

Data Augmentation for Instruction Following Policies via Trajectory Segmentation

Niklas Hoepner¹, Ilaria Tiddi², Herke van Hoof¹

¹University of Amsterdam

²Vrije Universiteit Amsterdam

n.r.hopner@uva.nl, i.tiddi@vu.nl, h.c.vanhoof@uva.nl

Abstract

The scalability of instructable agents in robotics or gaming is often hindered by limited data that pairs instructions with agent trajectories. However, large datasets of unannotated trajectories containing sequences of various agent behaviour (play trajectories) are often available. In a semi-supervised setup, we explore methods to extract labelled segments from play trajectories. The goal is to augment a small annotated dataset of instruction-trajectory pairs to improve the performance of an instruction-following policy trained downstream via imitation learning. Assuming little variation in segment length, recent video segmentation methods can effectively extract labelled segments. To address the constraint of segment length, we propose Play Segmentation (PS), a probabilistic model that finds maximum likely segmentations of extended subsegments, while only being trained on individual instruction segments. Our results in a game environment and a simulated robotic gripper setting underscore the importance of segmentation; randomly sampled segments diminish performance, while incorporating labelled segments from PS improves policy performance to the level of a policy trained on twice the amount of labelled data.

Code — <https://github.com/NikeHop/PlaySegmentation-AAAI2025>

Introduction

Advances in natural language conditioned generative models led to breakthrough results on text-to-image (Saharia et al. 2022), text-to-text (Ouyang et al. 2022) and text-to-audio generation (Le et al. 2023) forming the backbone of many recent AI applications (OpenAI 2023). A similarly capable model that learns to map natural language instructions to the corresponding trajectories of a game or robotic agent is an ongoing research challenge (Brohan et al. 2023). The main obstacle towards that goal is the lack of data, i.e. annotated trajectories containing diverse behaviours in different environments (Walke et al. 2023). To address this data limitation, recent work focuses on enhancing the sample efficiency of training algorithms (Yu and Mooney 2023), transforming foundation models into policies (Brohan et al. 2023) or performing data augmentation via simulation-based approaches (Wang et al. 2024). Another remedy for the lack

of data is to develop training algorithms that learn from data sources that, although abundant, deviate from the conventional framework of instruction-trajectory pairs (Ma et al. 2023; Wang et al. 2023b). One such source of less conventional data are large collections of videos of agent behaviour. For example, consider gameplay videos on YouTube (Baker et al. 2022) or play data in robotics (Lynch et al. 2019; Mees et al. 2022; Wang et al. 2023b), where long unsegmented trajectories are generated by humans teleoperating a robotic agent with the instruction to play. We will refer to trajectories that contain sequences of diverse agent behaviour as play trajectories. Our goal is to generate training data for an instruction following policy from them. This task requires segmenting the play trajectories and labelling the resulting segments with the corresponding instructions. We tackle this problem in a semi-supervised setting, in which a small amount of subsegments of the play trajectories are labelled with the corresponding instructions (see Figure 1).

Previous work on trajectory segmentation has focused on unsupervised methods, aiming to discover skills that maximise the likelihood of the unsegmented trajectories (Kipf et al. 2019; Rao et al. 2022; Jiang et al. 2022). However, these approaches do not guarantee that the identified skills align with any natural language instructions. Research on labelling trajectory segments with corresponding instructions have generally assumed that labelled and unlabelled segments are drawn from the same distribution (Xiao et al. 2023; Ge et al. 2023). Fewer studies have explored the segmentation of trajectories with multiple behaviours into segments corresponding to individual instructions (Shiarlis et al. 2018; Sharma, Torralba, and Andreas 2022). The annotations in those works assume entire trajectories are labelled with natural language plans which describe the sequence of instructions followed by the agent.

Extracting labelled segments from play trajectories poses the challenge of identifying segments that align with specific instructions. Adding non-representative or wrongly labelled segments, potentially harms the performance of a policy trained downstream. We adapt existing segmentation methods from the video segmentation literature and show that they struggle to generalise from the labelled dataset to the longer play trajectories. To address this, we introduce play segmentation (PS), a probabilistic model of segmentations of play trajectories, that can be trained from single instruc-

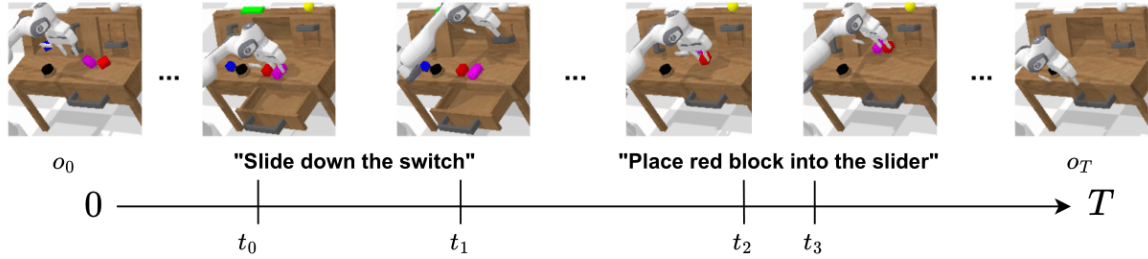


Figure 1: Example of play data, where the play trajectory contains sequences of instructable agent behaviour. The trajectory is represented by the observation sequence of the agent. Parts of the trajectory are labelled with the corresponding instructions and form the annotated dataset. Sampling random segments bears the risk to capture incomplete instructions or multiple ones.

tion segments. To summarise, our contributions are:

- Adapting video segmentation methods to play data and analysing their ability to improve policy performance via data augmentation.
- Developing Play Segmentation, a segmentation method capable of segmenting trajectories containing sequences of instructions while only being trained on individual instruction segments.
- Demonstrating that the extracted labelled segments of Play Segmentation lead to policy improvements across two environments.

Related Work

Trajectory Segmentation: A significant body of work has focused on learning reusable skills from a set of trajectories in an unsupervised manner (Niekum et al. 2012; Fox et al. 2017; Kipf et al. 2019; Jiang et al. 2022; Fu et al. 2024). The underlying approach shared by these methods is to model the distribution of trajectories using a latent variable model, where the latent variables represents the active skill at each timestep. After training the probabilistic model via maximum likelihood (Fox et al. 2017; Kipf et al. 2019), inferring the latents for a trajectory results in a segmentation of the trajectory (Jiang et al. 2022). However, there is no guarantee that the segments will correspond to specific instructions. To ensure that the extracted subsegments align with natural language instructions, a set of annotated examples can be employed (Shiarlis et al. 2018). Given the high cost of annotations, it is desirable for this set to be only a small fraction of the available demonstrations, which naturally leads to a semi-supervised setup (Sharma, Torralba, and Andreas 2022). Prior work looked at data setups in which trajectories are paired with plans describing the sequence of skills performed by the agent (Shiarlis et al. 2018; Sharma, Torralba, and Andreas 2022). Here, the annotation are segments of individual instructions within longer trajectories. Therefore we do not need to model the segmentation as a latent variable but can learn the parameters of a conditional distribution over segmentations given a trajectory.

Semi-Supervised Instruction Following: The standard approach to train instructable agents is to perform imitation

learning on a dataset of instruction-trajectory pairs (Stepputtis et al. 2020; Jang et al. 2022; Lynch et al. 2022). Due to the difficulty of generating natural language annotated trajectories, the problem is often studied in a semi-supervised learning setup (Xiao et al. 2023; Ge et al. 2023). For example, Xiao et al. leverage CLIP embeddings (Radford et al. 2021) to learn a labelling model on a small annotated dataset, which is used to label a large unannotated dataset. Training a policy on the joint dataset improves the task accuracy. However, the unlabelled and labelled trajectories come from the same distribution, making it unnecessary to identify segments corresponding to instructions in the unlabelled trajectories. The model that has been trained in a setting closest to the one proposed here is SL3 (Sharma, Torralba, and Andreas 2022). Given a small set of trajectories labelled with an overall goal and the sequence of instructions contained in it (Shridhar et al. 2020), SL3 applies an iterative procedure of segmenting trajectories, labelling segments and learning an instruction conditioned policy. The labelled dataset here consists of individual instructions and not plans, but unlike the case of SL3 does contain information about the start and end of these instructions.

Video Segmentation: The vision community has studied the task of identifying actions in uncropped videos under related but distinct setups. While Action Segmentation (AS) methods (Farha and Gall 2019; Yi, Wen, and Jiang 2021) try to predict for each frame the correct action class, Temporal Action Localization (TAL) methods try to predict the boundary timesteps of an action segment and a corresponding action label (Shi et al. 2023; Wang et al. 2023a). Some frameworks are developed to handle multiple of these tasks (Yan et al. 2023). Datasets for training contain uncropped videos containing multiple, not necessarily contiguous, action segments that are annotated on a frame level or with segment boundary information (Tang et al. 2019; Kuehne, Arslan, and Serre 2014). Both classes of methods have been studied under different levels of supervision (Kumar et al. 2022; Ding and Yao 2022; Lu and Elhamifar 2021). However our setting provides a unique challenge where the length of the training videos and the number of activities in the training videos differ from the videos at test time. To the best of our knowledge we are the first to investigate the usage of video

segmentation models for data augmentation in sequential decision making settings.

Methodology

The full pipeline of our approach starts with segmenting play trajectories, followed by labelling the extracted segments with instructions and lastly training a policy on the extracted labelled segments. Depending on the segmentation model, the segmentation and labelling can be done jointly. First, we describe the structure of play data and introduce notation, followed by a description how video segmentation models can be adapted to the play data setting. Finally, we introduce our play segmentation approach.

Play Data

Given is a small annotated dataset $D_{\text{ann}} = (\tau_k, d_k^{T_k})_{k=1}^N$ of instruction-trajectory pairs, next to a large unannotated dataset of only play trajectories $D_{\text{unann}} = (d_k^{T'_k})_{k=1}^M$, where $T_k \ll T'_k$. Here, $d^T = (o_0, a_0, \dots, o_T)$ denotes a trajectory of length $T + 1$ with o_t and a_t being the observation and action of the agent at timestep t and τ is an instruction in natural language from a finite set of possible instructions ζ . We denote the segmentation of a trajectory as $\alpha_{0:T-1} \in \{0, 1\}^T$, where $\alpha_i = 1$ means that a segment ended at timestep i . A labelled segmented trajectory is the triple $(d^T, \alpha_{0:T-1}, \tau_{1:K})$, such that $\sum_{t=0}^{T-1} \alpha_t = K$.

Segmentation Models

Here, we discuss different strategies to extract labelled segments from play trajectories. We start with the simplest way, i.e. labelling random segments, and then discuss how video segmentation methods can improve upon this approach by cropping enlarged random segments. We then present Play Segmentation, a conditional probabilistic model over segmentations given play trajectories, that can be trained having access only to individual instruction segments.

Random Segments: The simplest method to augment the labelled dataset involves randomly sampling windows from the play trajectory and labelling them using a labelling model trained on D_{ann} . The segment length is sampled uniformly between the minimum and maximum length of the segments in D_{ann} . It is unlikely that random segments begin and end precisely at the start and end of an instruction, leading to out-of-distribution challenges for the labelling model. An alternative is to sample larger segments that contain the start and end of a single instruction and develop methods that crop out the instruction from the larger segment. We adapt UnLoc (Yan et al. 2023), an Action Segmentation method, and TriDet (Shi et al. 2023), a Temporal Action Localisation method, to perform the cropping.

UnLoc (Yan et al. 2023): UnLoc is a CLIP-based framework designed to tackle multiple video understanding tasks such as TAL and AS. Here, we focus on the action segmentation variation of UnLoc. It learns to predict frame-wise action labels given a segment of image-based environment observations by learning the probabilistic model

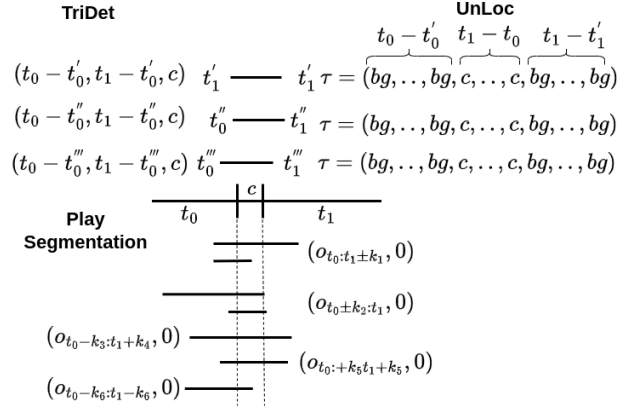


Figure 2: Overview of how training samples are generated from a single annotated segment for the different segmentation approaches. Here, c stands for the instruction class of the segment and bg for the background class.

$p_{\theta}(\tau_{0:T-1}|o_{0:T}) = \prod_{t=0}^{T-1} p_{\theta}(\tau_t|o_{0:T})$. The image observations and natural language instructions are encoded via CLIP. For each instruction the embedding is concatenated to the end of the observation sequence. Then a self-attention-based architecture outputs the logit of the instruction label for each frame. The usual training data (Shridhar et al. 2020) consists of pairs of segments and frame-wise annotations, containing multiple actions and possibly background segments. Since the longer play trajectories do not have frame-wise annotations, we approximate this setup by sampling windows around the annotated segments and labelling the additionally sampled frames as background (see Figure 2).

At test time the trained model can provide frame-wise labels for the whole play trajectory. However, during training the model only receives input segments containing one action-segment. We sample windows of the same length during training. The final labelled segment can then be obtained by cropping away the frames that are labelled as background. UnLoc does not scale with the number of instructions as each segment needs to be processed with each possible instruction such that the batch size is multiplied for each sample by the number of possible instructions $|\zeta|$.

TriDet (Shi et al. 2023): TAL methods are commonly trained on a dataset consisting of uncropped video segments $o_{0:T}$ containing multiple action segments, where each action segment is given by a triplet of the form (s_i, e_i, c_i) (Zhao et al. 2019). Here s_i represents the starting timestep, e_i the ending timestep and c_i the action class. The key design aspects of most architectures are pretrained features from an action classifier, followed by a feature pyramid to account for action segments of different length (Wang et al. 2023a). The TriDet architecture proposes a new boundary prediction head on top of the feature pyramid achieving state of the art performance on multiple TAL benchmarks. Similarly to UnLoc, we sample random windows around the segments of the annotated dataset and train TriDet to identify the action segment in the enlarged segment. At inference time, we sample random windows from the play trajectories and let

TriDet identify the action segment in the large random window. As TriDet directly predicts the boundary locations and action class, no further postprocessing is necessary before adding the labelled segments to the annotated dataset.

One drawback from the cropping approach is the assumption that there exist a window size for the random segment that captures with high likelihood the start and end of exactly one instruction. If the shortest and longest instruction length differ a lot, a large window size potentially captures multiple shorter instructions and a shorter window size will never capture a longer instruction. It is important to note that UnLoc and TriDet both are capable of segmenting full play trajectories, if fully segmented play trajectories were part of the training data.

Play Segmentation (PS): We introduce Play Segmentation (PS), a model tailored to segmenting play data. The desideratum for the model is that it can be trained on short segments from D_{ann} , but generalises to segmenting longer play trajectories. We start by factorising the probabilistic model over labelled segmentations $(\tau_{1:K}, \alpha_{0:T-1})$ as follows:

$$\begin{aligned} & p(\tau_{1:K}, \alpha_{0:T-1} | o_{0:T}) \\ &= p_{\theta_{\text{LM}}}(\tau_{1:K} | \alpha_{0:T-1}, o_{0:T}) \cdot p_{\theta_{\text{Seg}}}(\alpha_{0:T-1} | o_{0:T}) \\ &= \prod_{k=1}^K p_{\theta_{\text{LM}}}(\tau_k | o_{\alpha(k):\alpha(k+1)}) \cdot \prod_{t=0}^{T-1} p_{\theta_{\text{Seg}}}(\alpha_t | o_{\gamma(\alpha_{0:t}):t+1}), \end{aligned}$$

where α maps the i -th segment to the timestep it starts, $\alpha(K+1) = T$ and $\gamma(\alpha_{0:t}) = \max\{i | \alpha_i = 1, i \in \{0, \dots, t\}\}$. Next to a model that can predict the correct label for a segment ($p_{\theta_{\text{LM}}}$), we need to train a model to predict whether an observation sequence corresponds to an instruction segment ($p_{\theta_{\text{Seg}}}$). However the labelled dataset contains only segments that correspond to instructions. To train $p_{\theta_{\text{Seg}}}$, we augment D_{ann} with different types of negative samples.

For each segment of the annotated dataset $(\tau, o_{t_0:t_1})$ we translate the observation sequence $o_{t_0:t_1}$ to the left and right by a random number of timesteps sampled from $\{t_{\min}, \dots, t_{\max}\}$. The minimum and maximum steps for the translation are hyperparameters of the method and should be chosen based on the segment lengths present in the annotated dataset. Additionally we obtain negative samples by sampling a random timestep from $\{t_{\min}, \dots, t_{\max}\}$ used for elongating or shortening the groundtruth segment. For a single annotated segment $(o_{t_0:t_1}, \alpha = 1)$ we obtain the following negative segments:

- $(o_{t_0:t_1 \pm k_1}, \alpha = 0), (o_{t_0 - k_2:t_1}, \alpha = 0)$
- $(o_{t_0 - k_3:t_1 + k_4}, \alpha = 0),$
- $(o_{t_0 + k_5:t_1 + k_5}, \alpha = 0), (o_{t_0 - k_6:t_1 - k_6}, \alpha = 0).$

The model is trained by minimising the loss:

$$\begin{aligned} & L(\theta_{\text{LM}}, \theta_{\text{Seg}}) = \\ & \mathbb{E}_{s \sim D_{\text{ann}}} [\log(p_{\theta_{\text{LM}}}(\tau | o_{t_0:t_1})) + \log(p_{\theta_{\text{Seg}}}(1 | o_{t_0:t_1}))] \\ & + \mathbb{E}_{s' \sim D_{\text{ann}}^{\text{aug}}} [\log(p_{\theta_{\text{Seg}}}(0 | o_{t_0:t_1}))], \end{aligned}$$

Method	UnLoc	TriDet	PS
Comp. Complexity	$\mathbb{O}(\zeta)$	$\mathbb{O}(1)$	$\mathbb{O}(T^3)$

Table 1: Comparison of the computational (comp.) complexity of different segmentation models. It is measured by the number of neural functional evaluations (NFE) needed to segment one trajectory.

where $s = (o_{t_0:t_1}, 1, \tau)$ and $s' = (o_{t_0:t_1}, 0)$. The labelling model is only trained on the positive samples as the negative ones cannot be labelled with an instruction. Both models share parameters, but have separate prediction heads. At segmentation time we determine the most likely segmentation $\alpha_{0:T-1}^*$ using $p_{\theta_{\text{Seg}}}$:

$$\alpha_{0:T-1}^* = \arg \max_{\alpha_{0:T-1}} p_{\theta_{\text{Seg}}}(\alpha_{0:T-1} | o_{0:T}) \quad (1)$$

We can use dynamic programming (DP) to find $\alpha_{0:T-1}^*$ via the recursion:

$$\begin{aligned} & \max_{\alpha_{0:T-1}} \log p_{\theta_{\text{Seg}}}(\alpha_{0:T-1} | o_{0:T}) = \\ & \max_{i \in \{0, \dots, T-1\}} (\max_{\alpha_{0:i}} \log p_{\theta_{\text{Seg}}}(\alpha_{0:i} | o_{0:i+1}) + \\ & \log p_{\theta_{\text{Seg}}}(\alpha_{i+1:T-1} = (0, \dots, 1) | o_{i+1:T})). \end{aligned} \quad (2)$$

The DP algorithm has cubic complexity in the number of timesteps and is related to the Viterbi algorithm for Hidden Semi Markov Models (Yu 2010). Due to this complexity we cannot segment full play trajectories for large T but instead segment windows of size ω . The first window is sampled starting at $t = 0$, the beginning of the play trajectory. Let $o_{t_k:\omega}$ be the last segment resulting from the segmentation process. Then the starting point for the next window is either ω if $p_{\theta_{\text{Seg}}}(\alpha = 1 | o_{t_k:\omega}) > 0.5$, or t_k otherwise. In that manner complete play trajectories can be segmented. In Table 1 we give an overview of the computational complexities of the introduced segmentation models at segmentation time with respect to the trajectory length.

Experiments

The goal of the evaluation is to assess the capability of the different segmentation models to extract labelled segments from the play trajectories that can be used for data augmentation in an imitation learning context. In Section 4.1 the two evaluation environments are introduced and in Section 4.2 the importance of segmentation for successful data augmentation is highlighted. In Section 4.3 we compare the downstream-policy performance resulting from the different segmentation models and investigate reasons for the performance differences.

Environments

BabyAI (Chevalier-Boisvert et al. 2019) is a grid-based environment with a range of difficulty levels designed to test instruction following agents. Here we choose the GoTo-environment with 7 distractors (Figure 3). The agent needs

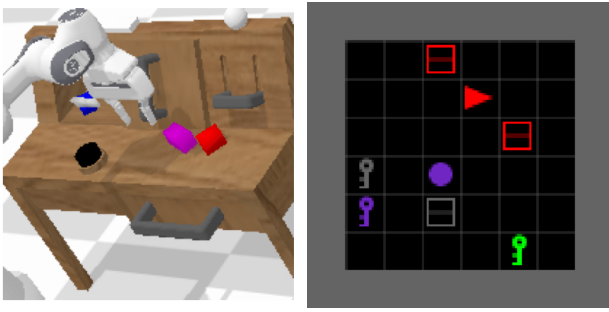


Figure 3: Example observations from the CALVIN environment (left) and the BabyAI environment (right).

to navigate to the object described in the instruction. If multiple objects of the same type are present going to any of them solves the task. The objects can be one of three types with six different colours. The discrete action space is 4-dimensional and the observations are 64x64 RGB images. Demonstrations can be generated via a bot that solves the task. To generate the play trajectories we sample a new goal object every time the agent solves its current task until it has solved a sequence of 10 different tasks. The length of the unsegmented trajectories ranges from 11 to 71 timesteps. The controlled generation of the play trajectories allows us to evaluate the quality of the extracted labelled segments, as we know which instruction is active at each timestep. One limitation of the dataset is the possibility that at a single timestep two instructions are active, i.e. the agents goal is a red key but it comes across a yellow ball along the optimal path. We evaluate policies by measuring the percentage of tasks solved within 25 timesteps over 512 episodes. Given a dataset of instruction-trajectory pairs we train a policy via imitation learning following the proposed implementation by Hui et al. (2020).

CALVIN (Mees et al. 2022) is a dataset containing play trajectories of a simulated 7-DOF Franka Emika Panda robot arm acting in a tabletop environment (Figure 3). Tasks include manipulating objects of different colours and shapes as well as opening/closing doors and switching lights on/off. There exist 34 different types of tasks. Individual segments of the play trajectories are labelled by human annotators with the corresponding instruction that is being executed. The length of the play trajectories varies between 1674 and 30838 timesteps, while the length of the instructions varies between 32 and 64 timesteps. The policy takes as input the image of a static camera showing the gripper arm and tabletop as well as an image of the gripper camera. An instruction following policy is evaluated on its capability of following sequences of natural language instructions. Each sequence consists of five instructions. The performance metric is the number of instructions completed by the policy until its first failure. The final evaluation score is the average number of instructions completed, computed over a 1000 sequences. Similar to prior work the policy is trained via multi-context imitation learning (MCIL) (Lynch and Sermanet 2021; Mees et al. 2022; Mees, Hermann, and Burgard 2022).

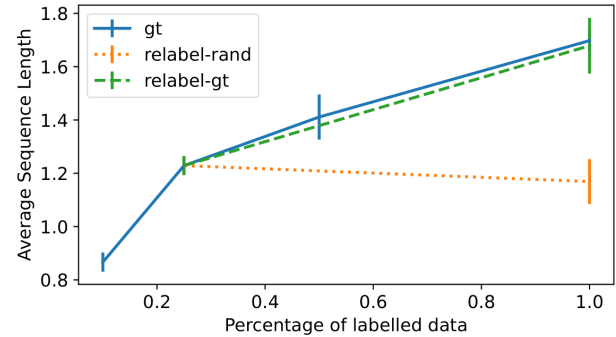
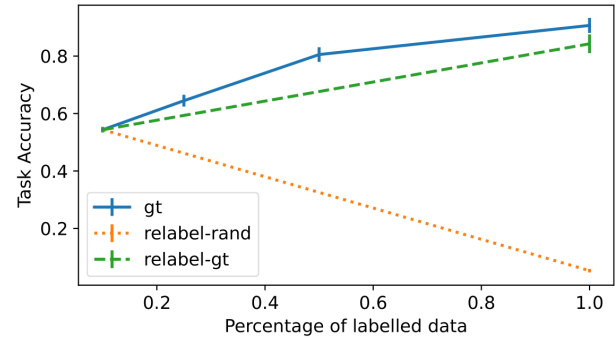


Figure 4: Effect of different amounts of annotated data on policy performance as well as the effect of data augmentation via labelled groundtruth segments and labelled random segment for BabyAI (top) and CALVIN (bottom).

Does Segmentation Matter?

To assess the policy’s performance with varying levels of annotated data, we subsample for both environments the annotated dataset into subsets of 10%, 25%, and 50%. Subsequently, the policy is trained for each subset. As anticipated, instruction completion decreases with less annotated data, as shown in Figure 4 (gt condition). For the next steps in the analysis, we will attempt to recover the policy’s performance at a 100% of the labelled data by starting with a subset of it and adding labelled segments extracted from the unsegmented trajectories of the unlabelled dataset. For the BabyAI environment we start with a subset consisting of 10% of the labelled dataset and for the CALVIN environment with a subset consisting of 25% of the labelled dataset. In the following we will refer to these subsets as the starting split. To understand whether it matters which segments are labelled and used for data augmentation we compare the performance between labelling and adding randomly sampled segments and groundtruth segments to the starting split. First, we train a labelling model based on the I3D architecture from the action recognition literature (Carreira and Zisserman 2017) on the starting split. It achieves a validation accuracy of $\sim 92\%$ BabyAI and $\sim 95\%$ on CALVIN. To add labelled groundtruth segments, we remove the labels from the left out data not in the starting split and add the

predictions of our labelling model as new labels.

In Figure 4, we can see that for both environments adding labelled groundtruth segments recovers the performance of a policy trained on all of the labelled data. One can conclude that the high accuracy transfers to the left out data only introducing a few wrongly labelled segments not having much impact on the policies performance. For the randomly sampled segments performance is either hampered (BabyAI) or stays the same (CALVIN). Adding randomly sampled segments has a larger negative effect in the BabyAI environment due to the greater variance in instruction length. This variance makes it less likely that a randomly sampled segment corresponds to a single completed instruction that can be correctly labelled by the labelling model.

Comparison of Segmentation Models

All segmentation models are trained on the starting split and then applied to extract labelled segments from the play trajectories. The starting split is augmented with labelled segments until it has the same number of instruction-trajectory pairs as the original labelled dataset. A policy is trained on top of the augmented dataset. We do not have any annotations for the play trajectories of the CALVIN dataset. This is a more realistic scenario but makes it harder to analyse the quality of the extracted labelled segments. Therefore, we start with the analysis in the BabyAI environment and then move to the CALVIN environment.

We measure the quality of the labelled segmentation via the accuracy of the assigned labels as well as the precision and recall of the segmentation points. It is important to note that these metrics are lower bounds for the actual performance. In the unsegmented trajectories of the BabyAI environment two instructions can overlap, i.e. while following instruction A the agent also completes instruction B. Therefore, the segmentation model can choose a correct segmentation point and label that is not part of the annotations. Video segmentation models have low precision and label accuracy (see Table 2), indicating that the chosen segments do not correspond to groundtruth segments. This indicates that video segmentation models cannot generalise their segmentation performance from the training segments to the random segments from play trajectories.

Play Segmentation achieves a higher precision and label accuracy than both video segmentation models. The recall of play segmentation is notably lower than the precision, indicating undersegmentation. This is also visible from the distribution of segment length of the extracted segments (Figure 5). While the segment obtained via play segmentation have a similar length distribution compared to the groundtruth, the number of segments with 1-3 timesteps is noticeably smaller. The performances of the policies trained downstream on the augmented datasets reflect the segmentation performances. While extracted labelled segments from TriDet have a negative impact on policy performance, data augmentation via Play Segmentation improves the performance to a level higher than if we had twice the amount of labelled data available (see Table 3). The extracted segments from UnLoc have little effect on the policies performance. This is a result of the short segment length of the extracted

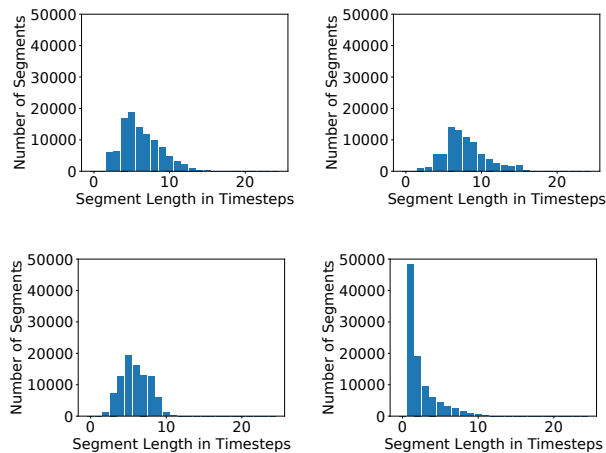


Figure 5: Distribution of segment length of the segments present in the groundtruth data (top left) and extracted via Play Segmentation (top right), TriDet (bottom left) and UnLoc (bottom right).

Method	Precision	Recall	F1	Label Acc.
PS	0.826	0.627	0.716	0.516
UnLoc	0.283	-	-	0.353
TriDet	0.338	-	-	0.323

Table 2: Quality of the extracted labelled segments. For UnLoc and TriDet we do not measure recall as they do not provide a complete segmentation of a trajectory but only the start and end of a single segment.

segments (Figure 5). During training of the policy a state of the trajectory is sampled jointly with the corresponding instruction to predict the next action. As a result longer segments make up a larger part of the training dataset for the policy.

Since the play trajectories in the CALVIN environment have no annotations we cannot investigate the quality of the segmentations. To augment the starting split we follow Xiao et al. (2023) and filter out labelled segments for which the labelling model has low confidence. The confidence threshold is set such that a label accuracy of 90% on the validation set is achieved. In Table 4 the results of the data augmentation are shown. Extracting labelled segments via video segmentation models improves policy performance. In comparison to the BabyAI environment a random segment from the play trajectories has a high likelihood of containing at most one instruction and additional frames from other instructions. Video segmentation models can then successfully crop away the undesirable frames as learned during training.

Play segmentation outperforms the best video segmentation model and achieves a performance higher than if twice the labelled data would have been available. However, there is still a large gap to the relabelling groundtruth segments. To investigate the cause for the performance differences we looked at the label distribution of the extracted segments.

Dataset	Task Accuracy (\pm Std.)
GT - 100%	0.907 \pm 0.026
GT - Relabel	0.843 \pm 0.032
GT - 50 %	0.805 \pm 0.025
PS	0.702 \pm 0.023
GT - 25%	0.645 \pm 0.020
UnLoc	0.583 \pm 0.020
GT - 10%	0.544 \pm 0.009
TriDet	0.261 \pm 0.134
Random - Relabel	0.053 \pm 0.006

Table 3: Policy performance in the BabyAI environment trained on the augmented datasets from the different segmentation models. The best segmentation model is highlighted. All results are averaged over 8 seeds.

Dataset	Avg. Seq. Length (\pm Std.)
GT - 100%	1.698 \pm 0.040
Relabel - GT	1.679 \pm 0.105
PS	1.477 \pm 0.092
GT - 50 %	1.411 \pm 0.085
Tridet	1.394 \pm 0.043
Unloc	1.315 \pm 0.035
GT - 25%	1.230 \pm 0.036
Relabel - Random	1.169 \pm 0.084
GT - 10%	0.867 \pm 0.036

Table 4: Performance of the MCIL policy in the CALVIN environment trained on the different augmented datasets. The best segmentation model is highlighted. All results are averaged over 4 seeds.

In the original dataset the labels are uniformly distributed. From Figure 6 one can see that certain task labels are over-represented in the added segments leading to an imbalanced dataset. There is a positive correlation between the amount of labelled segments added for a task and the policies improvement on the task (Figure 7). The failure to extract labelled segments for certain tasks is a possible explanation for the observed performance differences.

Limitations

Play Segmentation extracts better labelled segments than video segmentation models, which comes at the cost of higher computational complexity during segmentation. A potential remedy could be to sample from the distribution over segmentations. However, this requires the learned distribution to only assign high probability to reasonable segmentations. Another difficulty for the play data setting is to judge the quality of the augmented dataset if no groundtruth annotations are available. In more realistic settings, such as the CALVIN dataset, the play trajectories have no annotation to check the quality of the extracted segments.

Discussion

We demonstrate the potential to leverage unsegmented trajectories for data augmentation to enhance an instruction-

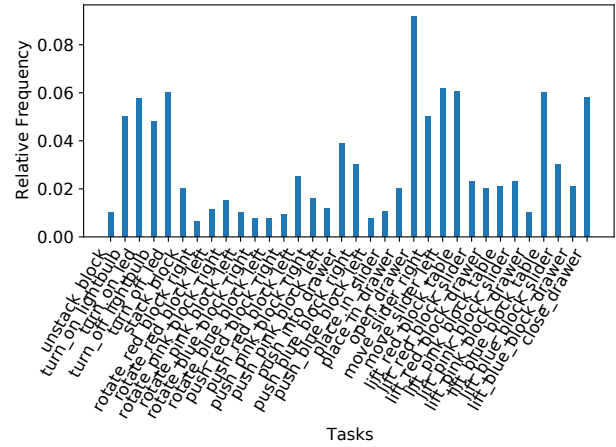


Figure 6: Distribution of labels in the segments extracted via Play Segmentation.

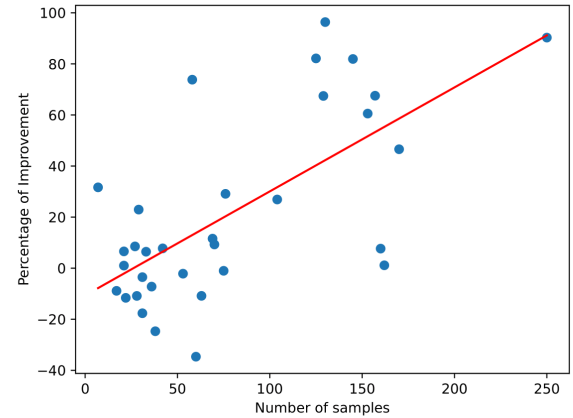


Figure 7: Displaying the relation between the number of samples added for a task and the improvement the policy achieves for that specific task in the CALVIN environment. The improvement is measure by the change in task accuracy between the original dataset and the augmented dataset relative to the improvement possible on that task.

following policy. Video segmentation methods enable the extraction of instruction segments via cropping. However, they are only applicable in scenarios, in which instructions are of similar length. Play Segmentation successfully segments trajectories being trained only on individual segments at the cost of higher computational complexity. The possibility for training segmentation models depends on the type of annotated data available. The setup studied here has a special type of annotation, i.e. labelled subsegments of play trajectories. The tradeoff between the cost of different types of annotations and the possibilities for training annotation models is an area for future research.

Ethical Statement

The long-term goal of the presented approach is to develop a text-to-trajectory model for game or robotic agents, similar to current text generation models. The two main risks are incorrect instruction execution and misuse by malicious actors. There is a trade-off between enhancing instruction-following policies and the potential for their misuse; better models can be more easily exploited. While the research aims to improve instruction following, positively impacting accuracy, it also heightens misuse risks.

Acknowledgments

This research was (partially) funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, <https://hybrid-intelligence-centre.nl>. This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-6630.

References

- Baker, B.; Akkaya, I.; Zhokhov, P.; Huizinga, J.; Tang, J.; Ecoffet, A.; Houghton, B.; Sampedro, R.; and Clune, J. 2022. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Chormanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; Florence, P.; Fu, C.; Arenas, M. G.; Gopalakrishnan, K.; Han, K.; Hausman, K.; Herzog, A.; Hsu, J.; Ichter, B.; Irpan, A.; Joshi, N. J.; Julian, R.; Kalashnikov, D.; Kuang, Y.; Leal, I.; Lee, L.; Lee, T. E.; Levine, S.; Lu, Y.; Michalewski, H.; Mordatch, I.; Pertsch, K.; Rao, K.; Reymann, K.; Ryoo, M. S.; Salazar, G.; Sanketi, P.; Sermanet, P.; Singh, J.; Singh, A.; Soricut, R.; Tran, H. T.; Vanhoucke, V.; Vuong, Q.; Wahid, A.; Welker, S.; Wohlhart, P.; Wu, J.; Xia, F.; Xiao, T.; Xu, P.; Xu, S.; Yu, T.; and Zitkovich, B. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *CoRR*, abs/2307.15818.
- Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *CoRR*, abs/1705.07750.
- Chevalier-Boisvert, M.; Bahdanau, D.; Lahlou, S.; Willems, L.; Saharia, C.; Nguyen, T. H.; and Bengio, Y. 2019. BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Ding, G.; and Yao, A. 2022. Leveraging Action Affinity and Continuity for Semi-supervised Temporal Action Segmentation. In *European Conference on Computer Vision*, 17–32. Springer.
- Farha, Y. A.; and Gall, J. 2019. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3575–3584.
- Fox, R.; Krishnan, S.; Stoica, I.; and Goldberg, K. 2017. Multi-Level Discovery of Deep Options. *CoRR*, abs/1703.08294.
- Fu, H.; Sharma, P.; Stengel-Eskin, E.; Konidakis, G.; Roux, N. L.; Côté, M.; and Yuan, X. 2024. Language-guided Skill Learning with Temporal Variational Inference. *CoRR*, abs/2402.16354.
- Ge, Y.; Macaluso, A.; Li, L. E.; Luo, P.; and Wang, X. 2023. Policy Adaptation from Foundation Model Feedback. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, 19059–19069. IEEE.
- Hui, D. Y.; Chevalier-Boisvert, M.; Bahdanau, D.; and Bengio, Y. 2020. BabyAI 1.1. *CoRR*, abs/2007.12770.
- Jang, E.; Irpan, A.; Khansari, M.; Kappler, D.; Ebert, F.; Lynch, C.; Levine, S.; and Finn, C. 2022. BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning. *CoRR*, abs/2202.02005.
- Jiang, Y.; Liu, E. Z.; Eysenbach, B.; Kolter, J. Z.; and Finn, C. 2022. Learning Options via Compression. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Kipf, T.; Li, Y.; Dai, H.; Zambaldi, V. F.; Sanchez-Gonzalez, A.; Grefenstette, E.; Kohli, P.; and Battaglia, P. W. 2019. CompILE: Compositional Imitation Learning and Execution. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 3418–3428. PMLR.
- Kuehne, H.; Arslan, A.; and Serre, T. 2014. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 780–787.
- Kumar, S.; Haresh, S.; Ahmed, A.; Konin, A.; Zia, M. Z.; and Tran, Q.-H. 2022. Unsupervised action segmentation by joint representation learning and online clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20174–20185.
- Le, M.; Vyas, A.; Shi, B.; Karrer, B.; Sari, L.; Moritz, R.; Williamson, M.; Manohar, V.; Adi, Y.; Mahadeokar, J.; and Hsu, W. 2023. Voicebox: Text-Guided Multilingual Universal Speech Generation at Scale. *CoRR*, abs/2306.15687.
- Lu, Z.; and Elhamifar, E. 2021. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8085–8095.
- Lynch, C.; Khansari, M.; Xiao, T.; Kumar, V.; Tompson, J.; Levine, S.; and Sermanet, P. 2019. Learning Latent Plans from Play. In Kaelbling, L. P.; Kragic, D.; and Sugiura, K., eds., *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, 1113–1132. PMLR.
- Lynch, C.; and Sermanet, P. 2021. Language Conditioned Imitation Learning Over Unstructured Data. In Shell, D. A.; Toussaint, M.; and Hsieh, M. A., eds., *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*.
- Lynch, C.; Wahid, A.; Tompson, J.; Ding, T.; Betker, J.; Baruch, R.; Armstrong, T.; and Florence, P. 2022. Interactive Language: Talking to Robots in Real Time. *CoRR*, abs/2210.06407.
- Ma, Y. J.; Sodhani, S.; Jayaraman, D.; Bastani, O.; Kumar, V.; and Zhang, A. 2023. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

- Mees, O.; Hermann, L.; and Burgard, W. 2022. What Matters in Language Conditioned Robotic Imitation Learning Over Unstructured Data. *IEEE Robotics Autom. Lett.*, 7(4): 11205–11212.
- Mees, O.; Hermann, L.; Rosete-Beas, E.; and Burgard, W. 2022. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334.
- Niekum, S.; Osentoski, S.; Konidaris, G. D.; and Barto, A. G. 2012. Learning and generalization of complex tasks from unstructured demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, 5239–5246. IEEE.
- OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.
- Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.
- Rao, D.; Sadeghi, F.; Hasenclever, L.; Wulfmeier, M.; Zambelli, M.; Vezzani, G.; Tirumala, D.; Aytar, Y.; Merel, J.; Heess, N.; and Hadsell, R. 2022. Learning transferable motor skills with hierarchical latent mixture policies. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, S. K. S.; Lopes, R. G.; Ayan, B. K.; Salimans, T.; Ho, J.; Fleet, D. J.; and Norouzi, M. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Sharma, P.; Torralba, A.; and Andreas, J. 2022. Skill Induction and Planning with Latent Language. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 1713–1726. Association for Computational Linguistics.
- Shi, D.; Zhong, Y.; Cao, Q.; Ma, L.; Li, J.; and Tao, D. 2023. TriDet: Temporal Action Detection with Relative Boundary Modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, 18857–18866. IEEE.
- Shiarlis, K.; Wulfmeier, M.; Salter, S.; Whiteson, S.; and Posner, I. 2018. TACO: Learning Task Decomposition via Temporal Alignment for Control. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 4661–4670. PMLR.
- Shridhar, M.; Thomason, J.; Gordon, D.; Bisk, Y.; Han, W.; Motlaghi, R.; Zettlemoyer, L.; and Fox, D. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 10737–10746. Computer Vision Foundation / IEEE.
- Stepputtis, S.; Campbell, J.; Phielipp, M. J.; Lee, S.; Baral, C.; and Amor, H. B. 2020. Language-Conditioned Imitation Learning for Robot Manipulation Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Tang, Y.; Ding, D.; Rao, Y.; Zheng, Y.; Zhang, D.; Zhao, L.; Lu, J.; and Zhou, J. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1207–1216.
- Walke, H.; Black, K.; Lee, A.; Kim, M. J.; Du, M.; Zheng, C.; Zhao, T. Z.; Hansen-Estruch, P.; Vuong, Q.; He, A.; Myers, V.; Fang, K.; Finn, C.; and Levine, S. 2023. BridgeData V2: A Dataset for Robot Learning at Scale. *CoRR*, abs/2308.12952.
- Wang, B.; Zhao, Y.; Yang, L.; Long, T.; and Li, X. 2023a. Temporal Action Localization in the Deep Learning Era: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Wang, C.; Fan, L.; Sun, J.; Zhang, R.; Fei-Fei, L.; Xu, D.; Zhu, Y.; and Anandkumar, A. 2023b. MimicPlay: Long-Horizon Imitation Learning by Watching Human Play. *CoRR*, abs/2302.12422.
- Wang, L.; Ling, Y.; Yuan, Z.; Shridhar, M.; Bao, C.; Qin, Y.; Wang, B.; Xu, H.; and Wang, X. 2024. GenSim: Generating Robotic Simulation Tasks via Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Xiao, T.; Chan, H.; Sermanet, P.; Wahid, A.; Brohan, A.; Hausman, K.; Levine, S.; and Tompson, J. 2023. Robotic Skill Acquisition via Instruction Augmentation with Vision-Language Models. In Bekris, K. E.; Hauser, K.; Herbert, S. L.; and Yu, J., eds., *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*.
- Yan, S.; Xiong, X.; Nagrani, A.; Arnab, A.; Wang, Z.; Ge, W.; Ross, D.; and Schmid, C. 2023. Unloc: A unified framework for video localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13623–13633.
- Yi, F.; Wen, H.; and Jiang, T. 2021. ASFormer: Transformer for Action Segmentation. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, 236. BMVA Press.
- Yu, A.; and Mooney, R. J. 2023. Using Both Demonstrations and Language Instructions to Efficiently Learn Robotic Tasks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Yu, S.-Z. 2010. Hidden semi-Markov models. *Artificial Intelligence*, 174(2): 215–243. Special Review Issue.
- Zhao, H.; Torralba, A.; Torresani, L.; and Yan, Z. 2019. HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 8667–8677. IEEE.