



Learning Action Conditions for Automatic Behavior Tree Generation from Human Demonstrations

Lisa Scherf^{*}

Interactive AI & Cognitive Models for Human-AI Interaction (IKIDA), Technische Universität Darmstadt Darmstadt, Germany

Kevin Fröhlich^{*}

Interactive AI & Cognitive Models for Human-AI Interaction (IKIDA), Technische Universität Darmstadt Darmstadt, Germany

Dorothea Koert

Interactive AI & Cognitive Models for Human-AI Interaction (IKIDA), Technische Universität Darmstadt Darmstadt, Germany

ABSTRACT

The multitude of possible tasks and user preferences in real-world human-robot interaction scenarios renders pure pre-programming of robotic tasks inadequate. Recently, Behavior Trees (BTs) gained more focus as a modular internal task representation and particularly learning BTs directly from human video demonstrations offers also non-programming experts an opportunity to conveniently teach robots. However, automatically building BTs from human task demonstrations requires task constraints in form of action conditions. While existing work on automated BT generation often relies on pre-defined relevant features and heuristic condition computation, here we propose and evaluate different methods to automatically extract action pre- and post-conditions for BT generation from videos of human demonstrations. In particular, we first reduce the feature space using a correlation-based feature pre-selection, as well as a rule-based pre-selection based on a Decision Tree. Then, we select features that are relevant pre- and post-conditions for particular actions based on three different variance-based methods. We compare the different methods for feature selection and condition extraction on two pick-and-place tasks and discuss advantages and shortcomings of all methods in the context of learning BTs from human demonstrations.

CCS CONCEPTS

- Computing methodologies → Learning from demonstrations.

KEYWORDS

Learning from Demonstration, Behavior Trees, Interactive Task Learning

ACM Reference Format:

Lisa Scherf, Kevin Fröhlich, and Dorothea Koert. 2024. Learning Action Conditions for Automatic Behavior Tree Generation from Human Demonstrations. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion)*, March 11–14, 2024, Boulder, CO, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3610978.3640673>



This work is licensed under a Creative Commons Attribution International 4.0 License.

HRI '24 Companion, March 11–14, 2024, Boulder, CO, USA
 © 2024 Copyright held by the owner/author(s).
 ACM ISBN 979-8-4007-0323-2/24/03
<https://doi.org/10.1145/3610978.3640673>

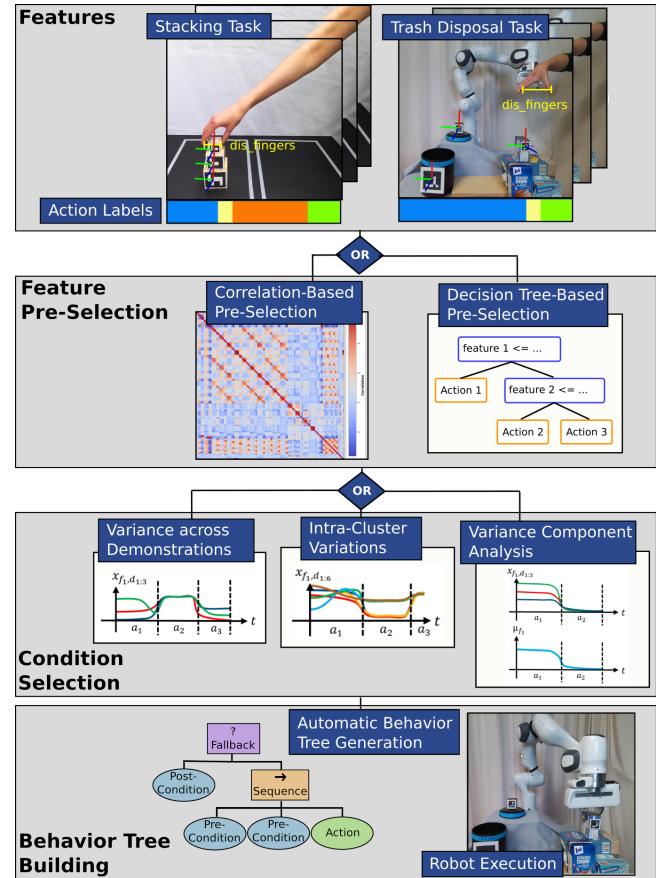


Figure 1: Overview of the proposed approaches including two feature pre-selection methods to reduce the feature space of human task demonstrations and three variance-based methods to select action conditions for Behavior Tree building.

1 INTRODUCTION AND RELATED WORK

In real-world human-robot interaction scenarios, Learning from Demonstration becomes an essential capability for future robots to expand their skill set and enable also non-expert users to teach robots new tasks [12]. Starting as an alternative to Finite State Machines in computer game programming [13], Behavior Trees (BTs) provide a modular, reactive, and interpretable task representation that has recently been more widely used for robotics [2, 8]. A Behavior Tree (BT) is a directed rooted tree that manages the hierarchical execution of behaviors (Figure 1). It serves as a control structure

for transitions between distinct tasks within an autonomous agent. Due to its tree structure, a BT allows a modular design of complex decision-making systems composed of several simple, reusable behaviors with built-in reactivity [2].

While BTs can be hand-coded or constructed manually using supporting interfaces [9], there are only a few approaches to learn BTs directly from demonstrations enabling non-expert programming [3, 5, 12–14]. These demonstrations can be provided in form of kinesthetic teaching [5], step-by-step action advice via a user interface [3, 13], or human task demonstrations [4, 14]. There are different approaches to automatically build a BT based on these demonstrations. One such approach is to first learn a Decision Tree (DT) from demonstrated state-action pairs and then convert this DT into an equivalent BT [3, 4, 13]. Gustavsson et al. [5] introduce a method that merges task constraints and context inference to optimize the BT creation through backchaining. Starting from the goal condition, actions are successively added based on their pre- and post-conditions. However, identifying relevant features and extracting action pre- and post-conditions is challenging, and action conditions are therefore often specified manually [5, 7]. To make the procedure usable for non-experts, it is desirable to determine the conditions automatically. Scherf et al. [14] automatically extract continuous action pre- and post-conditions from human task demonstrations to build a BT using backchaining. Nevertheless, they rely on pre-defining the set of task-relevant features instead of automatically selecting them based on the data. In addition, they use all features as pre-conditions, resulting in a high number of Behavior Tree nodes for more complex tasks.

In this paper, we propose two approaches to reduce a larger feature space to a task-relevant set of features and three variance-based approaches to identify relevant features as pre- and post-conditions for demonstrated actions for Behavior Tree building. The variance-based approaches are inspired by Abdo et al. [1], who uses variations between demonstration as an indication for a feature's relevance and Păis et al. [11], who propose additionally computing the variance of a feature within a demonstration assuming importance for the task of variables with substantial within-demonstration variance but little across-demonstration variance. We compare the different methods for feature selection and condition extraction on a stacking and a trash disposal task. From the experimental results we discuss advantages and shortcomings of the different methods for learning Behavior Trees from human demonstrations.

2 LEARNING BT ACTION CONDITIONS

Here, we propose different methods for automatic action condition extraction from human demonstrations that can be used to automatically build a Behavior Tree as described in [14]. Formally, let \mathcal{D} be a set of human task demonstrations. For each demonstration $d_i \in \mathcal{D}$ with $1 \leq i \leq N_d$ a set of features \mathcal{F} , e.g. hand poses and distances between objects is recorded. We define pre- and post-conditions as value ranges between minimum values $c_{pre}^-(a, f), c_{post}^-(a, f)$ and maximum values $c_{pre}^+(a, f), c_{post}^+(a, f)$ for a feature f and action a

$$\begin{aligned}\mathfrak{C}_{pre}(a) &\coloneqq \{[c_{pre}^-(a, f), c_{pre}^+(a, f)] | f \in \mathcal{F}_{pre}(a)\} \\ \mathfrak{C}_{post}(a) &\coloneqq \{[c_{post}^-(a, f), c_{post}^+(a, f)] | f \in \mathcal{F}_{post}(a)\},\end{aligned}\quad (1)$$

where $\mathcal{F}_{pre}(a)$ and $\mathcal{F}_{post}(a)$ are feature subsets of pre- and post-conditions for an action a . A condition is true if all features lie within the condition ranges. A pre-condition has to be true before action execution and is checked during action execution to allow reactivity of the Behavior Tree. Post-conditions specify which and to what range an action changes a particular feature. The goal of this paper is to identify task-relevant features and define $\mathcal{F}_{pre}(a)$ and $\mathcal{F}_{post}(a)$ for each demonstrated action a . In particular, we propose two approaches to pre-select potentially task-relevant features $\hat{\mathcal{F}} \subseteq \mathcal{F}$, as well as three variance-based approaches to select relevant pre- and post-conditions for each action. Here, the feature pre-selection is an optional intermediate step before the actual condition computation can help to reduce the feature space.

2.1 Feature Pre-Selection

In order to pre-select features from human demonstrations and reduce the size of a larger feature set, we discuss two methods.

Rule Extraction from Decisions Trees. Inspired by [3, 4, 13], we propose using Decision Trees (DTs) to pre-select features as potential candidates for action conditions. In particular, a DT is used to learn a mapping (Ψ) from the features Ψ of all human demonstrations to actions $c_\theta(\Psi) : \mathbb{R}^j \mapsto \mathcal{A}$, where θ denotes the model parameters. During training, the most informative features in distinguishing between different actions are considered to build a hierarchical tree. This tree contains successive decision rules that lead to one of the actions within the leaf nodes. The nodes higher up in the tree represent decisions, i.e. feature thresholds, with the highest discriminatory power. This DT can then be traversed to extract all features that are used to distinguish between all actions as $\hat{\mathcal{F}}$.

Correlation-based Feature Pre-Selection. The idea in correlation-based feature pre-selection is to group similar features and select representatives from each group to reduce the feature set while ensuring that no vital information is lost. For this approach and all following condition selection approaches, the feature time series of all demonstrations are first transformed to the same length using uniform scaling to stretch all feature sequences to the longest one. Missing values are linearly interpolated. The correlation between all transformed features is calculated, resulting in a correlation matrix with dimensions $N_f \times N_f$. Here, we use Kendall's Tau as a similarity measure between the feature time series. The correlation matrix is then used to perform hierarchical clustering with varying numbers of clusters. The final number of clusters is selected by the highest achieved silhouette score. Representatives are selected from each cluster by selecting the feature that has the lowest average intra-cluster distance, i.e. the lowest average distance to all other features within this cluster. All selected representatives form $\hat{\mathcal{F}}$.

2.2 Condition Selection

To decide whether a feature is a relevant pre- or post-condition for an action, metrics for the feature values at the beginning and end of an action are computed and compared to thresholds. Here, we define $p_{pre}(a) = [t_{a,start}, t_{a,start} + N_p]$ and $p_{post}(a) = [t_{a,end} - N_p, t_{a,end}]$ as the pre- and post-condition phases of an action where $t_{a,start}$ and $t_{a,end}$ marks the start and the end of the corresponding action a and N_p is a threshold for the number of values considered.

Variations across Demonstrations. The variations a feature across different demonstrations can provide insights into its importance w.r.t. the demonstrated task [1, 11]. We compute the variance across demonstrations $VAD_f(t)$ for a feature f at timestep t as

$$VAD_f(t) = \frac{\sum_{i=1}^{N_d} (\psi_t^{f,d_i} - (\mu_t^f)^2)}{N_d} \quad (2)$$

where μ_t^f corresponds to the mean value of the feature across all demonstrations and ψ_t^{f,d_i} represents the value of feature f in the i -th demonstration, both respective to a specific time step t .

A feature f is a pre-condition of action a , i.e. $f \in \mathcal{F}_{pre}(a)$ if

$$\mathcal{M}_{VAD} > VAD_f(t), \quad \forall t \in p_{pre}(a), \quad (3)$$

where \mathcal{M}_{VAD} is a threshold. Likewise, f is a post-condition of action a i.e. $f \in \mathcal{F}_{post}(a)$ if

$$\mathcal{M}_{VAD} > VAD_f(t), \quad \forall t \in p_{post}(a). \quad (4)$$

Intra-Cluster Variations. To cover situations where an action is shown in different ways, leading to a high variance in a feature between those demonstrations, it has been proposed to first cluster data points from demonstrations [1] and compute the average squared intra-cluster distance

$$SSD_{f(K)} = \frac{1}{KN} \sum_{c=1}^K \sum_{i=1}^{N_c} \text{dist}(v_i^c, \mu_c)^2 \quad (5)$$

with the number of data points N , the number of clusters K , and the number of data points assigned to a cluster N_c . Furthermore, μ_c corresponds to the mean and v_i^c to the value of the i -th data point of cluster c . We propose to use the K-Means clustering approach with an increasing number of clusters and a maximum of K of $\sqrt{N}/2$, which corresponds to a heuristic proposed by Mardia et al. [10]. Formally, feature f is a pre-condition of action a , i.e. $f \in \mathcal{F}_{pre}(a)$ if it holds that $\exists K \in [1, \sqrt{N}/2] : SSD_{f(K)} < \mathcal{M}_{SSD}$ for all N datapoints in $p_{pre}(a)$ and a threshold \mathcal{M}_{SSD} . The same criterion is applied to all datapoints in $p_{post}(a)$ to determine if a feature is a post-condition, i.e. $f \in \mathcal{F}_{post}(a)$.

Variance Component Analysis. While the previous approaches only consider variations across demonstrations, the variance within a demonstration can also provide insights into the relevance of a feature [11, 15]. Relevant features might show a high variance within a demonstration while at the same time showing a low variance across demonstrations. The variance within a demonstration for a feature f depending on the respective time step t can be calculated using a rolling window approach according to [15]

$$VWD_f(t) = \text{Var}\left(\mu_f\left(t - \frac{\omega}{2} : t + \frac{\omega}{2}\right)\right), \quad (6)$$

where ω is the window size. Feature f is a pre-condition of action a , i.e. $f \in \mathcal{F}_{pre}(a)$ if it holds that

$$\mathcal{M}_{VCA} < VWD_f(t) - VAD_f(t), \quad \forall t \in p_{pre}(a), \quad (7)$$

where \mathcal{M}_{VCA} is a threshold. Similarly, f is a post-condition of action a , i.e. $f \in \mathcal{F}_{post}(a)$ if

$$\mathcal{M}_{VCA} < VWD_f(t) - VAD_f(t), \quad \forall t \in p_{post}(a). \quad (8)$$

Table 1: Results Trash Disposal Task Condition Selection.

threshold	VAD		SSD		VCA	
	0.025	0.200	2.0	15.0	-0.005	-0.075
N_{cs}	0.71	0.92	0.72	0.95	0.81	0.94
N_{sbb}	1.0	1.0	1.0	1.0	1.0	0.95
N_{nbt}	146	276	120	284	180	262

Table 2: Results Stacking Task Condition Selection.

threshold	VAD		SSD		VCA	
	0.025	0.200	2.0	15.0	-0.005	-0.075
N_{cs}	0.91	0.98	0.91	0.98	0.84	1.0
N_{sbb}	0.8	0.9	0.5	1.0	0.6	0.4
N_{nbt}	511	895	456	843	530	763

The calculation of the minimum and maximum values for the final action conditions, as well as the BT building based on these pre- and post-conditions and a Backchaining approach are adopted from [14] for the experimental evaluation. The thresholds \mathcal{M}_{VAD} , \mathcal{M}_{SSD} , and \mathcal{M}_{VCA} are manually tuned using a heuristic grid search.

3 EXPERIMENTAL EVALUATION

We evaluate the different approaches for feature pre-selection and condition selection on human demonstrations of two tasks, namely a trash disposal task and a stacking task. For the trash disposal data set from [14], 19 participants demonstrated how to pick up trash and dump it in a trashcan. In addition, we recorded videos of 10 participants performing a stacking task where they had to stack three cubes on top of each other. Both data sets include object-object and hand-object distances as features and action labels such as *Move-to-Trash*, *Grasp-Trash*. Here, we use AR markers to track the position of relevant objects. In a real-world scenario, instead of markers a proper object tracking algorithm can be used. The feature sets include 67 features in total for the trash disposal task and 97 for the stacking task. For the trash disposal task, only three demonstrations were available per participant. Here, we used jittering [6] as a form of data augmentation to overcome the problem of low variation. For the stacking task, we recorded six demonstrations per participant and used these six demonstrations to evaluate all proposed methods.

In Tables 1 and 2, the results of all three condition selection approaches (VAD, SSD, VCA) without feature pre-selection on the trash disposal and stacking task features are shown. We report N_{cs} as the average percentage of conditions that match manually defined baseline action conditions necessary to define the task, i.e. the hand-cube distance is a relevant pre-condition for the Grasp-Cube action. Here, we identified 13 necessary action conditions for the trash disposal task and 44 conditions for the stacking task. N_{sbb} describes the percentage of successful BT buildings over all participants based on the extracted action conditions and N_{nbt} corresponds to the number of nodes in the resulting BT. For all approaches, we compare two exemplary thresholds, one restrictive threshold leading to a lower number of conditions and a more permissive threshold. The results show that in general, there is always a trade-off between a high N_{cs} and a low N_{nbt} . For the



Figure 2: The approaches are evaluated on demonstrations of a trash disposal task (A) and a stacking task (B).

Table 3: Trash Disposal Task, Combinations of Feature Pre-Selection and Condition Selection Approaches

	VAD+DT		VAD+C		SSD+DT		SSD+C		VCA+DT		VCA+C		ILBERT
threshold	0.025	0.200	0.025	0.200	2.0	15.0	2.0	15.0	-0.005	-0.075	-0.005	-0.075	multiple
N_{cs}	0.47	0.74	0.48	0.76	0.56	0.76	0.49	0.79	0.59	0.64	0.56	0.74	0.69
N_{sbb}	0.79	1.0	0.84	1.0	0.89	1.0	0.89	1.0	1.0	1.0	0.84	1.0	0.95
N_{nbt}	58	98	53	92	65	102	64	101	76	95	66	89	37

Table 4: Stacking Task, Combinations of Feature Pre-Selection and Condition Selection Approaches

	VAD+DT		VAD+C		SSD+DT		SSD+C		VCA+DT		VCA+C	
threshold	0.025	0.200	0.025	0.200	2.0	15.0	2.0	15.0	-0.005	-0.075	-0.005	-0.075
N_{cs}	0.82	0.91	0.75	0.84	0.82	0.89	0.75	0.84	0.75	0.91	0.66	0.84
N_{sbb}	0.8	0.9	0.8	0.9	0.8	1.0	0.8	1.0	0.5	1.0	0.6	0.8
N_{nbt}	290	473	232	365	283	443	223	347	306	419	233	315

trash disposal task, almost all methods are able to extract conditions leading to a successful BT building for all 19 participants. In comparison, the baseline condition computation from the ILBERT approach [14] leads to 18/19 successfully built BTs, N_{nbt} of 0.69 and a very low $N_{nbt} = 37$ using their pre-defined feature set of in total only seven features. For the more complex stacking task, only SSD with a threshold of 15 is able to extract conditions leading to a successful BT building for all 10 participants. However, in this case, N_{nbt} is very high, with 843 nodes. Even for the more restrictive thresholds, N_{nbt} is high for all approaches, resulting in large BTs.

Tables 3 and 4 show the results for the combination of the feature pre-selection based on Decision Trees (*DT*) or feature correlations (*C*) with the condition selection approaches. Both feature pre-selection methods previous to the condition selection were able to drastically reduce the number of nodes in the resulting BT while at the same time still being able to successfully build a BT for most participants. While the lowest N_{nbt} achieved without feature pre-selection was 120 and 456 for the two tasks (Table 1 and 2), the node number could be reduced to 53 and 223 BT nodes, respectively.

Overall, a lower N_{nbt} through the use of feature pre-selection methods can prevent BT failures due to irrelevant action conditions. However, there is a trade-off between a low N_{nbt} and a high N_{cs} , which can also result in failures during BT building or robot execution. While all condition selection approaches achieved good results in combination with a suitable feature pre-selection, VCA without pre-selection often lead to problems during BT building, resulting in a lower N_{sbb} . While some features might not change during an action resulting in a low within-demonstration variance, they can still be relevant for this action and successful task execution.

4 CONCLUSION AND FUTURE WORK

We proposed and evaluated different methods for automatic pre- and post-condition extraction in order to learn Behavior Trees from human demonstrations. In particular, the combination of feature pre-selection and condition selection demonstrated a significant reduction in tree size through the pre-selection methods. Our experimental evaluation on two tasks confirmed that the proposed methods are able to automatically select relevant features as action pre- and post-conditions, which are suitable to subsequently build a Behavior Tree as a robotic task representation.

While variance-based methods for condition computation showed promising first results, not all important task-relevant features might necessarily be discoverable through those. Therefore, in the future, we want to explore additional metrics. Moreover, additional experiments with more participants are necessary to evaluate the generalizability of the generated BTs and influence of a larger number of demonstrations. Also transfer of discovered relevant features across similar tasks, automated tuning of the thresholds in our condition-selection methods and incorporation of interactive user input in the feature discovery process and subsequent BT building are, in our opinion, interesting future research directions.

ACKNOWLEDGMENTS

This Work was funded by German Federal and Research (project IKIDA 01IS20045).

REFERENCES

- [1] Nichola Abdo, Henrik Kretzschmar, Luciano Spinello, and Cyrill Stachniss. 2013. Learning manipulation actions from a few demonstrations. In *2013*

- [1] *IEEE International Conference on Robotics and Automation*. 1268–1275. <https://doi.org/10.1109/ICRA.2013.6630734>
- [2] Michele Colledanchise and Petter Ögren. 2018. *Behavior Trees in Robotics and AI*. CRC Press. <https://doi.org/10.1201/9780429489105>
- [3] Kevin French, Shiyu Wu, Tianyang Pan, Zheming Zhou, and Odest Chadwicke Jenkins. 2019. Learning Behavior Trees From Demonstration. In *2019 International Conference on Robotics and Automation (ICRA)*. 7791–7797. <https://doi.org/10.1109/ICRA.2019.8794104>
- [4] Simona Gugliermo, Erik Schaffernicht, Christos Koniaris, and Federico Pecora. 2023. Learning Behavior Trees From Planning Experts Using Decision Tree and Logic Factorization. *IEEE Robotics and Automation Letters* 8, 6 (2023). <https://doi.org/10.1109/LRA.2023.3268598>
- [5] Oscar Gustavsson, Matteo Iovino, Jonathan Styrud, and Christian Smith. 2022. Combining Context Awareness and Planning to Learn Behavior Trees from Demonstration. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 1153–1160. <https://doi.org/10.48550/arXiv.2109.07133>
- [6] Guillermo Iglesias, Edgar Talavera, Angel Gonzalez-Prieto, Alberto Mozo, and Sandra Gomez-Canaval. 2023. Data Augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications* 35, 14 (2023), 10123–10145. <https://doi.org/10.1007/s00521-023-08459-3>
- [7] Matteo Iovino, Fethiye Irmak Doğan, Iolanda Leite, and Christian Smith. 2022. Interactive Disambiguation for Behavior Tree Execution. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. 82–89. <https://doi.org/10.1109/Humanoids53995.2022.10000088>
- [8] Matteo Iovino, Edvards Seukins, Jonathan Styrud, Petter Ögren, and Christian Smith. 2022. A survey of behavior trees in robotics and ai. *Robotics and Autonomous Systems* 154 (2022), 104096.
- [9] Matteo Iovino, Jonathan Styrud, Pietro Falco, and Christian Smith. 2023. A Framework for Learning Behavior Trees in Collaborative Robotic Applications. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. [https://doi.org/10.48550/arXiv.2303.11026 arXiv:2303.11026](https://doi.org/10.48550/arXiv.2303.11026)
- [10] Kantilal Varichand Mardia, John T Kent, and John M. Bibby. 1981. Multivariate analysis. (1981).
- [11] Lucia Pais, Keisuke Umezawa, Yoshihiko Nakamura, and Aude Billard. 2013. Learning robot skills through motion segmentation and constraints extraction. In *HRI Workshop on Collaborative Manipulation*. Citeseer, 5. <https://doi.org/10.1109/HUMANIDS.2012.6651507>
- [12] Harish Ravichandar, Athanasios Polydoros, Sonia Chernova, and Aude Billard. 2020. Recent Advances in Robot Learning from Demonstration. *Annual Review of Control, Robotics, and Autonomous Systems* 3 (05 2020), 297–330. <https://doi.org/10.1146/annurev-control-100819-063206>
- [13] Ismael Sagredo-Olivenza, Pedro Pablo Gómez-Martín, Marco Antonio Gómez-Martín, and Pedro Antonio González-Calero. 2019. Trained Behavior Trees: Programming by Demonstration to Support AI Game Designers. *IEEE Transactions on Games* 11, 1 (2019), 5–14. <https://doi.org/10.1109/TG.2017.2771831>
- [14] L. Scherf, A. Schmidt, S. Pal, and D. Koerl. 2023. Interactively learning behavior trees from imperfect human demonstrations. *Frontiers in Robotics and AI* 10 (2023). <https://doi.org/10.26083/tuprints-00024370>
- [15] Ana Lucia Pais Ureche, Keisuke Umezawa, Yoshihiko Nakamura, and Aude Billard. 2015. Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations. *IEEE Transactions on Robotics* 31, 6 (2015), 1458–1471. <https://doi.org/10.1109/TRO.2015.2495003>