30**	**60.49**	**93.44**	**77.47**	**81.60**	59.68	**59.41**
Model Ablations											
- GGCN (World Representation)	59.43	60.59	84.61	27.27	78.02	38.70	70.42	58.79	60.00	56.35	38.64
- Metric (World Representation)	58.8	60.84	84.61	62.34	72.42	51.83	59.68	67.19	60.79	**84.47**	21.70
- Goal-Conditioned Attn	53.14	60.35	53.85	11.69	37.02	8.80	35.33	15.05	32.14	41.67	6.51
- Temporal Action History	45.91	49.94	24.61	0.00	8.55	0.00	0.00	0.00	0.00	30.56	1.15
- Factored Likelihood	61.32	**61.34**	**95.38**	**85.71**	34.22	43.44	90.50	14.82	30.65	64.67	53.26
- ConceptNet	**63.52**	60.35	89.23	57.14	81.86	56.97	82.33	68.61	74.57	65.73	47.92

Table 3: A comparison of *Action prediction* and *Plan execution* accuracies for the baseline, the proposed TANGO model, and ablations. Results are presented for test and generalization data sets (under five sampling strategies) derived from the home and factory domains.

based on relative sizes of interacting objects. E.g., the model successfully predicts that *fruits* can be transported using a *tray* but larger *cartons* require a *box* for the same task. The ablation of this component leads to a reduction of 10.2 points in the *Alternate* generalization tests as the ablated model unable to adapt the tool when there are unseen objects with different sizes than those seen during training.

Next, we assess the impact of removing the *Goal-Conditioned Attention component*. This experiment shows a a significant reduction (≈ 50 points) in the *Plan execution accuracy* on the Generalization Test set, particularly in scenes with a large number of objects. The attention mechanism allows learning of a restricted context of tool objects that may be useful for attaining the provided goal, in essence, filtering away goal-irrelevant objects populating the scene. Additionally, note that the inclusion of this component allows tool predictions to be *goal-aware*. Consequently, we observe ablating this component leads to a reduction of 53 points in the *Goal* generalization test set where the goal objects are perturbed.

The *Action History* component utilizes the agent's past interactions for the purpose of predicting the next tool interaction. The inclusion of this component allows learning of correlated and commonly repeated action sequences. For instance, the task of exiting from a room typically involves a plan fragment that includes moving to a door, opening it and exiting from the door and are commonly observed in a number of longer plans. The ablation of this component leads to erroneous predictions where a particular action in a common plan fragment is missing or incorrectly predicted. E.g., a robot attempting to pick an object inside an enclosure without opening the lid. In our experiments, we observe that ablating the model leads to a significant decrease in goal reach-ability, causing a 70 point decrease in the *Plan Execution accuracy* and 72 point drop in the *Generalization accuracy*.

The need for generalization to novel scenes implies that our model cannot assume a fixed and a-priori known set of objects that the robot can interact with. Generalization to an arbitrary number of objects in the scene is accomplished by by factoring model predictions over individual objects in a recurrent manner. Ablating the factored likelihood components results in a simpler model that performs predictions over a known fixed-size object set. The simplified model displays a higher action-prediction and plan-execution accuracies in known world. Crucially, we observe that ablating this component results in a significant decrease of 51 and 63 points in

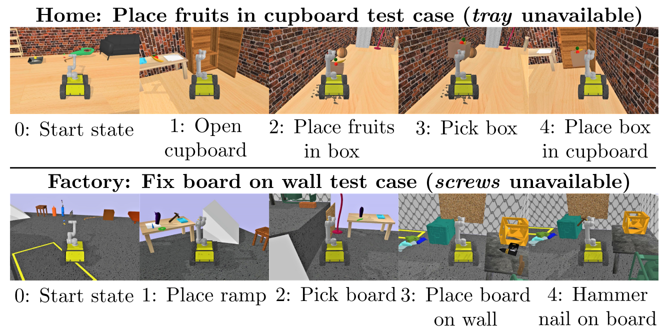


Figure 4: A simulated robot manipulator uses TANGO to synthesize tool interactions in novel contexts with unseen objects.

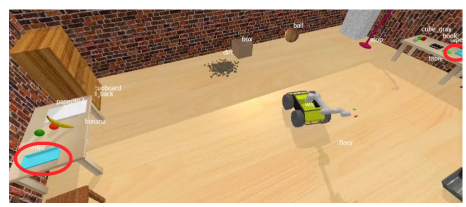


Figure 5: The model predicts the instance of *tray* (on the left) which is closer to the *fruits* (goal objects) than the one on the right.

the *Unseen* and the *Alternate* generalization test sets.

Finally, the *ConceptNet embeddings* are important for semantic generalization to unseen tool types. We replace ConceptNet embeddings with FastText [Mikolov *et al.*, 2018] embeddings in the *-ConceptNet* model to show their importance. The *-ConceptNet* model shows poorer generalization (6.5% decrease) as it models word affinity as expressed in language only. ConceptNet embedding space models relational affinity between objects as present in the knowledge-base.

6.3 Analysis of Resulting Plans

Figure 4 shows the robot using the learned model to synthesize a plan for a declarative goal. Here, if the goal is to transport fruits and human demonstrates usage of *tray* and the model never sees *box* while training. TANGO uses *box* in a scene where tray is absent, showing that it is able to predict semantically similar tools for task completion. Similarly, for the goal of fixing a board on the wall, if humans use *screws* the agent uses *nails* and *hammer* when screws are absent from the scene. Figure 5 shows how the model uses the position information of tool objects to predict the tool closer to the

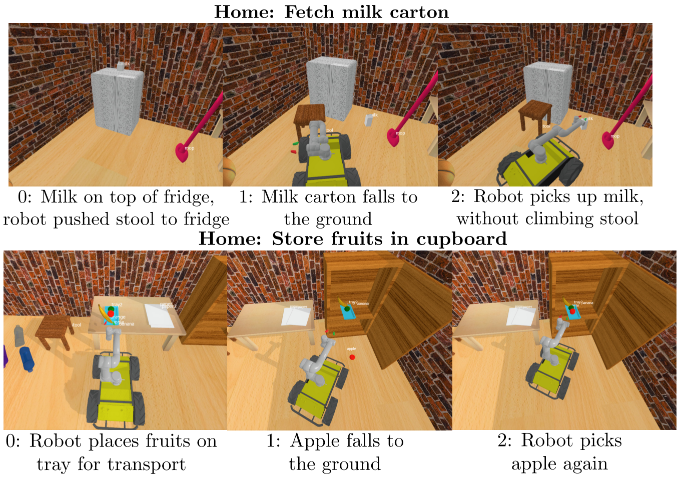


Figure 6: Interleaved action prediction and execution enables adaptation in case of unexpected errors during action execution.

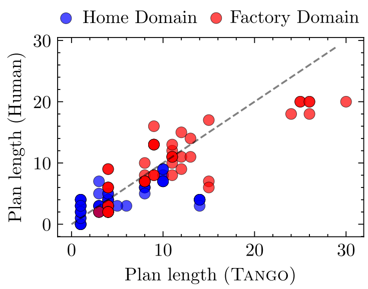


Figure 7: Scatter plot comparing the lengths of plans obtained from model predictions and those from human demonstrations.

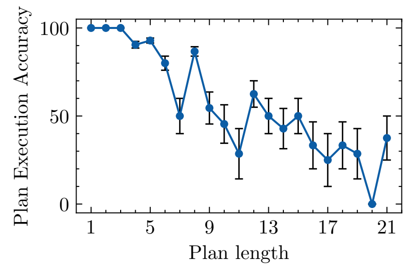


Figure 8: Execution accuracy of inferred plans with plan length.

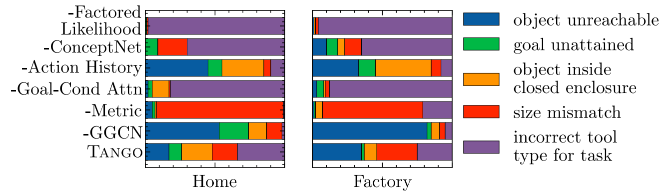


Figure 9: An analysis of fractional errors during plan execution using the learned TANGO model.

goal object or the agent. The world representation encodes the metric properties of objects (position and orientation) that allows the robot to interact with nearer tool objects.

Figure 6 shows the robustness to unexpected errors and stochasticity in action execution. Consider the task of "fetching a carton", where the milk carton is on an elevated platform, the model predicts the uses a stool to elevate itself. The carton falls due to errors during action execution. Following which, the robot infers that the stool is no longer required and directly fetches the carton. Similarly, for the task of "storing away the fruits in the cupboard", the robot predicts the use of tray for the transport task. During execution the apple falls off the tray. The robot correctly re-collects the apple.

Figure 7 compares the length of robot plans predicted by the learned model against human demonstrated plans. We observe that, on average, the predicted plan lengths are close to the human demonstrated ones. In 12% cases, the plans predicted by TANGO utilize tools satisfying the goal condition in fewer steps compared to the human demonstrated plan.

7 Limitations and Future Work

Figure 8 assesses the model accuracy with the lengths of the inferred plans. We observe that the plan execution accuracy decreases by 20% on the Test sets and 30% on Generalization Test sets. This merits investigation into planning abstractions [Vega-Brown and Roy, 2020] for scaling to longer plan lengths in realistic domains. Figure 9 analyzes the errors encountered during plan execution using actions predicted by the proposed model. In 27% of the cases, the model misses a pre-requisite actions required for a pre-condition for initiating the subsequent action. For example, missing the need to open the door before exiting the room (object unreachable 19%) or missing opening the cupboard before picking an object inside it (object inside enclosure 8%). There is scope for improvement here by incorporating explicit causal structure [Nair *et al.*, 2019]. Finally, we will explore ways to incorporate the uncertainty in the symbolic state (arising from physical sensors) and extend the model to partially-known environments.

8 Conclusions

This paper proposes TANGO, a novel neural architecture that learns a policy to attain intended goals as tool interaction sequences leveraging fusion of semantic and metric representations, goal-conditioned attention, knowledge-base corpora. TANGO is trained using a data set of human instructed robot plans with simulated world states in home and factory like environments. The imitation learner demonstrates accurate commonsense generalization to environments with novel object instances using the learned knowledge of shared spatial and semantic characteristics. It also shows the ability to adapt to erroneous situations and stochasticity in action execution. Finally, TANGO synthesizes a sequence of tool interactions with a high accuracy of goal-attainment.

Acknowledgements

Mausam is supported by an IBM SUR award, grants by Google, Bloomberg and 1MG, Jai Gupta chair fellowship, and a Visvesvaraya faculty award by Govt. of India. Rohan Paul acknowledges support from Pankaj Gupta Faculty Fellowship and DST's Technology Innovation Hub (TIH) for Cobotics. We thank the IIT Delhi HPC facility and Prof. Prem Kalra and Mr. Anil Sharma at the CSE VR Lab for compute resources.

References

[Allen *et al.*, 2019] Kelsey R Allen, Kevin A Smith, and Joshua B Tenenbaum. The tools challenge: Rapid trial-and-error learning in physical problem solving. *arXiv preprint arXiv:1907.09620*, 2019.

[Bae *et al.*, 2019] Hyansu Bae, Gidong Kim, Jonguk Kim, Dianwei Qian, and Sukgyu Lee. Multi-robot path planning method using reinforcement learning. *Applied Sciences*, 9(15):3057, 2019.

[Bahdanau *et al.*, 2014] D Bahdanau, K Cho, and Y Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Bansal *et al.*, 2020] Rajas Bansal, Shreshth Tuli, Rohan Paul, and Mausam. Toolnet: Using commonsense generalization for predicting tool use for robot plan synthesis. In *Workshop on Advances & Challenges in Imitation Learning for Robotics at Robotics Science and Systems (RSS)*, 2020.

[Bisk *et al.*, 2020] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

[Boteanu *et al.*, 2015] Adrian Boteanu, David Kent, Anahita Mohseni-Kabir, Charles Rich, and Sonia Chernova. Towards robot adaptability in new situations. In *2015 AAAI Fall Symposium Series*. Citeseer, 2015.

[Calli *et al.*, 2017] B Calli, A Singh, J Bruce, A Walsman, K Konolige, S S Srinavasa, P Abbeel, and A M Dollar. YCB Benchmarking Project: Object Set, Data Set and Their Applications. *Journal of The Society of Instrument and Control Engineers*, 56(10):792–797, 2017.

[Coumans and Bai, 2016] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. *GitHub repo*, 2016.

[Finn *et al.*, 2017] C Finn, T Yu, T Zhang, P Abbeel, and S Levine. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*, 2017.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[Hermans *et al.*, 2011] Tucker Hermans, James M Rehg, and Aaron Bobick. Affordance prediction via learned object attributes. In *ICRA: Workshop on Semantic Perception, Mapping, and Exploration*, pages 181–184, 2011.

[Holladay *et al.*, 2019] Rachel Holladay, Tomás Lozano-Pérez, and Alberto Rodriguez. Force-and-motion constrained planning for tool use. In *IROS*, 2019.

[Huang *et al.*, 2019] De-An Huang, Suraj Nair, Danfei Xu, Yuke Zhu, Animesh Garg, Li Fei-Fei, Silvio Savarese, and Juan Carlos Niebles. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. In *CVPR*, pages 8565–8574, 2019.

[Jain *et al.*, 2015] Ashesh Jain, Debarghya Das, Jayesh K Gupta, and Ashutosh Saxena. Planit: A crowdsourcing approach for learning to plan paths from large scale preference feedback. In *ICRA*, pages 877–884, 2015.

[Kho *et al.*, 2014] Gabriel Kho, Christina Hung, and Hugh Cunningham. Robo brain: Massive knowledge base for robots. *Cornell Univ., USA, Tech. Rep*, 2014.

[Kolobov *et al.*, 2011] Andrey Kolobov, Mausam, Daniel S. Weld, and Hector Geffner. Heuristic search for generalized stochastic shortest path mdps. In *ICAPS*, 2011.

[Liao *et al.*, 2019] Yuan-Hong Liao, Xavier Puig, Marko Boben, Antonio Torralba, and Sanja Fidler. Synthesizing environment-aware activities via activity sketches. In *CVPR*, pages 6291–6299, 2019.

[Mausam and Kolobov, 2012] Mausam and Andrey Kolobov. Planning with Markov decision processes: An AI perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–210, 2012.

[Mikolov *et al.*, 2018] T Mikolov, E Grave, P Bojanowski, C Puhrsch, and A Joulin. Advances in pre-training distributed word representations. In *LREC*, 2018.

[Misra *et al.*, 2016] Dipendra K Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *IJRR*, 35(1-3):281–300, 2016.

[Nair *et al.*, 2019] Suraj Nair, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv:1910.01751*, 2019.

[Nyga *et al.*, 2018] Daniel Nyga, Subhro Roy, Rohan Paul, Daehyung Park, Mihai Pomarlan, Michael Beetz, and Nicholas Roy. Grounding robot plans from natural language instructions with incomplete world knowledge. In *Conference on Robot Learning*, pages 714–723, 2018.

[Puig *et al.*, 2018] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018.

[Speer *et al.*, 2017] R Speer, J Chin, and C Havasi. ConceptNet 5.5: An open multilingual graph of general knowledge. In *AAAI*, 2017.

[Speer *et al.*, 2019] R Speer, J Chin, and C Havasi. ConceptNet Numberbatch, the best pre-computed word embeddings you can use. *GitHub repository*, 2019.

[Toussaint *et al.*, 2018] M Toussaint, K Allen, KA Smith, and JB Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *RSS*, 2018.

[Vega-Brown and Roy, 2020] William Vega-Brown and Nicholas Roy. Asymptotically optimal planning under piecewise-analytic constraints. In *Algorithmic Foundations of Robotics XII*, pages 528–543. Springer, 2020.