Learning Temporal Task Models from Human Bimanual Demonstrations

Christian R. G. Dreher and Tamim Asfour

***Abstract*— Learning temporal relations between actions in a bimanual manipulation task is important for capturing the constraints of actions required to achieve the task's goal. However, given several demonstrations of a bimanual manipulation task, the problem of identifying the true temporal dependencies between actions – if there are any – is very challenging due to contradictions. We propose a model-driven approach for learning temporal task models from multiple bimanual human demonstrations that represents temporal relations on two levels. First, temporal relations between sets of actions that exhibit a tight temporal coupling, and second, temporal relations between these sets of actions. We build on Allen's interval algebra as a representation to express relations between temporal intervals. Semantically defining these interval relations allows us to soften their formulation to deal with inaccuracies in real data obtained when observing humans demonstrating the task. Our temporal task models can be learned incrementally from multiple modalities, and allow us to reason about viable alternatives during task execution in case of unexpected events. We evaluated the approach quantitatively on two datasets and qualitatively on a humanoid robot. The evaluation shows how inherent properties of bimanual human manipulation tasks can be exploited to derive a model useful for the reproduction by humanoid robots.**

I. Introduction

For a human, a humanoid robot is supposed to assume the role of another fellow human with human-like capabilities in order to be a useful help in common tasks, or completely take care of them. Humans often transfer knowledge about a task simply by showing it to others, and correcting the other during imitation if needed, for example kinesthetically, verbally, or by showing the task again. It is reasonable to expect such capabilities from a humanoid robot interacting with a human, too, especially since (re-)programming the robot by non-experts is generally not feasible. In robotics, the discipline endeavored to endow robots with such capabilities is referred to as *Programming by Demonstration* [1]. One integral component in the programming by demonstration cycle is concerned with the question of how to model what the robot perceives in such a way that it can reproduce the task at hand. These *task models* are representations of tasks, encoding all essential information needed to successfully execute the task in novel situations and contexts. Especially symbolic or geometric constraints to be derived from demonstration are important since they must be obeyed to ensure a successful task execution. Apart from that, these constraints are essential for generalization to novel situations.

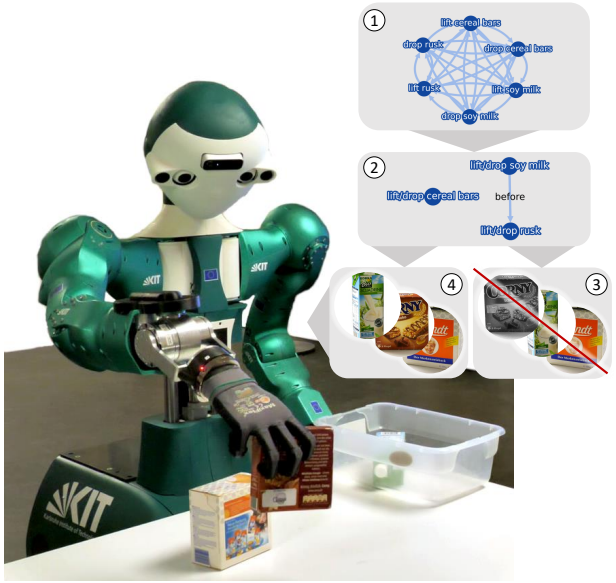


Figure 1. ARMAR-6 solving a clean-up task in a way not seen before in human demonstrations. A temporal task model ① is used to infer temporal constraints between actions ②. In this scenario, the robot could initially not locate the brown box of cereal bars, discarding the initial plan ③ and preferring an unseen action order instead ④, which is in compliance with the derived temporal constraints.

Consider for example the task of making muesli by pouring some cereals, milk, as well as banana slices into a bowl. The order in which the ingredients are poured into the bowl does not matter, as long as the banana has been cut into slices before pouring them into the bowl. Let's assume that the robot has a complete temporal task model that represent actions and their temporal relations, and it observes a human preparing muesli by cutting a banana, the robot could then proactively help the human by already pouring cereals or milk into the bowl. The robot can also make use of the learned temporal constraints to mitigate problems during the execution by replanning the order of actions to be executed. For example, if the robot temporally lost track of the milk due to occlusions, it might just fall back to pour the cereals instead until the bottle of milk is visible again. A similar problem is depicted in Figure 1.

In this work, we explore the question of how to model temporal relations between actions in human bimanual demonstrations in order to extract temporal constraints that are useful for the reproduction of the task on a humanoid robot. We propose a model-driven approach for learning and inferring relevant temporal information between actions

The research leading to these results has been funded by the Carl Zeiss Foundation in the projects AgiProbot and JuBot.

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. {c.dreher, asfour}@kit.edu

in demonstrated tasks by exploiting inherent properties of them. Specifically, we show how the decomposition of a task into subtasks can help identifying temporally tightly coupled human bimanual manipulation actions.

Our contributions are: (i) A temporal task model that is used to identify true temporal relations of actions from human demonstration for the reproduction of a task, (ii) a softened formulation of temporal relations that is usable for real-world data, (iii) an evaluation of the formulation of our temporal task model on two new publicly available datasets for learning temporal relations of complex manipulation actions, one manually labeled, one synthetic, as well as on the humanoid robot ARMAR-6, (iv) a synthetic dataset creation tool, datasets and source code for the implementation of the temporal task model are publicly available under an open-source license.

II. Related Work

Pivotal works on temporal relations were published by Allen in [2] and [3], who introduced a calculus to reason about all possible temporal relations two intervals can have, resulting in 13 temporal relations such as *before*, *during*, etc. A formal definition of this calculus follows later in Section III. These relations are often used in the context of temporal planning, however, the formalism is generally defined on intervals, and can be applied to non-temporal domains as well, such as modeling spatial occlusions [4]. For many real-world applications, Allen relations, which are defined on crisp intervals (meaning intervals whose elements have a definitive membership), are not flexible or expressive enough. To circumvent this, many extensions to Allen relations were proposed, defined on fuzzy time intervals, e. g., by Schockaert et al. [5]. In this work, we also build upon Allen relations as a way to represent temporal task knowledge.

We structure the related work in two parts. First, we discuss works including action precedence relations and action ordering, i. e., works that do not consider concurrent actions or multiple agents in Section II-A. Second, works considering complex and concurrent temporal relations are addressed in Section II-B.

A. Precedence Relations and Action Ordering

Nicolescu and Matatrić [6] proposed a system, which is able to generalize the precedence relations of actions across several demonstrations by finding the longest common sequence. Alternative sequences are represented in a directed acyclic graph, and new demonstrations can be added incrementally. The purpose of this system was to identify falsely detected actions in demonstrated tasks, not to solve the problem of finding precedences between actions. Ekvall and Kragić [7] presented an approach for temporal constraint generalization by identifying precedence relations and eliminating contradicting precedence relations from several demonstrations. A similar approach was followed by Pardowitz et al. [8], where Precedence Graphs were proposed to model observed action precedence relations. Here, nodes represent the actions, and a directed edge denotes a precedence relation. When a new demonstration is observed, which contradicts the precedence graph, the corresponding edge is removed from the graph, thus lifting the constraint. Both works follow a version space [9] approach by excluding contradictions in the transitive hull of precedence relations of all actions as they are observed in a task. Similarly, Kramberger et al. [10] tackle the problem of learning precedence constraints in assembly scenarios in combination with geometrical constraints.

Xiong et al. [11] propose the use of And-Or graphs to model spatial, causal, and temporal constraints. For temporal relations, either an *and* relation or an *or* relation between actions is defined. Here, an *and* relation is a precedence relation, while an *or* relation denotes mutual exclusiveness (i. e., the actions are conflicting). A stochastic framework was used to populate And-Or graphs with new demonstrations. Racca and Kyrki [12] propose a probabilistic approach to incrementally learn user-preferred task orders using a Dirichlet-multinomial model to learn Markov chain parameters. While demonstrating a task, the robot is able to ask questions about optional actions (so-called frequency queries) or to get the preferred action precedences of the user (so-called disambiguation queries). The authors provide a mapping of answers to these queries onto the Dirichlet-multinomial model to get an updated set of Markov chain parameters.

Especially the works based on version space approaches require perfect data, as they work on the assumption that contradictions imply that constraints can be lifted. In reality, contradictions may simply emerge from errors, whether they come from humans themselves during demonstration, or from a preprocessing step of the data. Additionally, all these works have in common, that they only consider precedence relations, i. e., no concurrency between actions. Thus, the inherent concurrency in human bimanual manipulation tasks cannot be represented with such models. Compared to these works, we consider all possible temporal relations between each pair of actions in a task, especially concurrent actions, to allow modeling bimanual demonstrations.

B. Complex and Concurrent Temporal Relations

Allen builds a temporal model represented as a graph, referred to as a *network*, where nodes represent actions, and edges all possible temporal relations [2], [3]. As already mentioned, these 13 relations (details in Section III) fully describe how two temporal intervals qualitatively relate to each other. The work aimed at the iterative construction of graphs, which capture all temporal relations that hold between a set of intervals. For this, a version-space-based approach was proposed, deducing new relations after new demonstrations, referred to as *facts*, are added using a transitivity table. In the context of temporal planning, Rossi et al. learn local temporal preferences of to-be scheduled activities in order to get better-suited plans without posing hard constraints to the planning problem. These preferences are modeled with functions, so-called preference functions, and pose soft temporal constraints which can introduce quantitative constraints [13], [14]. Talukdar et al. use a large text corpus to reason about temporal constraints between binary predicates.

The considered constraints are *before* or *simultaneous*, and are evaluated under the assumption that particular verbs in text indicate certain temporal relations [15].

In the context of robotics, Asfour et al. detect keypoints in bimanual manipulation tasks. Common keypoints across several demonstrations are used to infer temporal relations for the reproduction and temporal coordination of trajectories for the left and the right hand [16]. Ye et al. propose Manipulation Precedence Graphs, a graph structure very similar to other works [7], [8]. Their approach considers, which actions can be executed in parallel by finding nodes in the precedence graph, which do not have any incoming edges. This approach, however, only works under the assumption that the actions are both temporally and spatially independent, and also that each end-effector is able to perform all actions independent of the other end-effectors. Thus, the model cannot capture if actions *must* be executed concurrently, or any other specific degree of concurrency [17]. Carpio et al. [18] propose an n-gram-based model considering the current temporal context given via Allen relations to reason about the next action.

Similar to the version space approaches for sequential actions, the formulation of Allen assumes perfect data. For the work of Rossi et al., this is also the case. Compared to that, we build a temporal task model by encoding temporal relations observed between actions in a human demonstration, reasoning about which relations most likely hold. This is similar to the idea of Talukdar et al., except that information from human demonstration is much sparser than large text corpora. Compared to Ye et al., our model fully captures the qualitative temporal arrangement between actions. Contrary to Carpio et al., we infer true temporal relations from the complete set of all demonstrations, building a temporal task model, which allows to replan and reevaluate the next actions of the robot during execution in case of unexpected events.

III. Background

In this work, we define temporal relations following Allen's formulation [2], also referred to as "Allen's interval algebra", or "Allen relations". These are 13 relations on intervals, 7 basic ones, namely *before*, *meets*, *overlaps*, *starts*, *finishes*, *during*, *equals*, as well as 6 inverse relations, namely *after*, *met by*, *overlapped by*, *started by*, *finished by*, *contains*. The *equals* relation is symmetric and thus its own inverse.

An interval is a tuple $i = (i_s, i_e)$ with $i_s, i_e \in \mathbb{R}$ and $i_s < i_e$. Here, i_s is the start of the interval i, and i_e its end. The 13 Allen relations on intervals x and y can then be defined as shown in Table I. Further, the size s of an interval is defined as $s(i) = i_e - i_s$. The set of all intervals is denoted as I. In the following we will use the notation $R(x, y) \Leftrightarrow c$ as a short-hand to express that two intervals $x, y \in I$ are part of the relation R iff the condition c holds.

Instead of the definition provided in Table I, we semantically define the 7 basic Allen relations on intervals x and y over 8 point relations, precisely on interval starts and ends, as shown in Table II. The inverse Allen relations are defined through their respective basic counterparts. These point relations can be defined as shown in Table III to get an equivalent point-based formulation of the Allen relations from Table I. Later we will redefine the meaning of these 8 point relations to cope with the properties of real data. Theoretically, only 6 point relations are needed, since the relations *x starts before y starts*(x, y) and *x starts after y starts*(y, x), as well as the relations *x ends before y starts*(x, y) and *x ends after y starts*(y, x) express the same condition respectively, and their equivalence is given by swapping the arguments. Thus, one has *x starts before y starts*(x, y) = *x starts after y starts*(y, x), and *x ends before y starts*(x, y) = *x ends after y starts*(y, x). For the sake of clarity, and similarly to the inverse Allen relations, two redundant point relations were defined for these cases.

TABLE I

The 13 temporal relations on intervals after Allen [2].

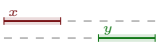
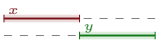
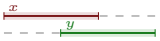
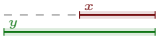
Allen Relation Inverse Relation		Condition	Example
before(x, y) *after*(y, x)	$\Leftrightarrow$	$x_e < y_s$	
meets(x, y) *met by*(y, x)	$\Leftrightarrow$	$x_e = y_s$	
overlaps(x, y) *overlapped by*(y, x)	$\Leftrightarrow$	$x_s < y_s \wedge x_e < y_e$ $\wedge\ x_e > y_s$	
starts(x, y) *started by*(y, x)	$\Leftrightarrow$	$x_s = y_s \wedge x_e < y_e$	
finishes(x, y) *finished by*(y, x)	$\Leftrightarrow$	$x_s > y_s \wedge x_e = y_e$	
during(x, y) *contains*(y, x)	$\Leftrightarrow$	$x_s > y_s \wedge x_e < y_e$	
equals(x, y) *equals*(y, x)	$\Leftrightarrow$	$x_s = y_s \wedge x_e = y_e$	

TABLE II

Semantic definition of the 7 basic Allen relations.

#	Allen Relation		Condition
1	*before*(x, y)	$\Leftrightarrow$	*x ends before y starts*(x, y)
2	*meets*(x, y)	$\Leftrightarrow$	*x ends when y starts*(x, y)
3	*overlaps*(x, y)	$\Leftrightarrow$	*x starts before y starts*(x, y) $\wedge$ *x ends before y ends*(x, y) $\wedge$ *x ends after y starts*(x, y)
4	*starts*(x, y)	$\Leftrightarrow$	*x starts when y starts*(x, y) $\wedge$ *x ends before y ends*(x, y)
5	*finishes*(x, y)	$\Leftrightarrow$	*x starts after y starts*(x, y) $\wedge$ *x ends when y ends*(x, y)
6	*during*(x, y)	$\Leftrightarrow$	*x starts after y starts*(x, y) $\wedge$ *x ends before y ends*(x, y)
7	*equals*(x, y)	$\Leftrightarrow$	*x starts when y starts*(x, y) $\wedge$ *x ends when y ends*(x, y)

TABLE III

The 8 point relations used to define the Allen relations.

#	Point Relation		Condition
1	*x starts before y starts*(x, y)	$\Leftrightarrow$	$x_s < y_s$
2	*x starts when y starts*(x, y)	$\Leftrightarrow$	$x_s = y_s$
3	*x starts after y starts*(x, y)	$\Leftrightarrow$	$x_s > y_s$
4	*x ends before y starts*(x, y)	$\Leftrightarrow$	$x_e < y_s$
5	*x ends before y ends*(x, y)	$\Leftrightarrow$	$x_e < y_e$
6	*x ends when y starts*(x, y)	$\Leftrightarrow$	$x_e = y_s$
7	*x ends when y ends*(x, y)	$\Leftrightarrow$	$x_e = y_e$
8	*x ends after y starts*(x, y)	$\Leftrightarrow$	$x_e > y_s$

IV. Approach

We will present our approach to build temporal task models from real data. We start with the concrete problem formulation in Section IV-A and describe how we derive temporal relations suitable for real data, where actual temporal equality is hardly observed in Section IV-B. Using these sets of relations, we show in Section IV-C, how a temporal task model describing temporal relations between actions in a task is built up. Finally, in Section IV-D we conclude by showing how subtasks, temporally tightly coupled sets of actions a task is composed of, can be identified from such a temporal task model.

A. Problem Formulation

An *action* in the context of this work is an elementary manipulation movement and always intertwined with an object. For example, we consider the actions *pour milk* and *pour water* to be different. This assumption is important to establish a temporal task model and is motivated by the concept of Object-Action-Complexes [19], which postulate that movements alone do not describe a manipulation action, but that an action is defined by a movement applied to an object, a fundamental aspect to consider.

A *task* is a set of actions, which need to be performed on objects in order to achieve the task goal. A task consists of one or many *subtasks*, which are elementary building blocks of a task. We assume in this work, that actions inside one subtasks may feature concurrent actions, but that the temporal relations between subtasks are either precedence relations such as *before*, *after*, *meets*, or *met by* or none at all, i. e., the execution order does not matter. A task aims at achieving a specific goal, e. g., the task *prepare muesli*, with the goal to have a prepared muesli ready. A subtask achieves a subgoal, which contributes towards this goal but is usually not meaningful on its own, e. g., the subtask *cut banana* to have banana slices to put it in a muesli as an ingredient.

A *demonstration* of a bimanual manipulation task is a two-track sequence of observed actions, one action sequence for each hand. Here, an action is always associated with a temporal segment, i. e., an interval with a temporal start and end point. To account for non-manipulation actions (such as approaching an object, retreating from an object, ...), as well as for potentially incomplete segmentation or classification of an action recognition system, we do not assume that the temporal segmentation is complete. Additionally, we assume that one hand can only execute one action at a time, hence the temporal relations between actions of one such track (e. g., for the left or right hand) are purely precedence relations (*before* or *meets*) and do not overlap. Theoretically, a demonstration can be extended to multi-track (or multi-agent) demonstrations, for example, to also account for platform movement in mobile manipulation scenarios.

In this work, we address the problems of building a model of temporal constraints between actions from multiple demonstrations, a *temporal task model* and using it to infer true temporal relations in subtasks (intra-subtask temporal relations) and between subtasks (inter-subtask temporal relations). The temporal constraints are based on Allen relations and should be able to capture bimanual manipulation tasks. These temporal task models allow identifying subtasks in human demonstrations, which are essential for the reproduction on a robot system. We represent the temporal task model as a fully-connected directed graph $G_n = (V, E)$, where the nodes V represent actions, and edges E the temporal relations between actions. Specifically, each edge $(a_1, a_2) \in E$ tracks the absolute frequencies of occurrence of the temporal relations, which were observed between the actions $a_1 \in V$ and $a_2 \in V$ across all demonstrations. This is done by assigning edge attributes to each edge for each temporal relation. Inferring true temporal relations between actions requires us to discard edges that show contradictory relations, and to *fix* exactly one temporal relation for each remaining edge given the frequencies of occurrence from all demonstrations. This means that we commit to one temporal relation from the distribution of all relations observed between a given pair of actions. This problem is hard, because naively choosing the most likely relation may lead to contradictory constraints.

B. Soft Allen Relations

When learning temporal task models from human demonstration, temporal constraints should be extracted from real sensor data which does not allow the exact identification of the starts and ends of actions. This makes the Allen relations not suitable to model temporal constraints in real-world applications.

To address this problem, we soften the Allen relations by allowing for a margin m wherein co-occurring starts or ends of intervals are still considered to be simultaneous. We derive the Allen relations from the semantic definition from Table II together with softened definitions of the 8 point relations from Table III using an interval-size-normalized margin $m_n(x, y) = \min(s(x), s(y), m)$. The softened conditions for the 8 point relations are shown in Table IV. With this approach, we change the meaning of what is considered *equal*, accounting for various sources of imprecision when dealing with intervals derived from real data.

TABLE IV
Redefinition of the 8 point relations to soften equality.

#	Point Relation		Condition
1'	*x starts before y starts*(x, y)	$\Leftrightarrow$	$y_s - x_s > m_n(x, y)$
2'	*x starts when y starts*(x, y)	$\Leftrightarrow$	$\lvert x_s - y_s \rvert \leq m_n(x, y)$
3'	*x starts after y starts*(x, y)	$\Leftrightarrow$	$x_s - y_s > m_n(x, y)$
4'	*x ends before y starts*(x, y)	$\Leftrightarrow$	$y_s - x_e > m_n(x, y)$
5'	*x ends before y ends*(x, y)	$\Leftrightarrow$	$y_e - x_e > m_n(x, y)$
6'	*x ends when y starts*(x, y)	$\Leftrightarrow$	$\lvert x_e - y_s \rvert \leq m_n(x, y)$
7'	*x ends when y ends*(x, y)	$\Leftrightarrow$	$\lvert x_e - y_e \rvert \leq m_n(x, y)$
8'	*x ends after y starts*(x, y)	$\Leftrightarrow$	$x_e - y_s > m_n(x, y)$

In our work, we empirically set $m = 330\,\mathrm{ms}$. Note that a margin $m = 0$ would recreate the behavior of the original definitions of the Allen relations given in Table III. For small intervals, which are roughly the size of the margin, this approach might result in more than one relation that holds. In this case, we re-evaluate the possible relations after reducing the margin by one. We do this recursively until only

one temporal relation holds. This happens at the latest when $m = 0$. As mentioned, this is then equivalent to evaluating the original Allen relations, which are mutually exclusive given a concrete pair of intervals, and thus always unambiguous.

C. Building a Temporal Task Model from Demonstrations

As posed in Section IV-A, we define a temporal task model as a graph $G_n = (V, E)$, where the nodes V represent actions, and edges E track the absolute frequencies of occurrence of the temporal relations, which were observed between the actions. Note that the handedness will be abstracted away in the temporal task model. This allows for better generalization between left and right-handed demonstrators, and better data efficiency since we can assume that bimanual robots are ambidextrous, and we think that a task representation should abstract from human motor limitations.

Consider for example the case where 10 demonstrations of a muesli-making task were observed. In 7 cases, the human demonstrator *poured the cereals* into a bowl *before* they *poured the milk* into it, while in the remaining 3 cases they *poured the milk* first. Then the edge attributes for the edge (*pour cereals*, *pour milk*) in the graph would exactly reflect these numbers: *before*: 7, *after*: 3, *meets*: 0, ..., *equals*: 0. Similarly, the contrary edge (*pour milk*, *pour cereals*) would reflect the inverse: *before*: 3, *after*: 7, *meets*: 0, ..., *equals*: 0.

The temporal task model can incrementally be updated with new demonstrations simply by incrementing the absolute occurrence frequencies, i. e., the edge attributes of the corresponding edges. Most importantly, the updates of the temporal task model can also be weighted. This is a very important feature to better reflect user preferences, to adequately consider very representative demonstrations, negative examples, or even incorporate direct user commands from different modalities such as speech. For example, the command "*Never pour the milk before the cereals into the bowl!*" could be mapped straight-forward to the temporal task model simply by freezing the absolute occurrence frequency of the *before* relation in the edge attributes of (*pour milk*, *pour cereals*) at zero (adjusting the contrary edge accordingly). The robot could also ask questions about unclear or contradicting temporal relations similar as done in other works [12].

D. Inferring Subtasks from the Temporal Task Model

Each edge in the temporal task model is a superposition of all observed temporal relations from all demonstrations between the actions. The process of inferring subtasks from a temporal task model involves *fixing* one temporal relation between each pair of actions in the temporal task model or discarding it if no temporal relation is evident. Naively fixing the most likely temporal relation in each edge is generally not promising, since it could introduce inconsistencies. Hence, in order to infer subtasks from the temporal task model, we perform the following steps on a mutable copy of it.

1. Remove contradictory precedences: Edges in the temporal task model that primarily show contradicting precedence relations (*before* and *after*) in approx. the same amounts are removed (similar to [7], [8]).

2. Identify execution threads: These threads are defined as paths in the temporal task model, where the primarily observed relation is *meets* and the same object is manipulated. In real demonstrations, it is unlikely that both hands simultaneously start/stop with an action. However, the *meets* relation naturally occurs when one hand stops with one action and transitions into another. Exploiting this property allows implicitly reconstructing sequences of actions, which were executed by one hand using the same object. The execution threads serve as seeds to identify subtasks and are the first relation to be fixed.

3. Identify concurrency: From all edges which were not fixed yet, and given the sequential execution threads, we search for predominantly concurrent temporal relations. The following temporal relations are considered: *equals*, *starts* and *finishes*, *during* and *overlaps* in that order. Here, we only fix those relations which do not contradict any already fixed relations and discard contradicting ones. The chosen order ensures that relations requiring a tighter temporal coupling are preferred (e. g., *equals* over *starts*).

4. Identify subtasks: The constructed graph is divided into its weakly connected components, which are parts of a graph not connected by any edges. We interpret these components as subtask candidates, that potentially need to be split further (e. g., along a common action which occurred several times in demonstrations). To test if a split is necessary, we count the number of execution threads of each subtask candidate by dividing it again into its weakly connected components while only considering edges fixed to *meets*. If the number of resulting execution threads is 1 or 2, the subtask candidate is a unimanual or bimanual subtask, respectively. If the number is larger than 2, several subtasks collapsed at one common action and thus require to be split for bimanual execution along the common action. Note that multi-manual robots could exploit this property to execute bimanually demonstrated tasks multi-manually.

5. Identify subtask temporal relations: We identify potential temporal relations between subtasks by pairwisely considering the temporal relations between all actions of one subtask to all actions of another subtask. The result can either be a trivial temporal relation (*before* or *after*), or none at all.

Note that for each update of the temporal task model, the procedure described above needs to be repeated to identify the temporal intra- and inter-subtask relations. In Figure 2, this process is shown for a temporal task model built with data from a muesli preparation task.

V. Experiments and Evaluation

We present two quantitative evaluations of the approach on two new datasets, one synthetic dataset (Section V-A), and one from real data (Section V-B), as well as a qualitative evaluation, showcasing the usefulness of our approach on the humanoid robot ARMAR-6 (Section V-C). Both datasets are publicly available for download on our homepage[1], together with links to open source implementations.

[1] `https://bimanual-actions.humanoids.kit.edu`

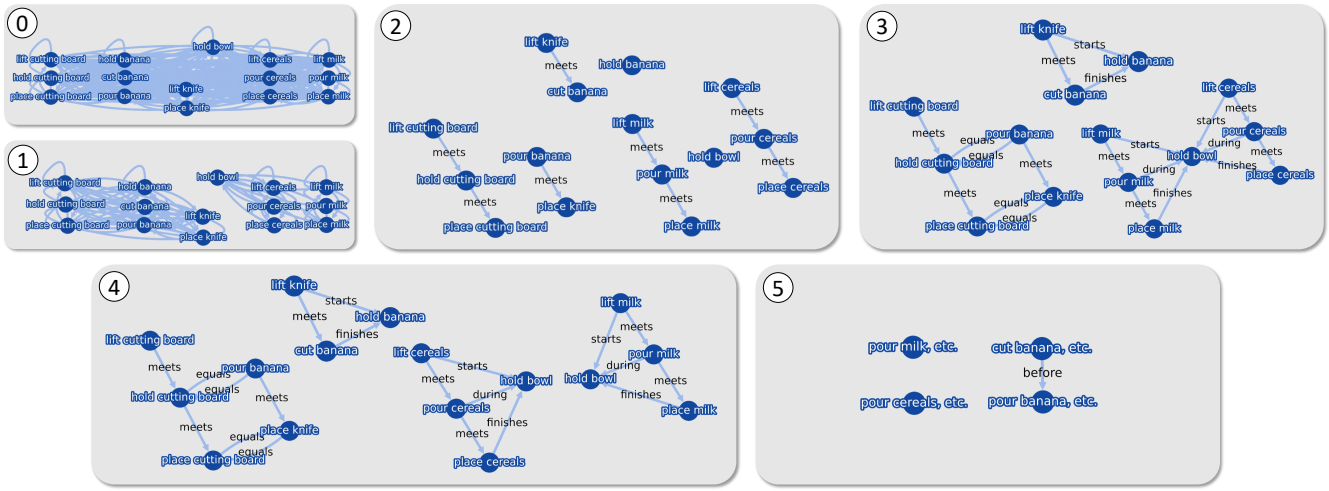


Figure 2. Depiction of the 5 steps to infer the temporal intra- and inter-subtask constraints from a temporal task model shown in ⓪. Top row: ① Removal of contradictory precedence relations. ② Identification of execution threads. ③ Identification of concurrency. Bottom row: ④ Identification of subtasks. Output: Intra-subtask temporal relations. ⑤ Identification of precedence relations between subtasks. Output: Inter-subtask temporal relations. Note that each of these gray areas outlines one graph, even though not all parts of them are connected.

A. *Synthetic Dataset*

Experimental Setup: For this evaluation, we define a hypothetical scenario with actions and their temporal relations. For this, a generator was implemented, which takes as input a set of actions, together with a set of temporal constraints between these actions, and outputs all possible demonstrations, which are valid given the set of temporal constraints. We perform a benchmark of the temporal task model using this synthetic data. New demonstrations are iteratively added to the temporal task model, after which subtasks and temporal constraints are identified. The proposed temporal constraints of our approach are compared against the ground truth temporal constraints used to generate the dataset. For each constraint identified by our approach or in the ground truth data, there can be 4 possible outcomes: (i) True positive: A temporal constraint is identified, which is in the ground truth data. (ii) False positive: A temporal constraint is hypothesized to be there, but it is not in the ground truth. (iii) False negative: A temporal constraint is in the ground truth, but it was not identified. (iv) True negative: No temporal constraints were identified between two actions, and there is no corresponding constraint in the ground truth. We report on the precision, as well as the recall scores. Note that false positives can and should occur by design because they can only be eliminated after several demonstrations. False negatives, on the other hand, should not occur. Thus, we aim for a recall as close to 1 as possible, and only secondarily for a high precision. The dataset consists of 216 demonstrations in an electric motor disassembly scenario, where the motor housing and the gearbox lid can be disassembled in any order. For this experiment, 100 learning scenarios were simulated, starting with one demonstration and consecutively adding a random demonstration, which was not seen before. All demonstrations were weighted equally.

Results: In Figure 3, the mean precision and recall with standard deviation over all 100 learning scenarios are plotted

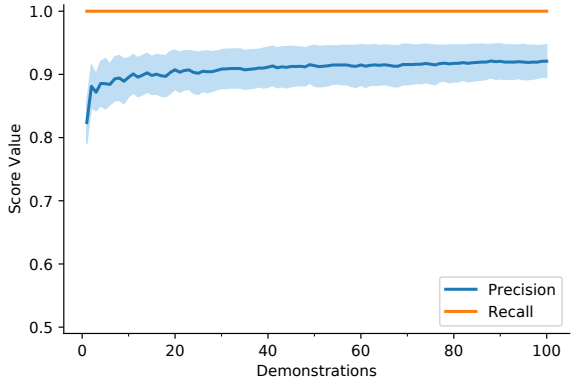


Figure 3. Mean precision and recall with standard deviation over 100 simulated learning scenarios, beginning with 1 demonstration, and with previously unseen demonstrations added consecutively.

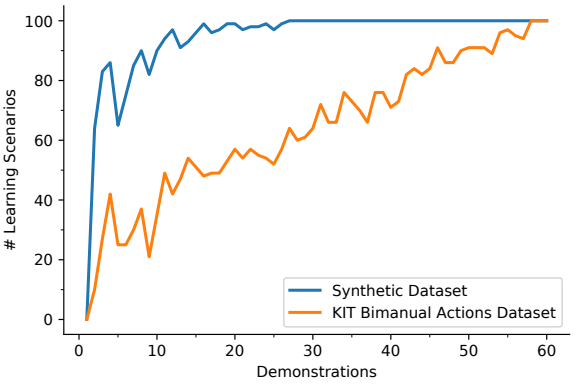


Figure 4. Number of learning scenarios over the number of demonstrations, which correctly identified ground truth temporal relations between subtasks for the synthetic dataset as well as for the KIT Bimanual Actions Dataset.

over the number of demonstrations considered for building the temporal task models. As can be seen, our approach shows straight from the beginning a saturated recall. As expected, the precision value increases with more demonstrations, as contradictions or different task variations lead to the elimination of earlier erroneously hypothesized constraints.

The false positives derived from the temporal task model mostly stem from *before* relations, which are not in the ground truth, for example *lift screwdriver before lift motor*. This becomes evident in Figure 4, where the number of learning scenarios is plotted over the number of demonstrations considered for building the temporal task model, which correctly identified that the *unscrew/place gearbox lid* subtask is temporally independent of the *unscrew/place motor housing* subtask. As can be seen, after only 10 demonstrations over 80 % of the learning scenarios are already able to identify this independence.

B. KIT Bimanual Actions Dataset

Experimental Setup: In this experiment, a subset of the KIT Bimanual Actions Dataset [20] was used, namely all recordings of the *prepare muesli* task. In total, 6 subjects demonstrated the task 10 times, resulting in 60 demonstrations with a total of 8 unique observed ways to achieve the goal. The subjects were asked to prepare muesli, given a banana, a pack of cereals, a bottle of milk, and a bowl. Additionally, a cutting board and a knife were on the table to cut the banana to put it as an ingredient into the muesli. In total, 4 subtasks could be observed, namely *cut banana, etc.*, *pour banana, etc.*, *pour cereals, etc.*, and *pour milk, etc.*. This is depicted in Figure 2 ⑤. Note that the subtasks are abbreviated after their effective actions (+ *etc.*) for simplicity, as the subtasks actually model a complex temporal arrangement of bimanual actions (cf. Figure 2 ④). In contrast to the synthetic dataset, though, the possible subtask sequences are not equally distributed, as can be seen in Table V. The data was re-labeled with ground truth object information and is available for download on our homepage. For this dataset, we do not have ground truth temporal relation data between actions of the demonstrated task, so an in-depth evaluation of all temporal relations is not feasible. Even more so, because these real demonstrations also include human errors, making it impossible to define a set of temporal constraints valid for all demonstrations. Similar to the evaluation on the synthetic dataset, though, we can define the desired outcome qualitatively, namely to identify that only the *cut banana* subtask temporally depends on *pour banana*.

TABLE V
DISTRIBUTION OF OBSERVED SUBTASK SEQUENCES IN THE DATASET.

Subtask Sequence (*etc.*-suffix omitted)	Number
pour cereals → *cut banana* → *pour banana* → *pour milk*	17
cut banana → *pour banana* → *pour cereals* → *pour milk*	15
cut banana → *pour banana* → *pour milk* → *pour cereals*	10
pour milk → *pour cereals* → *cut banana* → *pour banana*	8
pour cereals → *pour milk* → *cut banana* → *pour banana*	5
pour milk → *cut banana* → *pour banana* → *pour cereals*	3
cut banana → *pour cereals* → *pour banana* → *pour milk*	1
cut banana → *pour milk* → *pour cereals* → *pour banana*	1
Total	60

Results: Similar to the synthetic dataset, we also show in Figure 4 the number of learning scenarios, which correctly identified the true temporal relations between the subtasks over the number of demonstrations. It is evident, that the curve is not so steep in this case, mostly because of three reasons. First, the KIT Bimanual Actions Dataset features 4 subtasks, of which only 2 have a temporal dependence on each other, compared to the 2 subtasks from the synthetic dataset. Second, as already mentioned, the data is unbalanced, which can lead to wrong assumptions in terms of keeping constraints in the model, which later turn out to be false positives. Third, this dataset also features human errors, which leads to ground truth action segments very different from the others or even contradictory.

C. Robot Experiment

Experimental Setup: The approach was evaluated on the humanoid robot ARMAR-6 [21], which was tasked to clean up a table. Specifically, the task was to clean up the table by lifting and dropping a pack of rusk, a pack of cereal bars, and a bag of soy milk into a box as shown in Figure 5 ①.

Figure 5. ① Scene of the clean-up task involving 3 objects, namely a pack of cereal bars, a bag of soy milk, and a pack of rusk. The soy milk needs to be cleared before the rusk. The cereal bars can be cleared at any time. ② Scene given to ARMAR-6 with obstructed cereal bars pack.

In the scenario, the soy milk is on top of the rusk, hence the soy milk needs to be cleared before lifting the rusk. The robot received a predefined temporal task model built with 9 synthetically generated demonstrations, which all showed the same action sequence, resulting in 3 subtasks: (i) *lift cereal bars meets drop cereal bars*, (ii) *lift soy milk meets drop soy milk*, and (iii) *lift rusk meets drop rusk* with the subtask relations: (i) *lift/drop cereal bars before lift/drop soy milk*, and (ii) *lift/drop soy milk before lift/drop rusk*. Afterwards, the human demonstrated a different way to solve the task by first lifting and dropping the soy milk, which was weighted in this scenario to account for the valuable human demonstration data, leaving only one subtask relation: *lift/drop soy milk before lift/drop rusk*. To localize the objects we used the ArmarX [22] integration of SimTrack [23], while a rule-based action recognition system was used to extract action segments. After updating the temporal task model with the new demonstration, the robot was asked to execute the task by itself. For the execution, however, the cereal bars were placed in such a way that the view is obstructed by the soy milk (cf. Figure 5 ②), preventing the employed object pose estimation system from localizing the object and thus presenting the robot with the problem of a partially unobservable scene.

Results: During the reproduction, the robot would try to execute action candidates, which do not violate any temporal constraint until all actions are executed. In this concrete qualitative evaluation, the robot first tried to locate the cereal bars to lift them and drop them in the box. Since the view to the cereal bars was obstructed, the robot was not able to locate them, thus resulting in a rejection of the action candidate and

a consideration of the next candidate involving a different object. In accordance with the temporal task model, the robot instead lifted and dropped the soy milk, clearing the line of sight to the cereal bars. After successfully dropping the soy milk, the action candidate generator once again proposed to lift and drop the cereal bars, which could then successfully be located and executed. Finally, the rusk is lifted and dropped.

This evaluation shows, how the humanoid robot ARMAR-6 was able to utilize the constraints in the temporal task model to solve the clean-up task in a new way that was not seen in the demonstrations, born out of the necessity of not being able to locate a particular object. The initially strict constraints in the robot's temporal task model were relaxed with one demonstration, which was specially weighted to account for the importance of real demonstration data. Additionally, we showed how properties of our approach can be exploited to achieve the desired behavior with very limited human demonstration data (one real demonstration in this case). The weights used in this experiment were predefined, but can also stem from verbal commands or simply from the fact that real human demonstrations have to be weighted higher than other sources of information. Please also refer to our video attachment for this experiment.

VI. Conclusion and Future Work

We presented an approach to build a temporal task model from multiple human bimanual demonstrations that allows inferring subtasks of the demonstration as sets of actions with distinct temporal relations. To this end, we proposed a semantic and soft formulation of Allen's interval algebra to allow building temporal task models that represent the semantics of the task and extracting temporal constraints from real sensor data. We evaluated the approach quantitatively on two datasets and qualitatively on a humanoid robot. The knowledge about the temporal constraints between actions in a task is essential and can be used by a robot to achieve the task goal in new ways that are not seen in the demonstrations.

In future work, we will work on enriching the temporal task models with both temporal, as well as spatial/geometric information. For example, temporal keypoints in trajectories or motion primitives could be assessed to allow for a spatio-temporally coordinated bimanual execution of subtasks (e. g., similar to [16]). Additionally, we plan to use a more sophisticated representation of temporal relations in the future, by including quantitative information to reason about relations on a larger temporal horizon. In this work, we relied on simplifications such as a rule-based action recognition system and a known mapping of a symbolic action label to the execution. In the future, we plan to integrate a state-of-the-art action recognition system and learn bimanual actions as motion primitives from human demonstration.

Acknowledgment

We would like to thank André Meixner, Rainer Kartmann, Fabian Peller-Konrad, and in particular Fabian Reister, for the valuable discussions, helpful feedback, and technical considerations throughout all phases of this work.

References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., 2008, pp. 1371–1394.

[2] J. F. Allen, "Maintaining Knowledge About Temporal Intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[3] ——, "Towards a General Theory of Action and Time," *Artificial Intelligence*, vol. 23, no. 2, pp. 123–154, 1984.

[4] G. Ligozat and P. E. Santos, "Spatial Occlusion Within an Interval Algebra," in *AAAI Spring Symposium Series*, 2015.

[5] S. Schockaert, M. De Cock, and E. E. Kerre, "Fuzzifying Allen's Temporal Interval Relations," *Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 517–533, 2008.

[6] M. N. Nicolescu and M. J. Matarić, "Natural Methods for Robot Task Learning: Instructive Demonstrations, Generalization and Practice," in *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003, pp. 241–248.

[7] S. Ekvall and D. Kragić, "Learning Task Models from Multiple Human Demonstrations," in *International Symposium on Robot and Human Interactive Communication*, 2006, pp. 358–363.

[8] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zöllner, "Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments," *Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 322–332, 2007.

[9] T. M. Mitchell, "Generalization as Search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203–226, 1982.

[10] A. Kramberger, R. Piltaver, B. Nemec, M. Gams, and A. Ude, "Learning of Assembly Constraints by Demonstration and Active Exploration," *Industrial Robot: An Int. J.*, vol. 43, no. 5, pp. 524–534, 2016.

[11] C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu, "Robot Learning with a Spatial, Temporal, and Causal And-Or Graph," in *Int. Conf. on Robotics and Automation*, 2016, pp. 2144–2151.

[12] M. Racca and V. Kyrki, "Active Robot Learning for Temporal Task Models," in *Int. Conf. on Human-Robot Interaction*, 2018, pp. 123–131.

[13] F. Rossi, A. Sperduti, L. Khatib, P. Morris, and R. Morris, "Learning Preferences on Temporal Constraints: A Preliminary Report," in *International Symposium on Temporal Representation and Reasoning*, 2001, pp. 63–68.

[14] F. Rossi, A. Sperduti, K. B. Venable, L. Khatib, P. Morris, and R. Morris, "Learning and Solving Soft Temporal Constraints: An Experimental Study," in *Principles and Practice of Constraint Programming*, P. Van Hentenryck, Ed., 2002, pp. 249–264.

[15] P. P. Talukdar, D. Wijaya, and T. Mitchell, "Acquiring Temporal Constraints Between Relations," in *International Conference on Information and Knowledge Management*, 2012, p. 992.

[16] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots," *International Journal of Humanoid Robotics*, vol. 5, no. 02, pp. 183–202, 2008.

[17] X. Ye, Z. Lin, and Y. Yang, "Robot Learning of Manipulation Activities with Overall Planning Through Precedence Graph," *Robotics and Autonomous Systems*, vol. 116, pp. 126–135, 2019.

[18] E. Carpio, M. Clark-Turner, P. Gesel, and M. Begum, "Leveraging Temporal Reasoning for Policy Selection in Learning from Demonstration," in *Int. Conf. on Robotics and Automation*, 2019, pp. 7798–7804.

[19] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, A. Agostini, and R. Dillmann, "Object–Action Complexes: Grounded Abstractions of Sensory–Motor Processes," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 740–757, 2011.

[20] C. R. G. Dreher, M. Wächter, and T. Asfour, "Learning Object-Action Relations from Bimanual Human Demonstration Using Graph Networks," *Robotics and Automation Letters*, vol. 5, no. 1, pp. 187–194, 2020.

[21] T. Asfour, M. Wächter, L. Kaul, S. Rader, P. Weiner, S. Ottenhaus, R. Grimm, Y. Zhou, M. Grotz, and F. Paus, "ARMAR-6: A High-Performance Humanoid for Human-Robot Collaboration in Real-World Scenarios," *Robotics Automation Magazine*, vol. 26, no. 4, pp. 108–121, 2019.

[22] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The Robot Software Framework ArmarX," *it – Information Technology*, vol. 57, no. 2, pp. 99–111, 2015.

[23] K. Pauwels and D. Kragic, "SimTrack: A Simulation-Based Framework for Scalable Real-Time Object Pose Detection and Tracking," in *Intelligent Robots and Systems*, 2015, pp. 1300–1307.