This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Why Not Use Your Textbook? Knowledge-Enhanced Procedure Planning of Instructional Videos

Kumaranage Ravindu Yasas Nagasinghe[1] Honglu Zhou[2] Malitha Gunawardhana[1,3]
Martin Renqiang Min[2] Daniel Harari[4] Muhammad Haris Khan[1]
[1]Mohamed bin Zayed University of Artificial Intelligence, [2]NEC Laboratories, USA,
[3] University of Auckland, [4]Weizmann Institute of Science
ravindu.nagasinghe@mbzuai.ac.ae, muhammad.haris@mbzuai.ac.ae

Abstract

In this paper, we explore the capability of an agent to construct a logical sequence of action steps, thereby assembling a strategic procedural plan. This plan is crucial for navigating from an initial visual observation to a target visual outcome, as depicted in real-life instructional videos. Existing works have attained partial success by extensively leveraging various sources of information available in the datasets, such as heavy intermediate visual observations, procedural names, or natural language step-by-step instructions, for features or supervision signals. However, the task remains formidable due to the implicit causal constraints in the sequencing of steps and the variability inherent in multiple feasible plans. To tackle these intricacies that previous efforts have overlooked, we propose to enhance the agent's capabilities by infusing it with procedural knowledge. This knowledge, sourced from training procedure plans and structured as a directed weighted graph, equips the agent to better navigate the complexities of step sequencing and its potential variations. We coin our approach KEPP, a novel Knowledge-Enhanced Procedure Planning system, which harnesses a probabilistic procedural knowledge graph extracted from training data, effectively acting as a comprehensive textbook for the training domain. Experimental evaluations across three widely-used datasets under settings of varying complexity reveal that KEPP attains superior, state-of-the-art results while requiring only minimal supervision. Code and trained model are available at https://github.com/Ravindu-Yasas-Nagasinghe/KEPP

1. Introduction

The evolution of the internet has precipitated an unprecedented influx of video content, serving as a vital educational resource for myriad learners. Individuals frequently leverage platforms such as YouTube to acquire new skills, ranging from culinary arts to automobile maintenance [34]. While these instructional videos benefit the development of intelligent agents in mastering long-horizon tasks, the challenge extends beyond merely interpreting visuals. It requires high-level reasoning and planning to effectively assist in complex, real-life scenarios [13].

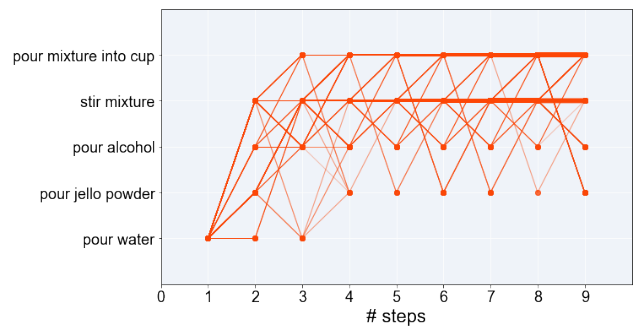

Figure 1. **Expert trajectories [7] of the 'Make Jello Shots' task from the CrossTask dataset [63].** Heavier color indicates more frequently visited path. This depicts the complexities of the procedure planning task, arising from the subtle causal links in step sequencing (e.g., steps like 'stir mixture' or 'pour mixture' typically occur after adding individual ingredients), the varied probabilities of transitioning between steps, and the diversity in plans viable for a given starting point and an intended outcome. Motivated by these nuanced challenges, we propose Knowledge-Enhanced Procedure Planning (KEPP) with the use of a probabilistic procedure knowledge graph to capture and represent these intricacies

Procedure Planning in Instructional Videos demands an agent to produce a sequence of actionable steps, thereby crafting a procedure plan that facilitates the transition from an initial visual observation of the physical world towards achieving a desired goal state [7, 9, 50, 53, 54, 59]. The task acts as a precursor to an envisioned future scenario in which an agent like a robot provides on-the-spot support, such as assisting an individual in preparing a recipe [6].

Current methods in procedure planning in instructional videos make extensive use of various annotations available within the datasets to enrich input features or provide supervisory signals (see Table 1). These include detailed, temporally localized visual observations of intermediate action steps throughout the procedure plan [7, 9, 50], high-level procedural task labels [53, 54], and step-by-step instructions in natural language [53, 59]. Despite advancements, significant challenges persist, including characterizing the implicit causal constraints in step sequencing, the varied probabilities of transitioning between steps, and the inherent variability of multiple viable plans (see Fig. 1).

To address these intricacies that previous efforts have overlooked, we propose to enhance the agent's capabilities in procedure planning by infusing it with comprehensive procedural knowledge [62], derived from training procedure plans and structured as a directed weighted graph. This graph, as a Probabilistic Procedural Knowledge Graph [5] where nodes denote steps from diverse tasks and edges represent step transition probabilities in the training domain, empowers agents to more adeptly navigate the complexities of step sequencing and its potential variations.

Our proposed approach, KEPP, is a novel Knowledge-Enhanced Procedure Planning system (Fig. 2) that harnesses a probabilistic procedural knowledge graph (P^2KG), constructed from training procedure plans. This graph functions like a detailed textbook, providing extensive knowledge for the training domain, and thereby circumventing the need for costly multiple annotations required by existing methods. Additionally, we decompose the instructional video procedure planning problem into two parts: one driven by objectives specific to step perception and the other by a procedural knowledge-informed modeling of procedure planning. In this problem decomposition, the first and last action steps are predicted based on the initial and goal visual states. Following this, a procedure plan is crafted by leveraging the procedure plan recommendations retrieved from the P^2KG. The recommendations correspond to the most probable procedure plans frequently used in training, conditioned on the predicted first and last action steps. In a similar vein to the approach by Li *et al*. [29], our proposed decomposition strategy reduces uncertainty by maximizing the use of currently available information, namely the initial and goal visual states. This allows for the improvement of procedure planning through more accurate predictions of start and end actions. Plus, this decomposition effectively incorporates procedural knowledge into procedure planning, thereby enhancing its effectiveness.

Our contributions are as follows:

- We propose KEPP, a Knowledge-Enhanced Procedure Planning system for instructional videos that leverages rich procedural knowledge from a probabilistic procedural knowledge graph (P^2KG). This approach necessitates only a minimal amount of annotations for supervision.
- We decompose the problem in procedure planning of instructional videos: predicting the initial and final steps from the start and end visuals, and then creating a plan using procedural knowledge retrieved based on these predicted steps. This approach prioritize the currently available information and effectively incorporates procedural knowledge, enhancing strategic planning.
- Experimental evaluations on three widely-used datasets, under settings of varying complexity, reveal that KEPP attains state-of-the-art results in procedure planning.

2. Related Work

Instructional Videos, which demonstrate multi-step procedures, have become a hotbed of research. The studies delve into various aspects, including comprehending and extracting intricate spatiotemporal content from video [12, 18, 19, 21, 23, 36, 43, 44, 48, 55, 57], interpreting the interrelationships between various actions and procedural events [47, 63], and developing capabilities for forecasting [39, 42] and strategic reasoning and planning [28] within the context of these videos. Furthermore, by leveraging the multimodality of visual, auditory, and narrative elements within these videos, research extends to areas like multimodal alignment [2, 58], grounding [10, 14, 25, 33, 51], representation learning [11, 35, 61], pre-training [15, 26, 62], and more [17, 24, 37, 56]. This paper focuses on procedure planning in instructional videos.

Procedure Planning is a vital skill for autonomous agents tasked with handling complex activities in everyday settings. Essentially, these agents must discern the appropriate actions to reach a specific goal. This aspect of artificial intelligence (AI) has been a prominent and integral subject in fields like robotics [20, 28, 32, 45, 46]. Yet, the challenge of procedure planning in the context of instructional videos is notably distinct, and potentially more complex, than its counterparts in natural language processing [8, 30], multimodal generative AI [13, 31], and simulated environments [27, 28, 45]. Its significance is underscored by the need for planning that is grounded in real-world scenes. This requires the development of AI agents capable of accurately perceiving and understanding the current real-world context, and then anticipating and mapping out a logical sequence of actions to fulfill a high-level goal effectively.

Procedure Planning in Instructional Videos has recently garnered research attention. DDN [9] initiates this trend by conceptualizing the problem as sequential latent space planning. Building on this, PlaTe [50] employs transformers for both action and state models, integrating Beam Search for enhanced performance. Meanwhile, Ext-GAIL [7] suggests employing contextual modeling through variational autoencoder and adversarial policy learning. This method considers contextual information as time-invariant knowledge, crucial for distinguishing specific tasks and allowing for multiple planning outcomes.

While these earlier approaches have viewed procedure planning as an autoregressive sequence generation problem, recent methods regard it as a distribution-fitting problem to mitigate error propagation in sequential decisions. In this vein, P^{3}IV [59] replaces intermediate visual states with linguistic representations for supervision, predicting all steps simultaneously instead of using autoregressive methods. To circumvent the complex learning strategies and high annotation costs of previous work, PDPP [54] models the entire intermediate action sequence distribution using a conditioned projected diffusion model. This approach redefines the planning problem as a sampling process from this distribution and simplifies supervision by using only instructional video task labels. E3P [53], also encoding task information, adopts a mask-and-predict strategy for mining step relationships in procedural tasks, integrating probabilistic masking for regularization. In contrast, our approach does not rely on annotations of intermediate states, natural language step representations, or procedural task labels.

Recognizing the difficulties inherent in high dimensional state supervision and the accumulation of errors in action sequences, SkipPlan [29] was developed. It strategically focuses on action predictions, breaking down longer sequences into shorter, more manageable sub-chains by skipping over less reliable intermediate actions. Drawing inspiration from SkipPlan, our approach decomposes the procedure planning problem to prioritize the most reliable information available (*ref.* § 3.1.2). However, we innovate further by incorporating a Probabilistic Procedure Knowledge Graph, significantly enriching the planning phase.

3. Methodology

We will first introduce the problem setup in § 3.1, and then discuss our novel Knowledge-Enhanced Procedure Planning system (KEPP) in § 3.2. See Fig. 2 for KEPP overview.

3.1. Problem and Method Overview

3.1.1 Problem Formulation

We follow the problem definition for procedure planning of instructional videos put forth by Chang *et al.* [9]: given an observation of the initial state v_{start} and a goal state v_{goal}, both are short video clips indicating different states of the real-world environment extracted from an instructional video, a model is required to plan a sequence of action steps $a_{1:T}$ to reach the indicated goal. Here, T is the planning horizon, inputting to the model, corresponding to the number of action steps in the sequence produced by the model so that the environment state can be transformed from v_{start} to v_{goal}. We use a_t to denote the action step at the timestamp t, and in the following, v_s and v_g are short for v_{start} and v_{goal}. Mathematically, the procedure planning problem is defined as $p\left(a_{1:T}|v_s, v_g\right)$ that denotes the conditional probability distribution of the action sequence $a_{1:T}$ given the initial visual observation v_{start} and the goal visual state v_{goal}.

3.1.2 Problem Decomposition

Considering the initial and final visual states are input, providing the most reliable information, we hypothesize that predicting the first and final action steps is more dependable than interpolating the intermediate ones, and consequently, an enhanced accuracy in predicting the first and final steps can lead to more effective procedure planning. Inspired by this hypothesis, we decompose the procedure planning problem into two sub-problems, as shown in Eq. 1:

$$p\left(\hat{a}_{1:T}|v_s, v_g\right) = p\left(\hat{a}_{2:T-1}|\hat{a}_1, \hat{a}_T\right) p\left(\hat{a}_1, \hat{a}_T|v_s, v_g\right), \quad (1)$$

where the first sub-problem is to identify the beginning step a_1 and the end step a_T, and the second sub-problem is to plan the intermediate action steps $a_{2:T-1}$ given a_1 and a_T. We use $\hat{a}_t$ to denote *predicted* action step at timestamp t.

Our proposed problem decomposition in Eq. 1 bears resemblance with the problem formulation from Li *et al.* [29]; they decompose procedure planning into $p\left(\hat{a}_{1:T}|v_s, v_g\right) = \prod_{t=2}^{T-1} p\left(\hat{a}_t|\hat{a}_1, \hat{a}_T\right) p\left(\hat{a}_1, \hat{a}_T|v_s, v_g\right)$. However, our formulation differs in its approach to modeling the second sub-problem. Specifically, we employ a conditioned projected diffusion model (*ref.* § 3.2) to jointly predict $a_{2:T-1}$ at once, whereas Li *et al.* [29] rely on Transformer decoders to predict each intermediate action independently. Further, we integrate a Probabilistic Procedure Knowledge Graph (*ref.* § 3.2.2) to address the second sub-problem.

Tackling the second sub-problem is nontrivial even when armed with an oracle predictor for the first sub-problem. Procedure planning in real-life scenarios remains daunting because of the following **challenges**: *(1)* the presence of implicit temporal and causal constraints in the sequencing of steps, *(2)* the existence of numerous viable plans given an initial state and a goal state, and *(3)* the need to incorporate the real-life everyday knowledge both in task-sharing steps and in managing the inherent variability in transition probabilities between steps. Previous studies tackled these challenges by extensively harnessing detailed annotations in the datasets to augment input features or offer supervision signals (see Table 1). In contrast, we propose harnessing a Probabilistic Procedural Knowledge Graph (P^{2}KG) which is extracted from the procedure plans in the training set. With the P^{2}KG at our hand, we further decompose the procedure planning problem to reduce its complexity and learn $f_\theta : (v_s, v_g, T) \rightarrow p\left(\hat{a}_{1:T}|v_s, v_g\right)$ as follows:

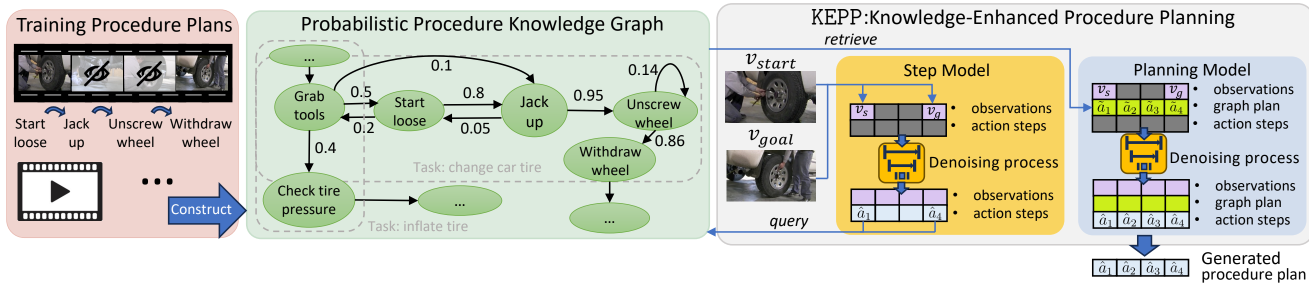

Figure 2. **Overview of our methodology.** We introduce KEPP, a Knowledge-Enhanced Procedure Planning system for instructional videos, leveraging a Probabilistic Procedural Knowledge Graph (P^{2}KG). KEPP breaks down procedure planning into two parts: predicting initial and final steps from visual states, and crafting a procedure plan based on the procedural knowledge retrieved from P^{2}KG, conditioned on the predicted first and last action steps. KEPP requires minimal annotations and enhances planning effectiveness

$$p\left(\hat{a}_{1:T}|v_s, v_g\right) = p\left(\hat{a}_{1:T}|\tilde{a}_{1:T}, v_s, v_g\right) p\left(\tilde{a}_{1:T}|\hat{a}_1, \hat{a}_T\right) p\left(\hat{a}_1, \hat{a}_T|v_s, v_g\right) \tag{2}$$

where f_θ denotes the machine learning model, and $\tilde{a}_{1:T}$ represents a graph path (i.e., a sequence of nodes) retrieved from P^{2}KG. This graph path provides a valuable procedure plan recommendation aligned with the training domain, thus mitigating the complexity of procedure planning. It is worth noting that the proposed approach to modeling procedure planning using Eq. 2 demands only a minimal level of supervision, merely relying on the ground truth training procedure plan; Eq. 2 circumvents the need for additional annotations. We describe details of our P^{2}KG-enhanced approach in the following subsection.

3.2. KEPP: Knowledge-Enhanced Procedure Planning

We propose KEPP (Fig. 2) utilizing a probabilistic procedure knowledge graph extracted from the training set. We firstly identify the beginning and conclusion steps according to the input initial and goal states; and then, conditioned on these steps and the planning horizon T, we query the graph to retrieve relevant procedural knowledge for knowledge-enhanced procedure planning of instructional videos.

3.2.1 Identify Beginning and Conclusion Steps

Given v_{start} and v_{goal} as input, we adapt a Conditioned Projected Diffusion Model [54] (*ref.* supplementary material) to identify the first action step and the final step; we refer to this model as the 'Step (Perception) Model'.

Standard Denoising Diffusion Probabilistic Model tackles data generation through a denoising Markov chain over variables $\{x_N \dots x_0\}$, starting with x_N as a Gaussian random distribution [22]. In the forward diffusion phase, Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is progressively added to the initial, unaltered data x_0, transforming it into a Gaussian random distribution. Conversely, the reverse denoising process transforms Gaussian noise back into a sample. Denoising is parameterized by a learnable noise prediction model, and the learning objective is to learn the noise added to x_0 at each diffusion step. After training, the diffusion model generates data akin to x_0 by iteratively applying the denoising process, starting from random Gaussian noise.

Adopting Conditioned Projected Diffusion Model as the Step Model. For our step model, the distribution we aim to fit is the two-action sequence $[a_1, a_T]$, based on the visual initial and goal states, v_{start} and v_{goal}. These conditional visual states are concatenated with the actions along the action feature dimension, forming a multi-dimensional array:

$$\begin{bmatrix} v_s & 0 & \dots & 0 & v_g \\ a_1 & 0 & \dots & 0 & a_T \end{bmatrix} \tag{3}$$

where the array is zero-padded to have a length corresponds to the planning horizon T. During the denoising process, these conditional visual states can change, potentially misleading the learning process. To prevent this, a condition projection operation [54] is applied, ensuring the visual state and zero-padding dimensions remain unchanged (shaded below). The projection operation is denoted as:

$$\begin{bmatrix} \hat{v}_1 & \hat{v}_2 & \dots & \hat{v}_{T-1} & \hat{v}_T \\ \hat{a}_1 & \hat{a}_2 & \dots & \hat{a}_{T-1} & \hat{a}_T \end{bmatrix} \xrightarrow{\text{Projection}} \begin{bmatrix} v_s & 0 & \dots & 0 & v_g \\ \hat{a}_1 & 0 & \dots & 0 & \hat{a}_T \end{bmatrix} \tag{4}$$

where $\hat{v}_t$ denotes the predicted visual state dimensions at timestamp t within the planning horizon T.

3.2.2 Construct the Probabilistic Procedure Knowledge Graph (P^{2}KG)

The Probabilistic Procedure Knowledge Graph [5] P^{2}KG $= (V, E, w)$ is a directed and weighted graph. In this structure, each step from the training set is represented as a node. During the graph construction process, we iterate over the training procedure plans, and for each direct step transition present in a plan, we add an edge from a_t to a_{t+1} if it does not already exist in the graph; otherwise, we increase its existing frequency count by one. Eventually, this process results in a frequency-based Procedural Knowledge Graph (PKG) [62], which adeptly encapsulating the complexities of step sequencing in procedures and its potential variations, thereby addressing challenges *(1)* and *(2)* of procedure planning (*ref.* § 3.1.2). To further tackle challenge *(3)*, this graph undergoes a transformation into a probabilistic format. In this transformed graph, the edges are not just connections but also signify the likelihoods of transitioning from one step to another. The weight of an edge from a_t to a_{t+1} is the count of transitions from action step a_t to a_{t+1} normalized by total count of a_t being executed [5]. The normalization converts the frequency-based weight into probability distribution and the sum of all out-going edges is one.

3.2.3 P^{2}KG-Enhanced Procedure Planning

Retrieving Procedure Plan Recommendations from the P^{2}KG. Humans use both previously-acquired knowledge and external knowledge when solving problems. The P^{2}KG provides extensive procedural knowledge, serving as a comprehensive textbook, particularly beneficial for the planning model that requires advanced skills.

To utilize this procedural knowledge, queries are made to the P^{2}KG using the first ($\hat{a}_1$) and last ($\hat{a}_T$) actions predicted by the step model. The aim is to find graph paths no longer than T steps, starting from $\hat{a}_1$ and ending at $\hat{a}_T$. This above process often results in multiple possible paths. To evaluate these paths, the probability of each is calculated by multiplying the probability weights of the edges along the path. For instance, the probability of a path $a_1 \rightarrow a_2 \rightarrow a_3$ is determined by the product $w_{a_1 \rightarrow a_2} \times w_{a_2 \rightarrow a_3}$. These paths are then ranked according to their probabilities, and the top R paths are selected as the recommended procedure plans from the P^{2}KG, where R is predefined. For paths shorter than T, padding is applied at any point in the middle of the sequence to explore all possible resultant paths. When R is greater than one, the top R paths are aggregated through linear weighting into a single path (See section A.2 of supplementary material). This final path is then used as an additional input for the procedure planning model, thereby enhancing its decision-making process.

Adopting Conditioned Projected Diffusion Model as the Planning Model. For the planning model, the conditional visual states and the procedure plan recommendation from the P^{2}KG are concatenated with the actions along the action feature dimension, forming a multi-dimensional array:

$$\begin{bmatrix} v_s & 0 & \dots & 0 & v_g \\ \tilde{a}_1 & \tilde{a}_2 & \dots & \tilde{a}_{T-1} & \tilde{a}_T \\ a_1 & a_2 & \dots & a_{T-1} & a_T \end{bmatrix} \tag{5}$$

The rest process is similar to the step model, except that the project operation guarantees that three specific aspects remain unaltered–the dimensions of the the visual state, P^{2}KG recommendation, and zero-padding.

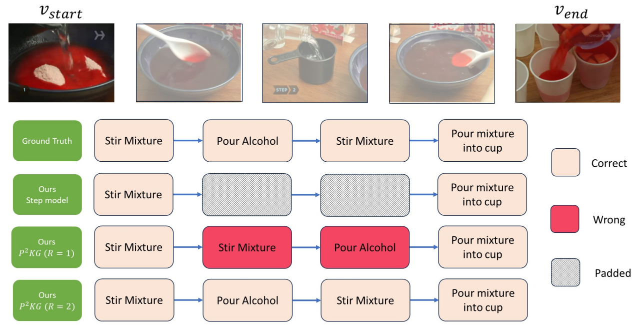

Figure 3. Qualitative analysis of the 'Make Jello Shots' task

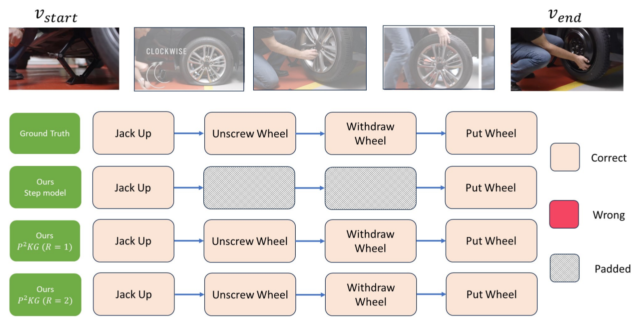

Figure 4. Qualitative analysis of the 'Change a Tire' task

4. Experiments

Datasets and implementation Details: In our evaluation, we employed datasets from three sources: CrossTask [63], COIN [52], and the Narrated Instructional Videos (NIV) [4]. See section C.4 of supplementary material for details on datasets. All ablation studies and analyses were conducted on CrossTask. We use two Tesla A100 GPUs for all the experiments. We chose horizon $T \in \{3, 4, 5, 6\}$ and P^{2}KG (R=1) condition for implementation. In some cases, we incorporate P^{2}KG (R=2) and LLM conditions which are indicated in the respective tables. Throughout this study, the P^{2}KG (R=1) is employed with a batch size of 256, unless explicitly stated otherwise. More implementation details are available in the section C.2 of the supplementary material.

Evaluation Metrics and baselines: We use mean intersection over union (mIoU), mean accuracy (mAcc), and success rate (SR) as evaluation metrics. **SR is the most stringent metric**. See sec. C.4 of supplementary for more details. We compare our model with state-of-the-art methods: WLTDO [16], UAAA [1], UPN [49], DDN [9], PlaTe [50], Ext-GAIL [7], P^{3}IV [59], PDPP [54], SkipPlan [29], and E3P [53]. More details of these methods are available in the section C.5 of supplementary material. Compared to other models, PDPP uses a different experimental setting. In PDPP, authors set the window after the start time of a_1 and before the end time of a_T, contrary to the standard practice of setting a 2-second window around the start and end

Models	Required Annotations				$T=3$			$T=4$		
	step class	visual states	step text	task class	$SR^{\uparrow}$	$mAcc^{\uparrow}$	$mIoU^{\uparrow}$	$SR^{\uparrow}$	$mAcc^{\uparrow}$	$mIoU^{\uparrow}$
Random	✓				< 0.01	0.94	1.66	< 0.01	0.83	1.66
Retrieval-Based	✓				8.05	23.3	32.06	3.95	22.22	36.97
WLTDO [16]	✓	✓			1.87	21.64	31.70	0.77	17.92	26.43
UAAA [1]	✓	✓			2.15	20.21	30.87	0.98	19.86	27.09
UPN [49]	✓	✓			2.89	24.39	31.56	1.19	21.59	27.85
DDN [9]	✓	✓			12.18	31.29	47.48	5.97	27.10	48.46
PlaTe [50]	✓	✓			16.00	36.17	65.91	14.00	35.29	55.36
Ext-GAIL wo Aug. [7]	✓	✓			18.01	43.86	57.16	-	-	-
Ext-GAIL [7]	✓	✓			21.27	49.46	61.70	16.41	43.05	60.93
P³IV ♣ [59]	✓		✓		23.34	49.96	73.89	13.40	44.16	70.01
PDPP ♣ [54]	✓			✓	26.38	55.62	59.34	18.69	52.44	62.38
E3P ♣ [53]	✓		✓	✓	26.40	53.02	74.05	16.49	48.00	70.16
SkipPlan [29] ♣	✓				28.85	61.18	**74.98**	15.56	55.64	**70.30**
Ours w/ P²KG (R=2)	✓				22.60	48.76	53.57	13.90	45.79	55.00
Ours ♣ w/ P²KG (R=1)	✓				33.34	**61.36**	64.14	20.38	55.54	64.03
Ours ♣ w/ P²KG (R=2)	✓				**33.38**	60.79	63.89	**21.02**	**56.08**	64.15
PDPP ♣ † [54]	✓			✓	37.20	64.67	66.57	21.48	57.82	65.13
Ours ♣ † w/ P²KG (R=1)	✓				**38.12**	**64.74**	**67.15**	**24.15**	**59.05**	**66.64**

Table 1. Performance of our method in comparison to existing baselines for CrossTask dataset. ♣ means that the input visual features are from the S3D network [35] pretrained on HowTo100M [34]; otherwise, precomputed features provided in CrossTask are used. † indicates the results are under the PDPP's task setting, while others are under the conventional setting

Models	$T=5$	$T=6$
DDN [9]	3.10	1.20
P³IV ♣ [59]	7.21	4.40
PDPP ♣ [54]	13.22	7.49
E3P ♣ [53]	8.96	5.76
SkipPlan ♣ [29]	8.55	5.12
Ours (R=2)	8.17	5.32
Ours ♣ (R=1)	**13.25**	8.09
Ours ♣ (R=2)	12.74	**9.23**
PDPP ♣ † [54]	13.45	8.41
Ours ♣ † (R=1)	**14.20**	**9.27**

Table 2. Success Rate ($SR^{\uparrow}$) comparison to existing baselines for CrossTask dataset under longer horizons

time (*ref.* [9]). We conduct experiments on both PDPP's proposed setting and the conventional setting.

Inference: During the inference phase, the model receives only the start observation v_s and the goal observation v_g. To proceed, it utilizes a step model to predict the initial action a_1 and the end action a_T for each data. Subsequently, leveraging the P²KG, highest probable procedure knowledge graph plans connecting a_1 and a_T are obtained. Then, a multi-dimensional array, is created as mentioned in Eq. 5. Finally, the planning model is used to predict the sequence of actions $[a_1, ..., a_T]$ by denoising the generated multi-dimensional array as in § 3.2.3.

4.1. Comparison with the State of the Art (SOTA)

CrossTask (short horizon): We evaluate on CrossTask for short horizons ($T = 3$ and $T = 4$). According to the results shown in Table 1, our proposed method outperforms the PDPP in PDPP's setting in every evaluation metric. More than 0.9% and 2% improvement in success rate in $T = 3$ and $T = 4$ respectively. In the conventional setting, our method with both P²KG (R=1) and P²KG (R=2) conditions outperform the success rate values by a significant margin compared to other baselines. P²KG (R=2) slightly outperforms P²KG (R=1), indicating potential benefits of incorporating more procedural knowledge from the P²KG.

CrossTask (long horizon): We use long-horizon predictions for $T = 5$ and $T = 6$ for further evaluating our model as shown in Table 2. In PDPP's setting (†), our method improves the success rate in both $T = 5$ and $T = 6$. In the conventional setting, our method utilizing P²KG (R=1) demonstrates the highest SR value for $T = 5$, and for a longer horizon at $T = 6$, our method delivers superior performance for P²KG (R=2). Our method performs well under the challenging scenario of a long planning horizon. Our success rate (SR) diminishes from approximately 40% to 10% when extending the planning horizon from T=3 to T=6, primarily due to the heightened uncertainty surrounding the predicted plan between the initial and final steps. This uncertainty stems from the increase in the number of potential procedural plans within the P²KG.

NIV and COIN: Results are shown in Table 3 and Table 4. On NIV, ours achieves the best result under the mIoU metric with T=3, and under both the SR and mIoU metrics with T=4. The results on NIV (T=5, T=6) are available in the supplementary material. For the COIN dataset we only report SR and mAcc due to space constraints; mIoU is reported in the supplementary material. Our method does not rank as the top performer on COIN when T=3 or T=4. The likely reason is that the COIN dataset features just an average of 3.9 actions per video–a scenario that demands only short-horizon planning and does not necessitate *advanced* procedural knowledge (which encompasses long sequence-level knowledge [62]). Furthermore, the dataset's extensive collection of over 11k videos provides a substantial resource for baselines to learn *basic* procedural knowledge.

Models	NIV (T=3)			NIV (T=4)		
	$SR^{\uparrow}$	$mAcc^{\uparrow}$	$mIoU^{\uparrow}$	$SR^{\uparrow}$	$mAcc^{\uparrow}$	$mIoU^{\uparrow}$
Random	2.21	4.07	6.09	1.12	2.73	5.84
DDN [9]	18.41	32.54	56.56	15.97	27.09	53.84
Ext-GAIL [7]	22.11	42.20	65.93	19.91	36.31	53.84
P^{3}IV [59]	24.68	49.01	74.29	20.14	38.36	67.29
E3P [53]	**26.05**	**51.24**	75.81	21.37	**41.96**	74.90
PDPP [54]	22.22	39.50	86.66	21.30	39.24	84.96
Ours	24.44	43.46	**86.67**	**22.71**	41.59	**91.49**

Table 3. Performance of baselines and ours for NIV dataset

Models	COIN (T=3)		COIN (T=4)		COIN (T=5)	
	$SR^{\uparrow}$	$mAcc^{\uparrow}$	$SR^{\uparrow}$	$mAcc^{\uparrow}$	$SR^{\uparrow}$	$mAcc^{\uparrow}$
Random	< 0.01	< 0.01	< 0.01	< 0.01	-	-
Retrieval	4.38	17.40	2.71	14.29	-	-
DDN [9]	13.90	20.19	11.13	17.71	-	-
P^{3}IV [59]	15.40	21.67	11.32	18.85	4.27	10.81
E3P [53]	19.57	31.42	13.59	26.72	-	-
PDPP [54]	19.42	43.44	13.67	42.58	13.02	**43.36**
SkipPlan [29]	**23.65**	**47.12**	**16.04**	**43.19**	9.90	38.99
Ours (R=2)	20.25	39.87	15.63	39.53	**16.06**	40.72

Table 4. Performance of baselines and ours for COIN dataset

4.2. Ablation Studies and Analyses

Ablation on the probabilistic procedure knowledge graph. We analyze the role of P^{2}KG in improving the performance of our proposed method. Table 5 shows the results which clearly demonstrate that using P^{2}KG conditions improves the performance significantly for every T value. Especially when $T = 4$, success rate (SR) improves more than 3% and mean IoU improves more than 2%.

Plan recommendations provided by probabilistic procedure knowledge graph v.s. LLM. We recognize the recent trend of utilizing LLMs to enhance action anticipation [60] or planning in other realms [3, 27, 28, 40, 46]. In Table 6, we compare the results between using P^{2}KG v.s. using LLM ('llama-2-13b-chat' and 'llama-2-70b-chat') to generate the plan recommendations. When examining Table 6, it becomes apparent that there are trade-offs between using LLM-generated recommendations and P^{2}KG recommendations. For instance, P^{2}KG recommendations are constrained by the data available in the training set, limiting their applicability to unseen procedural activities. On the other hand, LLMs tend to exhibit better generalization to such unseen activities. However, considering that the training and testing are conducted on the aforementioned three datasets with known activities, P^{2}KG recommendations can yield more accurate results compared to relying on LLM-generated recommendations.

Probabilistic procedure knowledge graph (P^{2}KG) v.s. Frequency-based procedure knowledge graph (PKG). The probabilistic procedure knowledge graph uses out-edge normalization to encode step transition probabilities (§ 3.2.2), while the frequency-based procedure knowledge graph uses min-max normalization over the frequency counts throughout the graph. In both cases, the planning model only uses one procedure plan recommendation from the graph as condition in our experimental analysis. By looking at the results shown in Table 7, it is evident that the probabilistic procedure knowledge graph outperforms the frequency-based procedure knowledge graph.

Effect of utilizing *predicted* steps for input conditions to train the procedure planing model. Our proposed problem decomposition allows training the planning model with ground truth (GT) first and last steps. We experiment with two ways to train the planning model. Method 1 uses the predicted start and end steps ($\hat{a}_1$ and $\hat{a}_T$) as input to generate P^{2}KG conditions and use them to train the planning model. Method 2 is where we augment the predicted start and end steps using the GT start and end steps (a_1 and a_T) by generating 3 more data samples as follows: [$\hat{a}_1$,a_T], [a_1,$\hat{a}_T$], and [a_1,a_T]. Then we generate P^{2}KG conditions for each data and train the model. From the results shown in Table 8, the method without GT data augmentation shows better results. This suggests that leveraging ground truth data in training can lead to worse performance in testing.

Qualitative results. Figures 3 and 4 provide qualitative examples of our method. Intermediate steps are padded in the step model because it only predicts the start and end actions. In the 'make jello shot' task (see Figure 3), the model gives a wrong prediction in the intermediate steps when using P^{2}KG (R=1) condition. However, it predicts correctly when using P^{2}KG (R=2) conditions. In the 'change a tire' task shown in Figure 4, the model is able to predict all the intermediate steps in given conditions.

Visualizations of the probabilistic procedure knowledge graph. We show a sub-graph from our probabilistic procedure knowledge graph (Figure 5). This graph is drawn around the 'jack up' node up to the depth of 2 nodes.

Visualizations of the expert trajectories. Figure 6 illustrates the steps involved in completing the 'make jello shots' task, along with their transitions to other steps within the entire training data. This figure demonstrates that our P^{2}KG encodes diverse sequencing possibilities for steps and also captures task-sharing steps across the entire training domain. For instance, 'pour water' is a step in 'make jello

Model	T=3			T=4			T=5			T=6		
	SR	mAcc	mIoU	SR	mAcc	mIoU	SR	mAcc	mIoU	SR	mAcc	mIoU
w.o P^{2}KG conditions †	35.69	63.91	66.04	20.52	57.47	64.39	12.8	53.44	64.01	8.15	50.45	64.13
Ours †	38.12	64.74	67.15	24.15	59.05	66.64	14.20	53.84	65.56	9.27	50.22	65.97
w.o P^{2}KG conditions	31.35	59.51	63.11	18.92	56.20	62.47	12.71	51.29	63.56	8.16	47.63	63.39
Ours	33.38	60.79	63.89	21.02	56.08	64.15	12.74	51.23	63.16	9.23	50.78	65.56

Table 5. Performance of our method with and without P^{2}KG conditions on CrossTask ♣ dataset

Model (T=6, CrossTask ♣)	SR	mAcc	mIoU
Ours with P^{2}KG (R=1)			
PDPP setting	**9.27**	50.22	**65.97**
Conventional setting	8.09	50.80	**65.39**
One LLM plan recommendation			
PDPP setting (13b)	7.74	50.28	64.05
Conventional setting (13b)	7.21	49.68	63.89
PDPP setting (70b)	8.62	**50.31**	64.34
Conventional setting (70b)	7.81	49.75	64.02
P^{2}KG (R=1) and one LLM plan recommendation			
PDPP setting (13b)	8.81	49.97	65.22
Conventional setting (13b)	8.20	51.46	64.30
PDPP setting (70b)	9.01	50.25	65.57
Conventional setting (70b)	**8.34**	**51.53**	64.96

Table 6. Performance of the plan recommendations provided by the probabilistic procedure knowledge graph v.s. LLM.

Models	SR	mAcc	mIoU
Frequency graph	7.66	48.61	64.21
Probabilistic graph	**8.09**	**50.80**	**65.40**

Table 7. Performance comparison between probabilistic procedure knowledge graph v.s. frequency-based procedure knowledge graph for T=6 on CrossTask ♣ dataset

Condition	SR	mAcc	mIoU
without GT data aug.	38.12	64.74	67.15
with GT data aug.	32.45	62.42	62.80

Table 8. Effect of different input conditions for performance on CrossTask ♣ dataset (T=3) in PDPP's setting

Models	T=3		T=4		T=5		T=6	
	$\hat{a}_1$	$\hat{a}_T$	$\hat{a}_1$	$\hat{a}_T$	$\hat{a}_1$	$\hat{a}_T$	$\hat{a}_1$	$\hat{a}_T$
Ours	53.69	50.60	55.56	52.51	55.58	51.81	57.09	51.92
Ours ♣	71.42	63.32	72.98	63.37	72.42	63.29	63.82	59.96

Table 9. The step model's start and end step prediction accuracies on the CrossTask dataset

shots' task, but it can also be part of other tasks, leading to a step transition from 'pour water' to 'add fish.' This structure allows models to leverage rich procedural knowledge.

Results for the step model. Table 9 reveals step model results, indicating potential enhancement areas to elevate the planning performance.

Limitations & Failure cases. Our model exhibits three distinct failure case patterns. See section B.5 of the supplementary material for detailed discussions.

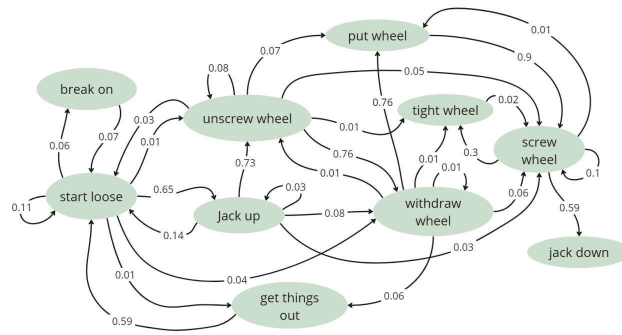

Figure 5. Example of a sub-graph in our probabilistic procedure knowledge graph (P^{2}KG) for CrossTask dataset. This graph effectively encapsulates real-world knowledge of distinct transition probabilities between steps, e.g., the probability of transitioning from 'start loose' to 'jack up' is 0.65, in contrast to a mere 0.14 for the reverse transition–the P^{2}KG reflects the common real-life practice where loosening the lug nuts before jacking up the car leads to a safer and more efficient tire change.

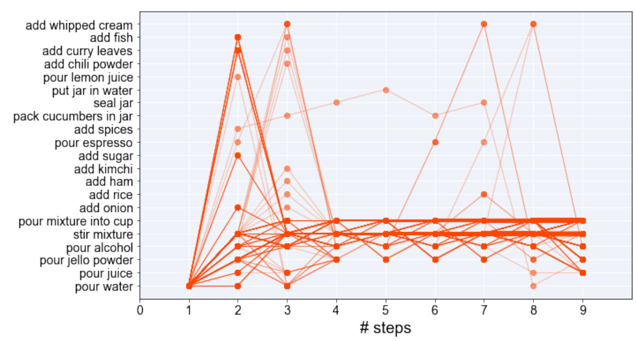

Figure 6. Expert trajectories of the 'Make Jello Shots' task, involving task-sharing steps and thus out-of-task step transitions. Thicker lines indicate paths that are more frequently visited

5. Conclusion

We focus on formulating procedural plans from an AI agent in instructional videos. We propose KEPP which employs a probabilistic procedural knowledge graph, sourced from the training domain, effectively serving as a 'textbook' for procedure planning. Results show that KEPP delivers SOTA performance with minimal supervision. Future work can focus on enhancing the accuracy of predictions for the initial and final steps. Additionally, our approach can be modified to aid in detecting erroneous steps and the misordering of steps in instructional videos [38, 41].

References

[1] Yazan Abu Farha and Juergen Gall. Uncertainty-aware anticipation of activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 5, 6

[2] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 2

[3] Anurag Ajay, Seungwook Han, Yilun Du, Shaung Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *arXiv preprint arXiv:2309.08587*, 2023. 7

[4] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 5

[5] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. *arXiv preprint arXiv:2307.08763*, 2023. 2, 4, 5

[6] Anonymous authors. Active procedure planning with uncertainty-awareness in instructional videos, 2023. Under review as a conference paper at ICLR 2024. https://openreview.net/pdf?id=JDd46WodYf. 2

[7] Jing Bi, Jiebo Luo, and Chenliang Xu. Procedure planning in instructional videos via contextual modeling and model-based policy learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15611–15620, 2021. 1, 2, 5, 6, 7

[8] Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D Hwang, Xiang Lorraine Li, Hirona J Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. Plasma: Making small language models better procedural knowledge models for (counterfactual) planning. *arXiv preprint arXiv:2305.19472*, 2023. 2

[9] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *European Conference on Computer Vision*, pages 334–350. Springer, 2020. 1, 2, 3, 5, 6, 7

[10] Brian Chen, Nina Shvetsova, Andrew Rouditchenko, Daniel Kondermann, Samuel Thomas, Shih-Fu Chang, Rogerio Feris, James Glass, and Hilde Kuehne. What, when, and where?–self-supervised spatio-temporal grounding in untrimmed multi-action videos from narrated instructions. *arXiv preprint arXiv:2303.16990*, 2023. 2

[11] Sixun Dong, Huazhang Hu, Dongze Lian, Weixin Luo, Yicheng Qian, and Shenghua Gao. Weakly supervised video representation learning with unaligned text for sequential videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2447, 2023. 2

[12] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 868–878, 2020. 2

[13] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023. 1, 2

[14] Nikita Dvornik, Isma Hadji, Hai Pham, Dhaivat Bhatt, Brais Martinez, Afsaneh Fazly, and Allan D Jepson. Flow graph to video grounding for weakly-supervised multi-step localization. In *European Conference on Computer Vision*, pages 319–335. Springer, 2022. 2

[15] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18952–18961, 2023. 2

[16] Kiana Ehsani, Hessam Bagherinezhad, Joseph Redmon, Roozbeh Mottaghi, and Ali Farhadi. Who let the dogs out? modeling dog behavior from visual data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4051–4060, 2018. 5, 6

[17] Sophie Fischer, Carlos Gemmell, Iain Mackie, and Jeffrey Dalton. Vilt: Video instructions linking for complex tasks. In *Proceedings of the 2nd International Workshop on Interactive Multimedia Retrieval*, pages 41–47, 2022. 2

[18] Kevin Flanagan, Dima Damen, and Michael Wray. Learning temporal sentence grounding from narrated egovideos. *arXiv preprint arXiv:2310.17395*, 2023. 2

[19] Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to segment actions from observation and narration. *arXiv preprint arXiv:2005.03684*, 2020. 2

[20] Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel LK Yamins, James J DiCarlo, Josh McDermott, Antonio Torralba, et al. The threedworld transport challenge: A visually guided task-and-motion planning benchmark for physically realistic embodied ai. *arXiv preprint arXiv:2103.14025*, 2021. 2

[21] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, and Behzad Dariush. Weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10128–10138, 2023. 2

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 4

[23] Guyue Hu, Bin He, and Hanwang Zhang. Compositional prompting video-language models to understand procedure in instructional videos. *Machine Intelligence Research*, 20 (2):249–262, 2023. 2

[24] De-An Huang, Joseph J Lim, Li Fei-Fei, and Juan Carlos Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2183–2192, 2017. 2

[25] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding" it": Weakly-supervised reference-aware visual grounding in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5948–5957, 2018. 2

[26] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. *arXiv preprint arXiv:2011.11760*, 2020. 2

[27] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022. 2, 7

[28] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022. 2, 7

[29] Zhiheng Li, Wenjia Geng, Muheng Li, Lei Chen, Yansong Tang, Jiwen Lu, and Jie Zhou. Skip-plan: Procedure planning in instructional videos via condensed action space learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10297–10306, 2023. 2, 3, 5, 6, 7

[30] Yujie Lu, Weixi Feng, Wanrong Zhu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. Neuro-symbolic procedural planning with commonsense prompting. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[31] Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Eric Wang, and William Yang Wang. Multimodal procedural planning via dual text-image prompting. *arXiv preprint arXiv:2305.01795*, 2023. 2

[32] Weichao Mao, Ruta Desai, Michael Louis Iuzzolino, and Nitin Kamra. Action dynamics task graphs for learning plannable representations of procedural tasks. *arXiv preprint arXiv:2302.05330*, 2023. 2

[33] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. Learning to ground instructional articles in videos through narrations. *arXiv preprint arXiv:2306.03802*, 2023. 2

[34] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 1, 6

[35] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 2, 6

[36] Davide Moltisanti, Frank Keller, Hakan Bilen, and Laura Sevilla-Lara. Learning action changes by measuring verb-adverb textual relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23110–23118, 2023. 2

[37] Medhini Narasimhan, Arsha Nagrani, Chen Sun, Michael Rubinstein, Trevor Darrell, Anna Rohrbach, and Cordelia Schmid. Tl; dw? summarizing instructional videos with task relevance and cross-modal saliency. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 2

[38] Medhini Narasimhan, Licheng Yu, Sean Bell, Ning Zhang, and Trevor Darrell. Learning and verification of task structure in instructional videos. *arXiv preprint arXiv:2303.13519*, 2023. 8

[39] Megha Nawhal, Akash Abdu Jyothi, and Greg Mori. Rethinking learning approaches for long-term action anticipation. In *European Conference on Computer Vision*, pages 558–576. Springer, 2022. 2

[40] Dhruvesh Patel, Hamid Eghbalzadeh, Nitin Kamra, Michael Louis Iuzzolino, Unnat Jain, and Ruta Desai. Pretrained language models as visual planners for human assistance. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15302–15314, 2023. 7

[41] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 8

[42] Fadime Sener, Rishabh Saraf, and Angela Yao. Transferring knowledge from text to video: Zero-shot anticipation for procedural actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[43] Anshul Shah, Benjamin Lundell, Harpreet Sawhney, and Rama Chellappa. Steps: Self-supervised key step extraction from unlabeled procedural videos. *arXiv preprint arXiv:2301.00794*, 2023. 2

[44] Yuhan Shen, Lu Wang, and Ehsan Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2021. 2

[45] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 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, 2020. 2

[46] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023. 2, 7

[47] Yale Song, Gene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 2

[48] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Look for the change: Learning object states and state-modifying actions from untrimmed

web videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13956–13966, 2022. 2

[49] Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, pages 4732–4741. PMLR, 2018. 5, 6

[50] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters*, 7(2):4924–4930, 2022. 1, 2, 5, 6

[51] Reuben Tan, Bryan Plummer, Kate Saenko, Hailin Jin, and Bryan Russell. Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. *Advances in Neural Information Processing Systems*, 34:14476–14487, 2021. 2

[52] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 5

[53] An-Lan Wang, Kun-Yu Lin, Jia-Run Du, Jingke Meng, and Wei-Shi Zheng. Event-guided procedure planning from instructional videos with text supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13565–13575, 2023. 1, 2, 3, 5, 6, 7

[54] Hanlin Wang, Yilu Wu, Sheng Guo, and Limin Wang. Pdpp: Projected diffusion for procedure planning in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14836–14845, 2023. 1, 2, 3, 4, 5, 6, 7

[55] Frank F Xu, Lei Ji, Botian Shi, Junyi Du, Graham Neubig, Yonatan Bisk, and Nan Duan. A benchmark for structured procedural knowledge extraction from cooking videos. *arXiv preprint arXiv:2005.00706*, 2020. 2

[56] Yue Yang, Joongwon Kim, Artemis Panagopoulou, Mark Yatskar, and Chris Callison-Burch. Induce, edit, retrieve: Language grounded multimodal schema for instructional video retrieval. *arXiv preprint arXiv:2111.09276*, 2021. 2

[57] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23056–23065, 2023. 2

[58] Jiahao Zhang, Anoop Cherian, Yanbin Liu, Yizhak Ben-Shabat, Cristian Rodriguez, and Stephen Gould. Aligning step-by-step instructional diagrams to video demonstrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2483–2492, 2023. 2

[59] He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. P3iv: Probabilistic procedure planning from instructional videos with weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2938–2948, 2022. 1, 2, 3, 5, 6, 7

[60] Qi Zhao, Ce Zhang, Shijie Wang, Changcheng Fu, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Antgpt: Can large language models help long-term action anticipation from videos? *arXiv preprint arXiv:2307.16368*, 2023. 7

[61] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14825–14835, 2023. 2

[62] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10727–10738, 2023. 2, 5, 7

[63] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. 1, 2, 5