

ViLPAct: A Benchmark for Compositional Generalization on Multimodal Human Activities

Terry Yue Zhuo¹ and Yaqing Liao² and Yuecheng Lei²
Lizhen Qu^{1*} and Gerard de Melo³

Xiaojun Chang⁴ and Yazhou Ren² and Zenglin Xu^{5,6*}

¹Monash University ²University of Electronic Science and Technology of China

³HPI/University of Potsdam ⁴University of Technology Sydney

⁵Harbin Institute of Technology, Shenzhen ⁶Peng Cheng Lab

Abstract

We introduce ViLPAct, a novel vision-language benchmark for human activity planning. It is designed for a task where embodied AI agents can reason and forecast future actions of humans based on video clips about their initial activities and intents in text. The dataset consists of 2.9k videos from Charades extended with intents via crowdsourcing, a multi-choice question test set, and four strong baselines. One of the baselines implements a neurosymbolic approach based on a multi-modal knowledge base (MKB), while the other ones are deep generative models adapted from recent state-of-the-art (SOTA) methods. According to our extensive experiments, the key challenges are compositional generalization and effective use of information from both modalities¹.

1 Introduction

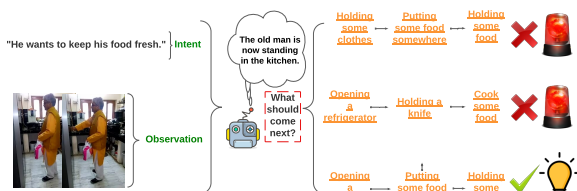


Figure 1: In daily life scenarios, an agent should be aware of future actions that will likely be taken by the user based on what it has observed. In this example, inputs of intent and observation are colored in green, while potential future action sequences are highlighted in orange. The first two sequences contain actions which do not align with the human intent. Thus, the agent needs to automatically detect which future actions are plausible by understanding the user’s intent.

One of the ultimate goals of Artificial Intelligence is to build intelligent agents capable of accurately understanding humans’ actions and intents,

*Corresponding authors: lizhen.qu@monash.edu, xuzenglin@hit.edu.cn

¹Our benchmark is available at <https://github.com/terryyz/ViLPAct>

so that they can better serve us (Kong and Fu, 2018). Newly emerging applications in robotics and multi-modal planning, such as Amazon Astro, have demonstrated a strong need to understand human behavior in multimodal environments. On the one hand, such an agent, e.g. an elderly care service bot, needs to understand human activities and anticipate human behaviors based on users’ intents. Here the intents may be estimated based on previous activities or articulated verbally by users. The anticipated behaviors may be used for risk assessment (e.g. falling of elderly people) and to facilitate collaboration with humans. On the other hand, recent advances in robotics show that it is possible to let robots learn new tasks directly from observed human behavior without robot demonstrations (Yu et al., 2018; Sharma et al., 2019). However, that line of work focuses on imitating observed human actions without anticipating future activities.

To promote research on action forecasting based on intents, we propose the *vision-language planning* task for human behaviors. As shown in Fig. 1, given an intent in textual form and a short video clip, an agent anticipates which actions a human is likely to take. We consider intents as given because there is already ample research on intent identification (Pandey and Aghav, 2020) and automatic speech recognition (Malik et al., 2021). To the best of our knowledge, there is no dataset to evaluate models for this task.

The task poses two major challenges. First, there are often multiple plausible action sequences satisfying an intent. Second, it is highly unlikely that a training dataset can cover all possible combinations of actions for a given intent. Hence, models need to acquire *compositional generalization* (Fodor and Pylyshyn, 1988), the capability to generalize to unseen action sequences composed of known actions.

In this work, we construct a dataset called ViLPAct for Vision-Language Planning of human Activities, which to the best of our knowl-

edge is the *first* dataset studying the above challenges. Specifically, we extend the Charades dataset (Sigurdsson et al., 2016) with intents via crowd-sourcing. As it is practically infeasible to find all possible future action sequences given an intent and a video clip of initial activities, we propose to evaluate all systems by letting each of them answer multi-choice comprehension questions (MQA) *without training them on those questions*. Given an intent and a video clip showing initial activities, each multi-choice question provides a fixed number of future action sequences as possible answers. A system is then asked to select the most plausible action sequence among them. We show that the rankings of all models using the MQAs correlate strongly with those obtained by asking human assessors to directly observe estimated action sequences. For training, we provide both a dataset for end-to-end training of sequence forecasting and a multimodal knowledge base (MKB) built from that dataset, which is also the *first* video-based multimodal knowledge base for human activities to the best of our knowledge.

We conduct the first empirical study to investigate compositional generalization for the target task. As baselines, we adapt three strong end-to-end deep generative models for this task and propose a neurosymbolic planning baseline using the MKB. The model is neurosymbolic because it combines both deep neural networks and symbolic reasoning (Garcez and Lamb, 2020). Given a video of initial activities and an intent, the deep models generate the top- k relevant action sequences, while the neurosymbolic planning model sends the intent and the action sequence recognized from the video as the query to the MKB, followed by retrieving the top- k relevant action sequences. Each model selects the most plausible answers by performing probabilistic reasoning over the relevant action sequences. We conduct extensive experiments and obtain the following key experimental results:

- We compare the evaluation results using MQA with the ones of human evaluation. The results of both methods are well aligned. Thus, MQA is reliable without requiring human effort.
- The likelihood functions of the deep generative models are not able to reliably infer which answers are plausible. In contrast, probabilistic reasoning is an effective method to improve compositional generalization.
- Despite information from both modalities be-

ing useful and complementary, all baselines heavily rely on intents in textual form but fail to effectively exploit visual information from video clips.

2 Related Work

Vision-Language Planning Task Vision Language Navigation (VLN) was among the first widely used goal-oriented vision-language tasks, requiring AI agents to navigate in an environment without interaction by reasoning on the given instruction (Anderson et al., 2018; Hermann et al., 2020; Misra et al., 2018; Jain et al., 2019). Recently, further goal-oriented vision-language tasks have been proposed. The Vision and Dialogue History Navigation (VDHN) task (De Vries et al., 2018; Nguyen and Daumé III, 2019; Thomason et al., 2020), which is similar to VLN, requires agents to reason on the instructions over multiple time steps. Other tasks such as Embodied Question Answering (EQA; Das et al. 2018; Wijmans et al. 2019), Embodied Object Referral (EOR; Qi et al. 2020b; Chen et al. 2019) and Embodied Goal-directed Manipulation (EGM; Shridhar et al. 2020; Kim et al. 2020; Suhr et al. 2019) rely on reasoning and interpreting the instruction with observation or object interaction in the environment. However, we argue that there are other ways to learn to plan without practising. Our task is one example of this, requiring agents to reason over the observation without performing actions.

Vision-Language Planning Datasets As existing vision-language planning datasets emphasize teaching embodied AI to perform the task like humans, they are constructed with interactive AI in mind. VLN (Anderson et al., 2018) datasets initially started exploring planning tasks with the textual instruction as a step-by-step abstract guide and minimal interaction with the environment. Extending the VLN task, VDHN (De Vries et al., 2018) datasets provide an interactive textual dialogue between the speaker and the receiver in multiple steps. The EQA (Das et al., 2018) task takes this a step further by providing data in an object-centric QA manner, advancing systems to understand the given environment through object retrieval. The EOR (Qi et al., 2020b) task designs object-centric datasets with detailed instructions, aiming at localizing the relevant objects accurately. The closest benchmark to ours is ALFRED (Shridhar et al., 2021) from the EGM task, which lets embodied agents decide

on actions and objects to be manipulated based on detailed instructions. However, in our setting, we ask intelligent systems to predict the most reasonable future action sequence based on human intents and answers in a Multiple Choice Question Answering (MQA) format. During prediction, we still give systems the flexibility to consider various combinations of actions and objects.

Vision-Language Planning Modeling According to Francis et al. (2021), several approaches have been used for planning. Greedy search in end-to-end models has been reported in several studies to work well in goal-oriented tasks (Fried et al., 2018; Das et al., 2018; Shridhar et al., 2020; Anderson et al., 2018). Task progress monitoring (Ma et al., 2019) is another method to tackle the planning. It allows models to backtrack on actions if the current action is found to be suboptimal. Mapping (Anderson et al., 2019) has as well been proposed for efficient planning via sensors. Topological and Exploration planning (Deng et al., 2020; Ke et al., 2019) enables modeling the planning in a symbolic manner. When goals are provided as several sub-goals, a divide and conquer strategy (Misra et al., 2018; Shridhar et al., 2020; Suhr et al., 2019) may be invoked to perform sub-task planning. In our work, we highlight another potential approach, knowledge base retrieval. As we construct an MKB containing various action sequences with detailed features, intelligent agents can retrieve the most suitable sequence from the MKB source in order to perform the planning.

3 Dataset Construction

We adopt videos from Charades (Sigurdsson et al., 2016) and solicit intents for videos via crowdsourcing. We consider videos that have action sequences of sufficient length appearing in both initial video clips and answers, which result in a dataset comprising 2,912 videos. The dataset is split into training/validation/test sets with a ratio of 70%, 10%, 20%. On the training dataset, we build an MKB by incorporating structural and conceptual information. On the test dataset, we collect a set of MQAs for model evaluation. The evaluation with MQAs is in fact an adversarial testing method, widely used for quality estimation in machine translation (Kanojia et al., 2021). Herein, the ability of a model to discriminate between correct outputs and meaning-changing perturbations is predictive of its *overall* performance, not just its robustness.

Thus MQAs are applied only for testing.

3.1 Data Normalization and Filtering

Charades is a large-scale video dataset of daily indoors activities collected via Amazon Mechanical Turk² (AMT). The average length of videos is approximately 30 seconds. It involves interactions with 46 object classes and contains 157 action classes, which are also referred to as **actions** for short. Each action is represented as a verb phrase, such as “pouring into a cup”. This dataset is chosen because *i*) it contains a sufficient number of long action sequences of human daily activities; *ii*) the intents are easily identifiable, as the activities in the videos are based on scripts; *iii*) there are rich annotations of videos that can be leveraged for dataset construction. The details of action sequence selection in videos are presented in Appendix 7.1, with the goal of choosing core action sequences having clear human goals.

In order to assess the quality of extracted action sequences, we randomly sample 100 videos from the test set for manual inspection. The primary action sequence of each video is evaluated in terms of three criteria: *i*) if all actions of a sequence occur in the video; *ii*) if the actions of a sequence appear in the same order as in the video; *iii*) if a sequence has any actions missing between the first and the last action. In total, we determined that 94 videos have all actions of their action sequences covered in the video. The actions of 92 videos appear in the same order as in the videos. Furthermore, 85 videos have no actions missing between the first and the last action of their sequences. Thus, the quality of such action sequences is adequate for VL planning evaluation.

Following prior work (Ng and Fernando, 2020), we consider the first 20% of a video as its initial visual state and aim to forecast future actions appearing in the remaining part of the video for a given intent. To have at least one future action per video, we retain only videos that contain at least one action sequence comprising more than three actions. As a result, we obtain 2,912 such videos, each of which is associated with one action sequence of length longer than three.

²<https://www.mturk.com>

3.2 Intent Annotation

An *intent* may be defined as “something that you want and plan to do”.³ Philosophers distinguish between future-directed intents and present-directed ones (Cohen and Levesque, 1990). The former guide the planning of actions, while the latter causally produce behavior. As the focus of this work is anticipating and planning actions, we encourage crowd-workers to also provide future-directed intents.

We recruit crowd-workers to annotate videos with future-directed and present-directed intents. Each annotator is provided with a full video clip and the associated action sequence. They are instructed to answer the question *what the person wants to do by taking the actions in the video*. Every annotator is asked to submit two intents. One of them should describe which activity the person intends to take, such as “drink a glass of water”. The other one needs to be at a high-level, such as “quench the thirst” or “be thirsty”. The permitted formats are either “S/He wants to + *do_something*” or “S/He is + *feeling*”. Thus, the annotators are encouraged to provide future-directed intents by differentiating them from ones causally leading to behaviours. To ensure the quality of intent annotations, we randomly assign three crowd-workers to write intents per video. The process of constructing the dataset for intent annotation involved a rigorous validation and selection process. One of the authors acted as an expert annotator, and conducted a thorough review of all crowd-sourced intents to identify and select the most reasonable annotations as the final results. The validation process was completed in three rounds, yielding increasingly higher percentages of reasonable annotations, with 82%, 94% and 100% respectively for each round. The annotations that did not meet the required criteria were discarded and not included in the final dataset. This rigorous validation process ensured that the final dataset is comprised of high-quality and relevant annotations, providing a robust foundation for subsequent modeling and analysis.

3.3 Multimodal Knowledge Base

We construct the MKB of human activities based on the **training set** and **validation set** by taking a neurosymbolic approach. The main challenges herein are twofold: i) how to represent multimodal

information from videos, action names, and intents adequately to facilitate information retrieval; ii) how to model shared knowledge of multimodal information. For the former, we allow both string and embedding based retrieval methods by attaching neural representations of video clips and texts to symbols of actions and action sequences. For the latter, we employ the classical planning language STRIPS (Bylander, 1994) and neural prototypes to encode abstract properties of actions.

At the core of the MKB is a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set \mathcal{V} comprises four types of nodes: action classes, action video clips, action sequences, and action sequence videos, while the edge set \mathcal{E} contains edges reflecting relationships between nodes.

An *action class* a^c is the abstraction of an action described in the language of STRIPS. The attributes of an action class include its ID, its name τ , its precondition set PRE, its add effect set ADD, and its delete effect set DEL. An action is executed only if its preconditions are satisfied. The effect sets ADD and DEL of an action class describe the add and delete operations applied to the current state after executing the action. For example, the precondition of *Closing a refrigerator* is $isOpen(refrigerator)$, $ADD = isClosed(refrigerator)$ and $DEL = isOpen(refrigerator)$. In this way, the properties described in STRIPS present the shared knowledge of each action class.

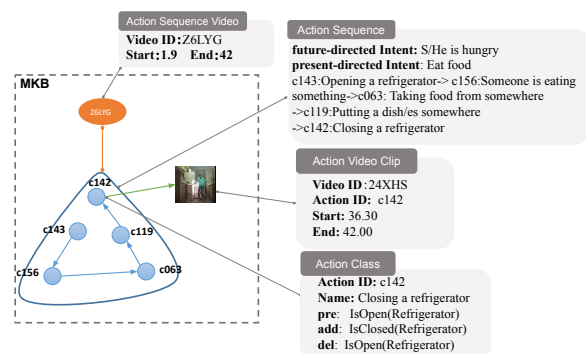


Figure 2: An example action sequence in the MKB.

An *action sequence* comprises a future-directed intent, a present-directed intent, and a sequence of action IDs. An intent is represented by both a word sequence and the distributed representation of the word sequence. We obtain the distributed representation of an intent by applying BERT (Devlin et al., 2018) and utilizing the representation of the CLS token. The collection of action sequences can

³Cambridge Dictionary, <https://dictionary.cambridge.org/>

be easily turned into a training set for end-to-end models by associating them with the corresponding video files.

The MKB includes two types of visual nodes: *action sequence videos* and *action video clips*. Each action sequence video is linked to the corresponding action sequence. For each action in an action sequence, we associate it with the corresponding video clip, as illustrated in Fig. 2. For each action video clip, we apply I3D to encode it into a sequence of frame-level visual feature vectors $\{\mathbf{f}_{s_1}, \mathbf{f}_{s_2}, \dots, \mathbf{f}_{s_t}\}$, where each vector $\mathbf{f}_{s_i} \in \mathbb{R}^{1024}$ corresponds to the features of an 8-frames snippet. To represent an action sequence video, we apply average pooling to the distributed representations of all involved video clips.

Relations. We consider two types of relations in the MKB. The first type of relation links an action sequence to the corresponding visual representation. The other type of relation associates an action in an action sequence with the corresponding action class. Therefore, it is easy to perform symbolic reasoning by using the STRIPS properties of each action class involved in an action sequence.

Statistics of MKB Table 1 provides statistics of the MKB. As we can observe, the MKB contains 2,402 action sequence videos and 12,118 action video clips. Each action sequence video is associated with one corresponding action sequence. There are 157 action classes in total and 1,969 unique action sequences. The average length of action sequences is 5.04.

Item	Statistics
# of action classes	157
# of action sequence videos	2,402
# of action video clips	12,118
# of action sequences (distinct seq)	2,402 (1,969)
# of action state templates	32
# avg. # of action sequence length	5.04

Table 1: Statistics of the MKB / training + validation set

3.4 Multi-Choice Comprehension Questions for Evaluation

Given the first 20% of a video as the initial state \mathbf{s} and a future-directed intent \mathbf{g} in text, the planning evaluation task involves choosing the most plausible future action sequence \mathbf{a}^f among six available choices. We determine the initial action sequence \mathbf{a}^i by checking if an action of a sequence starts before the end time of the initial state. To build such a dataset, we extended the test set with adversarially

generated incorrect answers. As the automatic approach may generate reasonable action sequences, we recruit another group of students to manually check all answers and determine the most plausible ones as the correct answers on AMT. Figure 3 shows an example of our planning task.

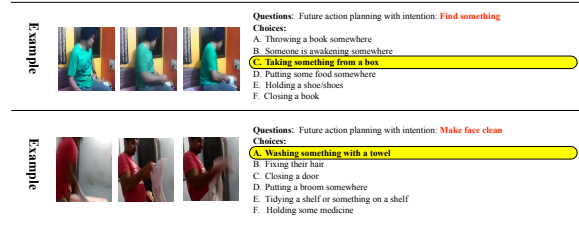


Figure 3: Two examples of ViLPAct MQA task

Generation of Incorrect Answers. We adapt the *Adversarial Matching* (AM) algorithm (Zellers et al., 2019) to turn the action sequence generation task into a multi-choice test. The key idea here is to substitute an action of an observed action sequence for an alternative action that is relevant to the preceding actions and is not overly similar to the action to be replaced. As many videos in the test set have only a single future action, the AM algorithm is extended to optionally insert a future action to generate an answer candidate.

More specifically, given the initial state, the action sequence, and the intent $(\mathbf{s}, \mathbf{a}, \mathbf{g})$ of a video, where $\mathbf{a} = (\mathbf{a}^i, \mathbf{a}^f)$, the algorithm starts by randomly deciding if it applies substitution or insertion to generate an answer candidate. If insertion is chosen, it inserts an action randomly selected among the 157 candidate actions, at a position that is randomly picked after the last action in \mathbf{a}^i . If instead substitution is chosen, we feed the initial action sequence \mathbf{a}^i to BERT and use the representation of the CLS token as the representation of \mathbf{a}^i . Then we apply BERT to turn each action into a vector by using the corresponding CLS representation. We randomly pick a future action a_i in \mathbf{a}^f and compute the score of a candidate action a_j as

$$s(a_j) = \log(P_{\text{sim}}(\mathbf{a}^i, a_j)) + \lambda \log(1 - P_{\text{sim}}(a_i, a_j)), \quad (1)$$

where $P_{\text{sim}}()$ is defined as cosine similarity. We set $\lambda = 0.7$ to find an optimal tradeoff between the obfuscation level of an incorrect answer and the probability of being a reasonable answer. We repeat this process until we have generated five answer candidates. For each set of generated answer

Statistics	Value
# of videos	510
avg. # of observed actions	2.79
avg. # of future actions	2.40
avg. # of actions	5.19
# of full action seq occurring in the training set	121
avg. # of distinct future action sequences for an intent	2.16
std. dev. of # of distinct future action sequences for all intents	3.69

Table 2: Basic statistics of MQA task / test set

candidates, we manually checked the grammaticality and fixed all the errors.

Quality Check via Crowd-Sourcing. We hired three crowd-workers per video on AMT to ascertain the quality of all auto-generated answers. For each video, a worker is presented with the first 20% of the video and the future-directed intents, which are paired with six answer candidates each (an original action sequence and five generated ones), because there were two annotators working on each video. They were instructed to choose the most reasonable pair of intent and action sequence among all possible combinations.

After checking the answers of all questions in the **test set**, we apply a set of heuristic rules to determine the final answer to each question. We calculate inter-annotator agreement by asking the group of workers that did the annotation to work on a sample of multi-choice questions of the MQA task. To evaluate the quality of the MQA choices, we determined the number of agreements between the ground truth (the correct answers) and the predicted answers. Then, we computed the number of agreements that would be expected by chance based on the distribution of answers. The corresponding Cohen’s kappa coefficient (Kraemer, 2014) is 0.91, which demonstrates the high quality.

Table 2 shows the basic statistics of the test set. The average number of observed actions in s is similar to the average number of future actions. Although all actions in the test appear in the training set, the most plausible action sequences of almost 400 videos are unseen in the training set. For intents in MQA, we also calculate the number of distinct future action sequences for each of them, and the standard deviation across all of them. The results indicate how diverse potential future action sequences can be for a single intent. Other details of MQA can be found in Appendix 7.2.

4 Baselines

VL planning of human activities requires predicting future action sequences given an initial visual

state video and an intent provided in textual form. The task poses two major challenges. First, information provided in two modalities are complementary to each other, while the majority of multimodal research focuses on the shared information by exploring fusion techniques (Guo et al., 2019). Second, the output space is exponentially large with respect to the action space. It is not realistic to assume that all action sequences are already observed in the training data. Hence, any models to tackle this task are expected to address *systematic composition* (Fodor and Pylyshyn, 1988) of human activities, the capacity to understand and produce a huge number of novel combinations of known actions. In contrast, state-of-the-art deep learning methods often perform poorly on compositional generalization (Lake, 2019; Keysers et al., 2019).

We compare deep generative models and a neurosymbolic planning model in the framework of retrieval and reasoning. Given the first 20% of a video and a future-directed intent, the first step is to obtain top- k relevant action sequences, followed by performing reasoning over the top- k action sequences to find the most plausible answers. Both types of models share the same reasoning module but differ in how they obtain top- k action sequences. For reproducibility, the details of all models are provided in Appendix 7.3 and 7.4.

4.1 Deep Generative Models

The deep generative models apply beam search to produce the top- k most likely future action sequences, followed by performing reasoning.

ACT-UNIVL We adapt UNIVL (Luo et al., 2020) for the target task (denoted as ACT-UNIVL), which is a SOTA unified pretrained vision-language model for multimodal understanding and generation. We consider ACT-UNIVL because it performs the best on the tasks that are closest to our target task, such as YouCook2 (Zhou et al., 2017). The pre-trained ACT-UNIVL takes as input an intent and an initial video clip, and is fine-tuned to forecast future action sequences.

Two Stage Planning Model. The two stage planning baseline, **TwoStagePlan** for short, starts by converting an initial video clip into an action sequence in text by using ACT-UNIVL, followed by applying a pre-trained language model, ProphetNet (Qi et al., 2020a) (denoted as ACT-PROPHETNET for ViLPAct), to predict future actions.

ACT-PROPHETNET To study the impact of visual information, we consider a text-only baseline by employing ACT-PROPHETNET to predict future action sequences only based on intents.

4.2 Neurosymbolic Planning Model

Given an intent and an initial visual state, the neurosymbolic planning model (**NSPlan**) retrieves top- k relevant action sequences from the MKB in two stages, and then utilizes the retrieved results to infer the most plausible answers.

In the first stage, we apply the pretrained ACT-UNIVL to convert a video clip into an action sequence and send it as a query to the MKB to retrieve top-50 results. For each retrieved result, the ranking score is the weighted sum of the BM25 (Robertson and Walker, 1994) score between two action sequences and the cosine similarity between the intents.

In the second stage, it re-ranks the initial retrieval results by using both visual and symbolic knowledge. Each retrieved action sequence is represented as a sequence of frame-level visual feature vectors, extracted by the visual encoder I3D. An Ordered Temporal Alignment Module (OTAM; Cao et al. 2020) is applied to compare two visual feature sequences. In order to rank the sequences with potential future actions higher, we use a rule-based score function to prefer longer sequences containing unseen actions. In the end, we keep only the top- k results for probabilistic reasoning.

4.2.1 Probabilistic Reasoning for MQA

We propose a novel approach for MQA called *ProbInf*, which, based on the top- K action sequences, performs probabilistic inference over the retrieved action sequences to identify the most likely answer for a question. From each retrieved result after re-ranking, obtained from NSPlan, we remove the predicted observed action sequence s_q^a to obtain potential future action sequences. For generative models, we directly use the generated outcomes. For each answer candidate c_i of a question, we compute $p(c_i | \mathbf{s}, \mathbf{g})$ by integrating over all retrieved results $\{r_1, r_2, \dots, r_K\}$, given the initial visual state \mathbf{s} and intent \mathbf{g} :

$$p(c_i | \mathbf{s}, \mathbf{g}) = \sum_{k=1}^K p(c_j | r_k) p(r_k | \mathbf{s}, \mathbf{g}), \quad (2)$$

where $p(r_j | \mathbf{s}, \mathbf{g}) = \frac{\exp(s_f(r_j))}{\sum_{k=1}^K \exp(s_f(r_k))}$ is the normalized ranking score for a result r_j and $p(c_j | r_k)$

is the normalized similarity between an answer candidate and each retrieved result. As both answers and retrieved results are action sequences represented in text, we employ the time series metric Time-warped edit distance (TWED; Marteau 2009) to compute their similarity as $\phi(f(c_i), f(r_j)) = 1 - d_{\text{twed}}(f(c_i), f(r_j)) / \max(|c_i|, |r_j|)$, where $f(c_i)$ denotes the visual prototype representation of an action sequence and $d_{\text{twed}}(f(c_i), f(r_j))$ denotes the distance computed by TWED algorithm. Then the normalized similarity over n possible answers of a question is given by:

$$P(c_i | r_j) = \frac{\exp \phi(f(c_i), f(r_j))}{\sum_{k=1}^n \exp \phi(f(c_k), f(r_j))} \quad (3)$$

The most plausible answer is the one with the maximal $p(c_j | \mathbf{s}, \mathbf{g})$ over all answer candidates.

5 Experiments

We conduct extensive experiments to answer the following three main research questions. The other research questions are addressed in Appendix 7.9.

Method	TwoStagePlan	NSPlan	ACT-PROPHETNET	ACT-UNIVL
Log-likelihood Accuracy(%)	19.02	-	10.78	22.35
top-1 Reasoner-scoring Accuracy(%)	63.72	60.58	67.45	69.01
top-10 Reasoner-scoring Accuracy(%)	60.19	64.11	69.01	70.58

Table 3: Comparison of all systems, with **Human performance** of 94.25% accuracy, which is obtained by asking humans to answer the MQAs directly.

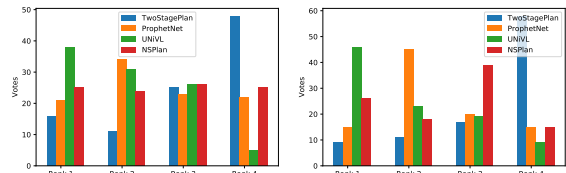


Figure 4: Human evaluation on the quality of top-10 (left) & top-1 (right) future action sequence.

RQ1: How reliable is the MQA evaluation method? We show that the evaluation results using MQA are consistent with those by asking humans to directly observe model outputs. For this, we recruit five crowd-workers to rank all models in comparison on each of the 100 questions randomly sampled from the test set, and compare them with the corresponding results using MQA. Specifically, for each question, a crowd-worker is asked to rank the top- k outputs of the four baselines in terms of how well they match the intent and the remaining 80% of the original videos. As a result, Figure 4 shows how frequent each model is ranked at position X judged by the crowd-workers w.r.t. the top-10 predictions (left) and top-1 predictions (right),

respectively. In both cases, we consistently find that the best model is ACT-UNIVL, followed by ACT-PROPHETNET, NSPlan, and TwoStagePlan. The ranking result is the same as using MQA on the same set of questions. The ranking differences on individual questions between the human evaluation and MQA are statistically insignificant according to Wilcoxon’s signed-rank test (Woolson, 2007), details of which can be found in Appendix 7.8.

Method	TwoStagePlan	NSPlan	ACT-PROPHETNET	ACT-UNIVL
Seen Accuracy(%)	60.33	65.28	70.24	74.38
UnSeen Accuracy(%)	60.15	63.75	68.63	69.40

Table 4: Top-10 Reasoner-scoring Accuracy on seen and unseen action sequences. Seen data refers to the MQAs with plausible action sequences observed in the training data. Unseen data refer to the ones with plausible action sequences not observed in the training data.

RQ2: What are the key challenges? We identify two major challenges of the target task.

Compositional Generalization Using Reasoning.

It is common practice to rank each answer by the likelihood yielded by a generative model (Holtzman et al., 2021). However, Table 3, which provides the overall evaluation results using MQA, shows that the generative baselines perform poorly when they rank answers based on the likelihood. In contrast, *ProbInf* effectively uses top- k results to boost the performance of all generative models by more than 44%. For the respective performance on seen and unseen action sequences (Table 4), *ProbInf* delivers stable results across models. *The performance on unseen combinations of seen actions measures exactly the ability of compositional generalization.* This raises the question of “*Why ProbInf helps compositional generalization ?*” for future research. As there is still a sizable gap between seen and unseen action sequences, and all models fall short of the human performance (Table 3) by at least 23%, how could we make further improvements?

Effective Use of Both Modalities. To understand the utility of each modality, we compare the two strongest multimodal models by varying their inputs: including both modalities or just a single modality. As shown in Table 5, intents provide the strongest signal, while visual information is useful overall for both models. This also explains why ACT-PROPHETNET comes close to ACT-UNIVL.

To further investigate the significance of visual information for multimodal models, we substitute

ACT-UNIVL w/o Vision	ACT-UNIVL
69.01	70.58 ↑
NSPlan w/o Vision	NSPlan
61.56	64.11 ↑
ACT-UNIVL w/o Intent	ACT-UNIVL
61.17	70.58 ↑
NSPlan w/o Intent	NSPlan
60.78	64.11 ↑

Table 5: Modality study on MQA accuracies (%) of different baselines via Reasoner-scoring.

the visual features of ACT-UNIVL for randomly selected ones during both training and inference, finding that ACT-UNIVL suffers from only a 4% drop of accuracy using MQA. Hence, the multimodal models capture only weak associations between visual features and future action sequences.

It is counter-intuitive that visual features do not play a significant role, because plans vary in accordance with different visual environments. We conjecture this is due to poor performance of action recognition. To verify this, we feed ground-truth actions observed in the first 20% of videos to both TwoStagePlan and NSPlan during training and inference. They reach an accuracy of 82.11% and 81.37% respectively, improved by more than 15%.

RQ3: To what degree can the top- k results reflect the performance differences of systems?

The reasoning method *ProbInf* leverages the top- k results produced by the models, hence it is useful to inspect those results for further insights. Therefore, we compare the top 10 results of each model in terms of precision and recall by treating each action sequence as a set (Ng and Fernando, 2020), as well as seq-hits@5 for measuring exactly matched action sequences. Moreover, to investigate the *diversity* of the top- k lists, we consider Dist1 and Dist2 (Li et al., 2016), which respectively measure the number of unique action and consecutive action pairs in the top- k lists. The definitions of a complete list of used metrics and their results are provided in Appendix 7.6 and 7.4.1.

According to Table 6, ACT-UNIVL outperforms all other models in terms of quality-oriented metrics but falls short of ACT-PROPHETNET in terms of both diversity metrics. However, none of the metrics obtains the same ranking of models in accordance with the human evaluation. Although NSPlan achieves higher recall than ACT-PROPHETNET, its precision and seq-hits@5 are significantly lower than those of ACT-PROPHETNET, explaining why it performs worse than ACT-PROPHETNET using MQA.

Setting	Quality			Diversity	
	precision	recall	seq-hits@5	Dist1	Dist2
TwoStagePlan	21.59	15.59	10.00	32.55	58.54
NSPlan	20.73	21.66	5.69	38.46	66.34
ACT-PROPHETNET	21.35	9.61	8.12	51.96	81.93
ACT-UNIVL	23.67	22.02	12.75	47.10	77.42

Table 6: Comparison of top-10 future sequences

6 Conclusion

We construct the novel benchmark ViLPAct to evaluate the ability of systems to anticipate and plan human actions in a multimodal vision-language setting, with a focus on evaluating their compositional generalization capabilities. In this benchmark, we extend Charades with intents, construct a test set with multi-choice questions, and include four strong baselines. Our empirical studies demonstrate that the task is easy for humans, but challenging for SOTA deep learning models due to the need for compositional generalization and an effective use of information from both modalities. The neurosymbolic planning baseline shows a promising research avenue for using symbolic and multimodal knowledge in an MKB.

Ethical Considerations

In order to mitigate the potential for exposure to problematic content in the Charades video dataset, we have implemented stringent safety measures to safeguard our annotators against adverse psychological effects. To ensure the suitability of the video content, the authors initially conducted a comprehensive review. However, it is recognized that the process of annotating feedback may still result in the exposure to potentially disturbing or offensive material. To mitigate this, we only engage annotators who are of legal age and clearly communicate that discretion is strongly advised when engaging in the annotation process. In the event that an annotator experiences discomfort or distress, we provide information on how they can seek support from the Substance Abuse and Mental Health Services Administration (SAMHSA)⁴, a free and confidential resource available 24/7. In addition, we have established a feedback mechanism to allow annotators to communicate their concerns in real-time. Our response time to any feedback received is within 24 hours. Furthermore, we compensate our annotators with competitive wages, with an average hourly rate of approximately \$12.

⁴<https://www.samhsa.gov/>

Limitations

In this work, we have proposed a new vision-language benchmark for compositional generalization on human activities. Although it contains numerous videos and diverse actions, it only emphasizes in-door activities, which is a subdomain of human activities. We encourage future research to investigate the compositional generalization on various scenarios of outdoor activities. In addition, despite the fact that our benchmark contains a reasonable number of actions, these actions are constrained by limited types of verb and noun phrases, due to the nature of Charades. We suggest the development of a more extensive dataset covering open-vocabulary actions in future applications.

Acknowledgements

This work was partially supported by the National Key Research and Development Program of China (No. 2018AAA0100204), a key program of fundamental research from Shenzhen Science and Technology Innovation Commission (No. JCYJ20200109113403826), the Major Key Project of PCL (No. PCL2021A06), an Open Research Project of Zhejiang Lab (NO.2022RC0AB04), and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (No. 2022B1212010005).

References

- Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. 2019. Chasing ghosts: Instruction following as bayesian state tracking. *arXiv preprint arXiv:1907.02022*.
- Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.
- Tom Bylander. 1994. The computational complexity of propositional strips planning. *Artificial Intelligence*, 69(1-2):165–204.
- Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Nieves. 2020. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10618–10627.

- João Carreira and Andrew Zisserman. 2017. [Quo vadis, action recognition? A new model and the kinetics dataset](#). *CoRR*, abs/1705.07750.
- Howard Chen, Alane Suhr, Dipendra Misra, Noah Snaveley, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.
- Philip R Cohen and Hector J Levesque. 1990. Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213–261.
- Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10.
- Harm De Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. Talk the walk: Navigating New York City through Grounded Dialogue. *arXiv preprint arXiv:1807.03367*.
- Zhiwei Deng, Karthik Narasimhan, and Olga Rusakovsky. 2020. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *arXiv preprint arXiv:2007.05655*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.
- Jonathan Francis, Nariaki Kitamura, Felix Labelle, Xiaopeng Lu, Ingrid Navarro, and Jean Oh. 2021. Core challenges in embodied vision-language planning. *arXiv preprint arXiv:2106.13948*.
- Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. *arXiv preprint arXiv:1806.02724*.
- Artur d’Avila Garcez and Luis C Lamb. 2020. Neurosymbolic ai: the 3rd wave. *arXiv preprint arXiv:2012.05876*.
- Wenzhong Guo, Jianwen Wang, and Shiping Wang. 2019. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394.
- Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. 2020. Learning to follow directions in street view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11773–11781.
- Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn’t always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.
- Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. *arXiv preprint arXiv:1905.12255*.
- Peter A Jansen. 2020. Visually-grounded planning without vision: Language models infer detailed plans from high-level instructions. *arXiv preprint arXiv:2009.14259*.
- Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.
- Diptesh Kanojia, Marina Fomicheva, Tharindu Ranasinghe, Frédéric Blain, Constantin Orăsan, and Lucia Specia. 2021. Pushing the right buttons: Adversarial evaluation of quality estimation. *arXiv preprint arXiv:2109.10859*.
- Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. 2019. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6741–6749.
- Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*.
- Hyoungun Kim, Abhay Zala, Graham Burri, Hao Tan, and Mohit Bansal. 2020. Arramon: A joint navigation-assembly instruction interpretation task in dynamic environments. *arXiv preprint arXiv:2011.07660*.
- Yu Kong and Yun Fu. 2018. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*.
- Helena C Kraemer. 2014. Kappa coefficient. *Wiley StatsRef: statistics reference online*, pages 1–4.
- Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. *arXiv preprint arXiv:1906.05381*.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*.
- Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. 2019. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6732–6740.
- Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. 2021. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80(6):9411–9457.
- Pierre-François Marteau. 2009. [Time warp edit distance with stiffness adjustment for time series matching](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping Instructions to Actions in 3D Environments with Visual Goal Prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84.
- Yan Bin Ng and Basura Fernando. 2020. Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting. *IEEE Transactions on Image Processing*, 29:8880–8891.
- Khanh Nguyen and Hal Daumé III. 2019. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. *arXiv preprint arXiv:1909.01871*.
- Pranav Pandey and Jagannath V Aghav. 2020. Pedestrian–autonomous vehicles interaction challenges: a survey and a solution to pedestrian intent identification. In *Advances in Data and Information Sciences*, pages 283–292. Springer.
- Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020a. ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410.
- Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020b. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SI-GIR'94*, pages 232–241. Springer.
- Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. 2019. Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32.
- Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*.
- Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer.
- Alane Suhr, Claudia Yan, Jacob Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. *arXiv preprint arXiv:1910.03655*.
- Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.
- Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. 2019. Embodied question answering in photorealistic environments

with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668.

Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3.

Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. 2018. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [From recognition to cognition: Visual commonsense reasoning](#).

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2017. [Towards automatic learning of procedures from web instructional videos](#).

7 Appendix

7.1 Action Sequence Extraction Algorithm

Each video of Charades is annotated with actions from at least one action sequence. The starting and ending points of an action are labelled, but it is not clear which actions jointly meet an intent. Therefore, we implement the greedy method in Algorithm 1 to automatically extract action sequences with clear intents from videos. For each video, the algorithm aims to identify a sequence of temporally and semantically coherent actions, which interact with the same or related objects. The scoring functions in Algorithm 1 measure coherence from three perspectives: i) semantic relevance based on TF-IDF (Jones, 1972) reweighted Word2Vec embeddings (Mikolov et al., 2013), ii) temporal relevance, iii) task relevance. Each action is assigned to one of 22 tasks manually, for example, "Opening a book" and "Closing a book" are assigned to the same task.

7.2 Other Data Details

An example of future action sequences of a selected intent is given in Figure 5. All of these conclusions pose a challenge not only for the generalization of multimodal matching, but also for compositional generalization.

Algorithm 1: Extract Action Sequences

Input: $Actions = \{a_1, a_2, \dots, a_n\}$, each action $a_i = \langle cls^{a_i}, t_s^{a_i}, t_e^{a_i} \rangle$, where cls^{a_i} is the action class, $t_s^{a_i}$ and $t_e^{a_i}$ is the start time and end time of action a_i . Relevance threshold

Output: $Activities = \{A_1, A_2, \dots, A_n\}$, where each activity represents an action sequence

Remaining actions set $R_a = Actions$

while $R_a \neq \emptyset$ **do**
 Sort R_a in ascending order by start time t_s
 pre action $a = R_a[0]$
 Activity $A = \{a\}$
 $Search = True$
 while $Search$ **do**
 candidates $C_a = \{a_j \in R_a | t_s^{a_j} \geq t_s^a\}$
 for $a_j \in C_a$ **do**
 Calculate relevance score: $s_{a_j} =$
 $score(a, a_j) = f_{semantic}(a, a_j) +$
 $f_{time}(a, a_j) + f_{task}(a, a_j)$.
 Where
 $f_{semantic}(a, a_j) = cosine(E_a, E_{a_j})$,
 $E_a = \sum_{w \in cls^a} TFIDF(w) * w2v(w)$,
 $f_{time}(a, a_j) =$
 $(1 - atanh(|t_s^a - t_s^{a_j}|) * \pi/2)$
 $f_{topic}(a, a_j) = \mathbf{1}(task^a = task^{a_j})$
 end
 $a_{max} = argmax(\{s_{a_j} | a_j \in C_a\})$
 if $s_{a_{max}} < threshold$ (1.3 by optimization) **then**
 Append A to $Activities$
 $Search = False$
 else
 Add a_{max} to Activity
 Remove a_{max} from R_a
 pre action $a = a_{max}$
 end
 end
end

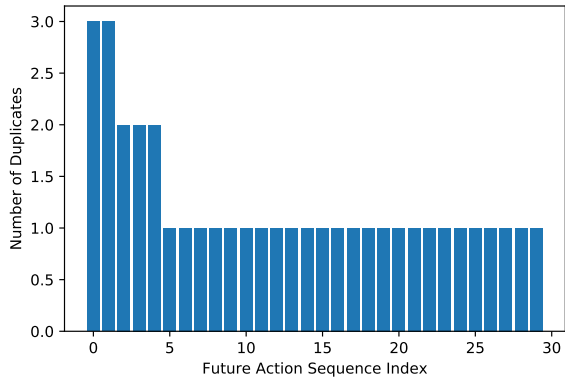


Figure 5: An example of the future action sequence frequency distribution of the *intent* "S/He wants to satisfy my hunger". There are 30 distinct future action sequences matching this intent.

7.3 Deep Generative Models Details

We mainly adapt the multimodal deep planning model ACT-UNIVL to tackle our task. The training set of ACT-UNIVL consists of 2,402 videos, each of which contains a video clip of the initial state s , an observed action sequence a^i , an intent g , and a future action sequence a^f . Both models are trained to minimize prediction errors of a^f .

ACT-UNIVL ACT-UNIVL (Luo et al., 2020) is a SOTA unified pretrained vision-language model for multimodal understanding and generation. We consider ACT-UNIVL because ACT-UNIVL still performs the best on video captioning tasks, such as YouCook2 (Zhou et al., 2017). YouCook2 contains task-oriented and instructional third-person videos about indoor cooking. The captions of a video are provided for the whole video without explicit alignments at the frame or segment levels. In addition, ACT-UNIVL considers two sources of textual inputs: transcripts and captions. Hence, it is most close to our target task. Taking as input a future-directed intent and a video clip of the initial state, ACT-UNIVL is fine-tuned to forecast future action sequences.

More specifically, we utilize ACT-UNIVL to map a video clip to a sequence of action names. Most of the action names are multi-word expressions. During training, ACT-UNIVL takes as input both the visual features of a video clip s and an observed action sequence a^i , and optimizes the model with multiple pre-training objectives. The visual features are extracted by the I3D model (Carreira and Zisserman, 2017) trained on Charades. During prediction, the model generates a future ac-

tion sequence by only taking an initial visual state and high-level intent as input. To fine-tune ACT-UNIVL, we set the max. frame, mean frame and feature frame rate of the encoded features to be 629, 113 and 3. We fine-tune ACT-UNIVL on two NVIDIA V100 GPUs for 50 epochs and choose the best one based on the BLEU-3 metric.

Two Stage Planning Model. The two stage planning baseline, **TwoStagePlan** for short, starts by converting the initial visual state s into a textual description of the observed action sequence, followed by applying a Seq2Seq language model, ACT-PROPHETNET (Qi et al., 2020a), to predict future actions.

At Stage 1, we adopt ACT-UNIVL on the video captioning task. Different from the single ACT-UNIVL baseline, we only train it with observed video clip inputs and let it generate the corresponding captions for observed action sequences. The other settings and training settings remain the same as for the single ACT-UNIVL baseline.

Given an observed action sequence recognized by ACT-UNIVL, we fine-tune ACT-PROPHETNET by following Jansen (2020) in Stage 2. We prefer ACT-PROPHETNET over GPT2 (Radford et al., 2019) because it can learn to predict n future tokens jointly, which is computationally efficient and mitigates overfitting on strong local correlations. For each video, we take as input the intent and the observed action sequence, separated by a special token *SEP*, and train the model to minimize prediction errors of future action sequences. Fine-tuning the model from the PROPHETNET-EN pretrained checkpoint for 50 epochs on 2 Nvidia Tesla V100 GPUs, we choose the best model based on the validation loss.

ACT-PROPHETNET To study the impact of visual information, we consider a text-only baseline by employing ACT-PROPHETNET. Herein, ACT-PROPHETNET takes as input an intent and generates the future action sequences. The training is done with the same training procedure as Stage 2 of TwoStagePlan. This model serves for an ablation study, in contrast to TwoStagePlan, which uses additionally recognized action sequences as input.

7.4 Neurosymbolic Planning Model

Instead of using the data in the training set to directly optimize model parameters, the neurosymbolic planning model (**NSPlan**) builds an MKB from the training data. Given a question in the test

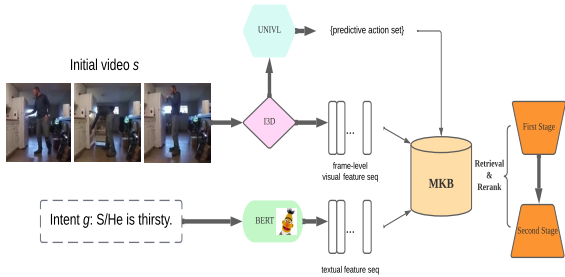


Figure 6: The neurosymbolic planning model is a multi-modal retrieval & re-rank pipeline.

set, the model retrieves relevant knowledge based on the initial visual state and the intent, and then applies the retrieved knowledge to infer the most plausible answers from all available choices.

7.4.1 Retrieval from Multimodal Knowledge Base

The neurosymbolic planning model retrieves relevant action sequences from the MKB in two stages. The first stage aims to computationally efficiently obtain all relevant action sequences. At the second stage, it re-ranks the initial retrieval results by using both visual and symbolic knowledge.

First Stage. Given the initial state of a video, we apply the pretrained ACT-UNIVL model used in the two-stage planning model to predict a sequence of observed actions. Then this action sequence in text form is sent as query to retrieve top-50 relevant action sequences from the MKB. For each retrieved result, the ranking score is the weighted sum of the BM25 (Robertson and Walker, 1994) score between two action sequences and the cosine similarity between the intents. At this stage, only textual information is taken into account, and the temporal order of actions in a sequence is not considered because BM25 considers each action sequence as a bag of words.

Second Stage. We re-rank the results from the first stage by taking temporal order and the visual features of action sequences into account. Each action sequence is represented as a sequence of frame-level visual feature vectors, which are extracted by the same visual encoder I3D. We apply the Ordered Temporal Alignment Module (OTAM) (Cao et al., 2020) to compare two visual feature sequences. OTAM computes a distance between a pair of sequences by integrating video segment distances only along the ordered temporal alignment path. We turn a distance into an alignment score by

$s_{\text{align}} = 1/(1 + d_{\text{otam}})$, where d_{otam} denotes the OTAM distance.

Many retrieved action sequences do not contain future actions. In order to rank the sequences with potential future actions higher, we add a rule to encourage long sequences containing unseen actions. The rule score $s_{\text{rule}} = s_{\text{last}} + s_{\text{len}}$ is the sum of two binary indicator functions s_{last} and s_{len} , where $s_{\text{last}} = 1$ if and only if the last action of the retrieved result is not contained in the query set, and $s_{\text{len}} = 1$ if and only if the length of the retrieved result is greater than that of the query. The final ranking score $s_f(r)$ of a result r is the weighted sum of the initial ranking score, the alignment score s_{align} and the rule-driven score s_{rule} . To reduce noise, we keep only the top-10 results for probabilistic reasoning. We provide a completed version of the comparison among all baselines on future sequence evaluation in Table 7.

7.5 Full Action Sequence Comparison

7.6 Metrics

- Seq-item-acc: Sequence item classification accuracy evaluates the exact action matching of the predicted action sequence with the ground truth, counting how many times the action in the predicted sequence matches the ground truth at the exact position. For top-10 sequences, we calculate the mean accuracy of all sequences.
- Precision and recall: The precision and recall do not consider the order of ground truth. They both treat the actions inside the sequence as a unified set. The precision of top-10 sequences is computed by averaging the precision of each sequence, which measures the number of true actions over the number of total actions in the sequence. Here, we define the true action as the action that occurred in the ground truth. Similarly, the recall of top-10 sequences is also computed by averaging all sequences' recall, which is a measure of the true actions over the number of ground truth actions.
- Seq-hit@ k Rate: The seq-hits scores measure the exact sequence matches, calculated as the number of examples whose top- k sequences include the ground truth sequence, and we report the seq-hits@5 and seq-hits@10 accordingly. As for the retrieval-based baseline, we

setting	Quality							Diversity	
	precision	recall	seq-item-acc	seq-hits@5	seq-hits@10	BLEU-1	BLEU-2	Dist1	Dist2
TwoStagePlan	21.59	15.59	9.26	10.00	16.86	12.50	3.58	32.55	58.54
NSPlan	20.73	21.66	8.74	5.69	7.65	19.25	6.80	38.46	66.34
ACT-PROPHETNET	21.35	19.75	8.12	9.61	10.59	18.66	5.52	51.96	81.93
ACT-UNIVL	23.67	22.02	9.71	12.75	16.08	20.52	6.52	47.10	77.42

Table 7: Comparison of top- k future sequences of all systems.

setting	Quality							Diversity	
	precision	recall	seq-item-acc	seq-hits@5	seq-hits@10	BLEU-1	BLEU-2	Dist1	Dist2
TwoStagePlan	38.71	30.73	11.06	0.59	1.37	29.60	13.71	15.45	35.37
NSPlan	41.63	35.81	11.46	5.69	8.43	34.14	15.90	28.03	62.08

Table 8: Comparison of top-10 full action sequences of all systems.

only consider the in-domain situation where the ground truth sequences have also appeared in the knowledge base.

- BLEU: We use the standard BLEU-1 and BLEU-2 scores that are widely used in the Machine Translation Field and adapt them to our setting by computing the action-level match.
- Dist: We report Dist1 (Distinct-1) and Dist2 (Distinct-2) following the standard definition (Li et al., 2015), to measure the diversity of action sequences, based on the number of distinct N -gram of top-10 sequences.

7.7 Full Table of Future Sequence Evaluation

In Table 8, we compare TwoStagePlan with NSPlan, where both models are designed to output the full action sequence including the observed actions. It turns out that NSPlan performs consistently across all metrics, indicating that NSPlan has a stronger ability to identify the most similar full action sequences in the MKB and training set.

7.8 Wilcoxon’s signed-rank test

Wilcoxon’s signed-rank test is a statistical hypothesis test used either to test the ranking of a set of samples or to compare the rankings of two populations using a set of matched samples. The calculated Wilcoxon signed-rank test t value is 55.5 with a p value of 0.7979, which shows that there is no significant difference between the two sets of human evaluation samples.

7.9 Other Research Questions

How useful are symbolic, neural, or neurosymbolic knowledge? The goal of reasoning is to perform the probabilistic inference $\arg \max_{c_i} p(c_i |$

Method	Action ID	Visual-proto	Text-proto	Visual + text proto
	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)
Mean	53.33	60.19	46.82	48.42
Max-Pooling	43.92	43.52	41.76	42.35
DTW	48.82	61.37	50.98	52.15
TWED	44.50	64.11	48.23	50.19

Table 9: Reasoner-scoring performance with varying combinations of similarity measures and action-level features.

s, g) over all possible answers. One of the key differences of NSPlan from the two generative models is that it introduces the time series similarity TWED to compare the future action sequences with each answer.

To understand the effects of TWED in the probabilistic reasoning module of our retrieval-based baseline, we compare it with three other similarity measures: (a) cosine similarity between the mean vectors of two sequences, (b) cosine similarity between the max-pooling results of two sequences, (c) the time series distance function Dynamic Time Warping (DTW) (Müller, 2007). All of them are evaluated based on the same best performing top-10 retrieval results.

We also evaluate different types of symbolic, neural, and neurosymbolic features used for computing action-level distance inside those measures: (a) action class ID, (b) the visual prototype features in the MKB, (c) the textual prototype features in the MKB, (d) concatenation of the visual prototype features and textual prototype features.

It is clear from Table 9 that TWED using the visual prototype features performs the best. The performance of the two time series metrics are comparable. Combining visual prototype features and textual prototypes features actually harms the performance. This is in contrast to the retrieval evalua-

tion, which finds the symbolic representations most useful. This highlights the flexibility of this hybrid neurosymbolic system, which naturally supports choosing the most appropriate types of information for its respective modules.

We also experiment with the symbolic knowledge described by the STRIPS language. More specifically, we implement a symbolic planner based on STRIPS, which is able to check to what degree each answer is compatible with the preconditions and effects defined for each action class. Such symbolic knowledge can boost the overall accuracy of NSPlan to 82% if we substitute the ground truth actions for the action sequences recognized by ACT-UNIVL. However, if we only use the predictions of ACT-UNIVL, which has both precision and recall around 32%, the overall accuracy drops by almost 10%.