Procedure-Aware Pretraining for Instructional Video Understanding

Honglu Zhou[1,2], Roberto Martín-Martín[1,3], Mubbasir Kapadia[2], Silvio Savarese[1] and Juan Carlos Niebles[1]
[1]Salesforce Research, [2]Rutgers University, [3]UT Austin
{hz289,mk1353}@cs.rutgers.edu, robertomm@cs.utexas.edu, {ssavarese,jniebles}@salesforce.com

Abstract

*Our goal is to learn a video representation that is useful for downstream procedure understanding tasks in instructional videos. Due to the small amount of available annotations, a key challenge in procedure understanding is to be able to extract from unlabeled videos the procedural knowledge such as the identity of the task (e.g., 'make latte'), its steps (e.g., 'pour milk'), or the potential next steps given partial progress in its execution. Our main insight is that instructional videos depict sequences of steps that repeat between instances of the same or different tasks, and that this structure can be well represented by a Procedural Knowledge Graph (`PKG`), where nodes are discrete steps and edges connect steps that occur sequentially in the instructional activities. This graph can then be used to generate pseudo labels to train a video representation that encodes the procedural knowledge in a more accessible form to generalize to multiple procedure understanding tasks. We build a `PKG` by combining information from a text-based procedural knowledge database and an unlabeled instructional video corpus and then use it to generate training pseudo labels with four novel pre-training objectives. We call this `PKG`-based pre-training procedure and the resulting model `Paprika`, **P**rocedure-**A**ware **PR**e-training for **I**nstructional **K**nowledge **A**cquisition. We evaluate `Paprika` on COIN and CrossTask for procedure understanding tasks such as task recognition, step recognition, and step forecasting. `Paprika` yields a video representation that improves over the state of the art: up to* **11.23**% *gains in accuracy in* 12 *evaluation settings. Implementation is available at* https://github.com/salesforce/paprika.

1. Introduction

Instructional videos depict humans demonstrating how to perform multi-step tasks such as cooking, making up and embroidering, repairing, or creating new objects. For a holistic instructional video understanding, an agent has to acquire *procedural knowledge*: structural information about tasks such as the identification of the task, its steps, or forecasting the next steps. An agent that has acquired procedural knowledge is said to have gained *procedure understanding* of instructional videos, which can be then exploited in multiple real-world applications such as instructional video labeling, video chapterization, process mining and, when connected to a robot, robot task planning.

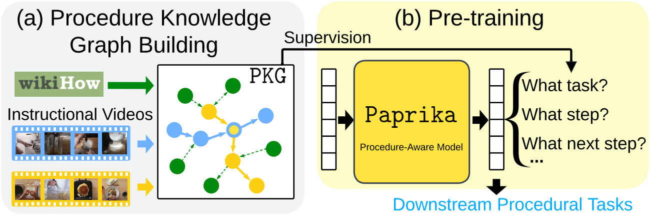


Figure 1. **Training a video representation for procedure understanding with supervision from a procedural knowledge graph**: the structure observed in instructions for procedures (from text, from videos) corresponds to sequences of steps that repeat between instances of the same or different tasks; this structure is well represented by a Procedural Knowledge Graph (`PKG`). (a) We build a `PKG` combining text instructions with unlabeled video data, and (b) obtain a video representation by encoding the human procedural knowledge from the `PKG` into a more general procedure-aware model (`Paprika`) generating pseudo labels with the `PKG` for several procedure understanding objectives. `Paprika` can then be easily applied to multiple downstream procedural tasks.

Our goal is to learn a novel video representation that can be applicable to a variety of procedure understanding tasks in instructional videos. Unfortunately, prior methods for video representation learning are inadequate for this goal, as they lack the ability to capture procedural knowledge. This is because most of them are trained to learn the (weak) correspondence between visual and text modalities, where the text comes either from automatic-speech recognition (ASR) on the audio [43, 77], which is noisy and error-prone, or from a caption-like descriptive sentence (e.g., "a video of a dog") [33], which does not contain sufficient information for fine-grained procedure understanding tasks such as step recognition or anticipation. Others are pre-trained on masked frame modeling [34], frame order modeling [34] or video-audio matching [1], which gives them basic video spatial, temporal or multimodal understanding but is too generic for procedure understanding tasks.

Closer to our goal, Lin et al. [38] propose a video foundation model for procedure understanding of instructional videos by matching the videos' ASR transcription (i.e., subtitle/narration) to procedural steps from a text procedural knowledge database (wikiHow [30]) and training the video-representation-learning model to match each part of an instructional video to the corresponding step. Their method only acquires isolated step knowledge in pre-training and is not as suitable to gain sophisticated procedural knowledge.

We propose Paprika, from **P**rocedure-**A**ware **PR**e-training for **I**nstructional **K**nowledge **A**cquisition, a method to learn a novel video representation that encodes procedural knowledge (Fig. 1). Our main insight is that the structure observed in instructional videos corresponds to sequences of steps that repeat between instances of the same or different tasks. This structure can be captured by a Procedural Knowledge Graph (PKG) where nodes are discretized steps annotated with features, and edges connect steps that occur sequentially in the instructional activities. We build such a graph by combining the text and step information from wikiHow and the visual and step information from unlabeled instructional video datasets such as HowTo100M [45] automatically. The resulting graph encodes procedural knowledge about tasks and steps, and about the temporal order and relation information of steps.

We then train our Paprika model on multiple pre-training objectives using the PKG to obtain the training labels. The proposed four pre-training objectives (Sec. 3.3) respectively focuses on procedural knowledge about the step of a video, tasks that a step may belong to, steps that a task would require, and the general order of steps. These pre-training objectives are designed to allow a model to answer questions about the subgraph of the PKG that a video segment may belong to. The PKG produces pseudo labels for these questions as supervisory signals to *adapt* video representations produced by a video foundation model [9] for robust and generalizable procedure understanding.

Our contributions are summarized as follows:
(i) We propose a Procedural Knowledge Graph (PKG) that encodes human procedural knowledge from collectively leveraging a text procedural knowledge database (wikiHow) and an unlabeled instructional video corpus (HowTo100M).
(ii) We propose to elicit the knowledge in the PKG into Paprika, a procedure-aware model, using four pre-training objectives. To that end, we produce pseudo lables with the PKG that serve as supervisory signals to train Paprika to learn to answer multiple questions about the subgraph of the PKG that a video segment may belong to.
(iii) We evaluate our method on the challenging COIN and CrossTask datasets on downstream procedure understanding tasks: task recognition, step recognition, and step forecasting. Regardless of the capacity of the downstream model (from simple MLP to the powerful Transformer), our method yields a representation that outperforms the state of the art – up to $\mathbf{11.23\%}$ gains in accuracy out of 12 evaluation settings.

2. Related Work

We focus on learning a novel video representation that can be easily adapted to downstream instructional video procedure understanding tasks [35, 38] such as procedural task recognition [20], step recognition [27, 47, 88], anticipation [21, 36, 44, 53, 80], localization [13, 15, 65, 85] or segmentation [19, 25, 40, 41, 55, 56, 68, 86], procedure planing [8, 10, 62], and so on [2, 12, 14, 23, 26, 58, 72, 73, 75, 78].

Our goal is related to self-supervised learning of video representations [51, 52]. *Self-supervised* pre-training objectives include predicting the video pace [6, 70], future frame [39, 59, 67], future ASR [54] or the context [50], motion and appearance statistics [69], solving space or/and time jigsaw puzzles [28, 31, 34, 74, 82], and identifying the odd video segment [16] – all exploiting the temporal signals. Masking has also gained popularity where signals of one or multiple modalities (frame/text/audio) are masked and required to be predicted [24, 32–34, 42, 61, 76, 81, 82]. Other pre-training objectives are based on spatiotemporal data augmentation [48, 81], cross-modality clustering [11] matching [1, 4, 5, 33, 37, 42, 43, 46, 77, 81, 82], as well as fine-grained noun/object-level or verb-level objectives [17, 33].

DS [38], MIL-NCE [43], VATT [1], VideoCLIP [77], VLM [76], MCN [11], MMV [4], Hero [34] and CBT [60] have utilized HowTo100M – a large-scale instructional video dataset [45] for pre-training. Except DS, they have utilized strictly or weakly temporally overlapped ASR with video frames as the source for contrastive learning or masked based modeling. DS [38] argued that ASR is a suboptimal source to describe procedural videos. They utilized a pre-trained language foundation model to match step headlines in wikiHow [30, 79, 83, 84, 87] to ASR sentences of video segments. The matched step headlines were then used to replace ASR sentences to learn a video representation learning model. On procedure understanding downstream tasks, DS outperforms models including S3D pre-trained with MIL-NCE [43] (MIL-NCE for short in the rest of the paper), ClipBERT [32] and VideoCLIP [77].

ActionCLIP [71] and Bridge-Prompt [35] are prompt-based models inspired by CLIP [49]; ActionCLIP focuses on atomic action recognition [18] (i.e., recognizing an atomic action such as "falling" from a short clip), whereas Bridge-Prompt is for ordinal action understanding related downstream applications. They are related to our work but both require action annotations for training. Instead, we focus on more effective pre-training methods for a procedure understanding model that encodes the procedural knowl-

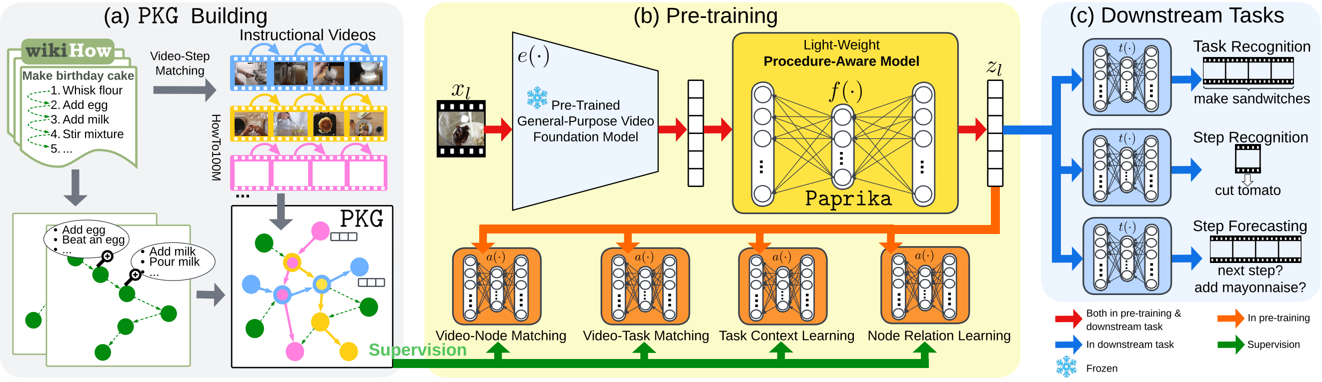


Figure 2. **Overview.** We encode procedural knowledge in a Procedural Knowledge Graph (PKG): nodes are (clustered) steps from wikiHow that are annotated with features, and edges connect steps that occur sequentially in the instructional activities from wikiHow or an unlabeled instructional video corpus. Four pre-training objectives elicit the knowledge in the PKG to Paprika, a procedure-aware model. We achieve this by querying the PKG to produce pseudo labels for pre-training as supervisory signals. Paprika learns a video representation that encodes procedural knowledge and thus lead to improved performance on multiple downstream procedure understanding tasks.

edge and avoids laborious annotations on step class and time boundary of instructional videos. This enables training on rich but unlabeled web data.

DS [38] leveraged wikiHow for instructional video understanding. This is similar to our goal of building the PKG from wikiHow and instructional videos. However, DS does not focus on encoding procedural knowledge during pre-training beyond video segment and text matching. For example, DS does not encode relationships between steps in the pre-trained video representation. This is in part due to multiple challenges that need to be addressed: ***(1)*** the order of steps to execute a task follows certain temporal or causal constraints, ***(2)*** the execution order of steps of the task in another video instance can be different from the order that is demonstrated in the current video instance, and ***(3)*** some steps may belong to tasks that are not demonstrated in the current video instance (i.e., the cross-task characteristics of steps). We propose the PKG to address these challenges. A model trained using our method can acquire the higher-level prior human procedural knowledge instead of just the *isolated* step knowledge that DS provides.

3. Methodology

3.1. Problem Formulation

Technically, video representation learning methods learn to represent a long video as a sequence of segment embeddings [1, 38, 43]. A video is viewed as a sequence of L segments $[x_1, \ldots, x_l, .., x_L]$, where $x_l \in \mathbb{R}^{H \times W \times 3 \times F}$, H and W denote the spatial resolution height and width, and F is #RGB frames of the video segment ("#" denotes "the number of"). A model is pre-trained to learn the mapping $x_l \rightarrow z_l \in \mathbb{R}^d$. Downstream models are applied on the (whole or partial) sequence of segment embeddings $[z_1, \ldots, z_l, .., z_L]$ to perform various tasks.

Our goal is to learn z_l that encodes procedural knowledge for downstream procedure understanding tasks for instructional videos. However, pre-training a new (or fine-tuning a pre-trained) video model becomes impractical for real-world settings as the model size grows rapidly [9, 51]. We propose instead a practical framework that trains a light-weight procedure-aware model $f(\cdot)$ that refines the video segment feature extracted from a *frozen* general-purpose video foundation model $e(\cdot)$, i.e., $z_l := f(e(x_l))$ (Fig. 2 (b)). Our framework exploits the success of existing large foundation models [9] and enables parameter-efficient transfer learning (similar practices used in [3, 33]). $f(\cdot)$ serves as a feature adapter [22, 63] to allow the refined video feature to encode the previously missing procedural knowledge for a stronger downstream procedure understanding capability. We coin our trained $f(\cdot)$ as Paprika.

Our key insight is that a text procedural knowledge database combined with unlabeled instructional videos can be utilized to build a Procedural Knowledge Graph (PKG) (Fig. 2 (a)) to encode procedural knowledge. The PKG can provide supervisory signals for training a *procedure-aware* model. We now describe how to build the PKG from wikiHow and unlabeled procedural videos (Sec. 3.2), and then introduce four pre-training objectives (Sec. 3.3) that allow Paprika to learn z_l infused with procedural knowledge by mining the PKG.

3.2. Procedural Knowledge Graph

The PKG is a homogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Nodes represent steps (e.g., 'add milk') from a wide variety of tasks (e.g., 'how to make latte'), and edges represent directed step transitions. That is, edge (i, j) indicates that a transition between steps in nodes i and j that was observed in real-life procedural data.
Step 1: Obtain nodes of the PKG. $\mathcal{V}$ contains steps of tasks that may appear in instructional videos. Pre-training uses *unlabeled* videos, thus, there are no step annotations

provided by the pre-training video corpus that can be directly used to form the discrete node entities. Inspired by DS [38], we resort to the step headlines in wikiHow [30].

wikiHow is a text-based procedural knowledge database $\mathbb{B}$ that contains articles describing the sequence of steps needed for the completion of a wide range of tasks. $\mathbb{B} = \big\{[s_1^{(1)}, \ldots, s_{b_1}^{(1)}], \ldots, [s_1^{(t)}, \ldots, s_{b_t}^{(t)}], \ldots, [s_1^{(T)}, \ldots, s_{b_T}^{(T)}]\big\}$ where T is #tasks, the subscript b_t is #steps of task t, and $s_i^{(t)}$ represents the natural language based summary (i.e., step headline) of i-th step for task t. Examples of wikiHow task articles are available in Fig. 4 and 5.

Since two step headlines in $\mathbb{B}$ can represent the same step but are described slightly differently, e.g., "jack up the car" and "jack the car up", we perform step deduplication by clustering similar step headlines. The resulting clusters are *step nodes* that constitute $\mathcal{V}$. We list the largest step nodes in Supplementary Material. We find cross-task characteristics of steps, i.e., one step may belong to multiple tasks.

`Step 2`: Add edges to the `PKG`. $\mathcal{E}$ is the set of direct transitions observed in data between any two step nodes. However, most tasks in $\mathcal{B}$ have only one article, which provides only one way to complete the task through a sequence of steps. How to encode the different ways to complete a task t that involve different execution order of steps or new steps that are absent in the article of task t, becomes a challenge.

Our solution is to additionally leverage an unlabeled instructional video corpus to provide more step transition observations. In practice, we use MIL-NCE [43], a pre-trained video-language model, to compute the matching score between a segment x_l and a step headline $s_i^{(t)}$. MIL-NCE was trained to learn video and text embeddings with high matching scores on co-occurring frames and ASR subtitles.

We then use a thresholding criterion and the correspondence between step *headlines* and step *nodes* obtained from `Step 1` to match step nodes to video segments and obtain step node transitions given the temporal order of segments in videos. The step node transitions from wikiHow or the video corpus constitute $\mathcal{E}$. Please refer to Supplementary Material for implementation details on graph construction.

$\mathcal{E}$ encodes the structure observed in instructional videos as it encompasses multiple sequences of steps that repeat between video instances of the same or different tasks. In addition, $\mathcal{E}$ captures the relations of steps; the type of relation is not strictly defined – the steps could be temporal or causal related – because the step transitions that form $\mathcal{E}$ are *observed* from human-provided real-life demonstrations.

`Step 3`: Populate graph attributes. It is possible to collect various forms of attributes for the `PKG` depending on the desired use cases of the `PKG`. For example, node attributes can be the step headline texts, task names associated with the step headlines of the node, video segments matched to the node, the distribution of timestamps of the matched segments, the aggregated multimodal features of the matched segments, and so on. Edge attributes can be the source of the step node transition (from wikiHow or the video corpus), task occurrence, distribution of timestamps of the transition, etc. We describe the graph attributes we used and how we used them in Sec. 3.3.

3.3. Training `Paprika`

The `PKG` is a rich source of supervision for training models for procedure understanding. We propose four pre-training objectives as exemplars to show the possible ways in which the `PKG` can provide supervisory signals to train $f(\cdot)$ to learn good video representations using unlabeled instructional videos.

Video-Node Matching (VNM) aims at answering: what are the step nodes of the `PKG` that are likely to be matched to the input video segment x_l? This pre-training objective leverages the *node identity* information of the `PKG`, and it resembles the downstream application of independently recognizing steps of video segments. Formally,

$$a(f(e(x_l))) \rightarrow \mathcal{V}_{\text{VNM}} \tag{1}$$

where $a(\cdot)$ denotes the answer head model that performs the pre-training objective given the refined video segment feature z_l produced by $f(\cdot)$ as input, and $\mathcal{V}_{\text{VNM}} \subseteq \mathcal{V}$.

Video-Task Matching (VTM) aims at answering: what are the tasks of the matched step nodes of the input video segment x_l? This pre-training objective leverages the node's task attribute in the `PKG`. VTM focuses on inferring the cross-task knowledge of the step nodes without the video context. Formally,

$$a(f(e(x_l))) \rightarrow \mathcal{T}_{\text{VTM}} \tag{2}$$

where $\mathcal{T}_{\text{VTM}} \subseteq \mathcal{T}$, and $\mathcal{T}$ is the set of tasks ($\|\mathcal{T}\| := T$). Since HowTo100M provides task names of the long videos, we experiment with using the task names from wikiHow and/or from HowTo100M (Sec. 4.4). When task names from both sources are used, VNM leads to 2 answer heads.

Task Context Learning (TCL) aims at answering: for tasks the input video segment may belong to (produced by VTM), what are the step nodes that the tasks would typically need? TCL also leverages the node's task attribute in the `PKG`, but it focuses on inferring step nodes that may co-occur with the matched step node of the video segment in demonstrations. TCL learns the task's step context that is commonly observed in data, without the context of the current video segment. Formally,

$$a(f(e(x_l))) \rightarrow \mathcal{V}_{\text{TCL}} \tag{3}$$

where $\mathcal{V}_{\text{TCL}} \subseteq \mathcal{V}$. When task names from both wikiHow and HowTo100M are used, TCL leads to 2 answer heads.

Node Relation Learning (NRL) aims at answering: what are the k-hop in-neighbors and out-neighbors of the matched step nodes of the input video segment x_l? k ranges from 1 to a pre-defined integer K, and thus NRL leads to $2K$ sub-questions ($2K$ answer heads). NRL leverages the edge information of the PKG, and it focuses on learning the local multi-scale graph structure of the matched nodes of x_l. Predicting the in-neighbors resembles predicting the historical steps, whereas predicting the out-neighbors resembles forecasting the next steps of x_l. Note that the answer to NRL can be steps that come from other tasks different from the task of the current video. Formally,

$$a(f(e(x_l))) \rightarrow \mathcal{V}_{\text{NRL}} \tag{4}$$

where $\mathcal{V}_{\text{NRL}} \subseteq \mathcal{V}$.

4. Experiments

4.1. Pre-training Dataset

HowTo100M [45] is a large-scale video dataset that contains over 1M long instructional videos (videos can be over 30 minutes) and is commonly used for video model pre-training. Videos were collected from YouTube using wikiHow article titles as search keywords [45]. To reduce the computational cost, most of our experiments, including the construction of the PKG, only use the HowTo100M subset of size 85K videos from [7].

4.2. Evaluation Settings

We study the transfer learning ability of our Paprika model trained using the PKG on **12** evaluation settings: 3 downstream tasks ×2 downstream datasets ×2 downstream models. The output of the trained procedure-aware model $f(\cdot)$ is the input to the downstream model $t(\cdot)$. Note that the PKG is only used for pre-training and it is discarded at the downstream evaluation time (test time for $f(\cdot)$).

4.2.1 Downstream Procedure Understanding Tasks

Long-Term Activity/Task Recognition (TR) aims to classify the activity/task given all segments from a video.
Step Recognition (SR) recognizes the step class given as input the segments of a step in a video.
Future Step Forecasting (SF) predicts the class of the next step given the past video segments. Such input contains the historical steps *before* the step to predict happens. As in [38], we set the history to contain at least one step.

4.2.2 Downstream Datasets

We use COIN [64, 65] and CrossTask [88] as the downstream datasets because the two cover a wide range of procedural tasks in human daily activities.
COIN contains 11K instructional videos covering 778 individual steps from 180 tasks in various domains. The average number of steps per video is 3.9.
CrossTask has 4.7K instructional videos annotated with task name for each video spanning 83 tasks with 105 unique steps. 2.7K videos have steps' class and temporal boundary annotations; these videos are used for the SR and SF tasks. 8 steps per video on average.

4.2.3 Downstream Task Models

Segment features from the trained *frozen* $f(\cdot)$ are the input to downstream task model $t(\cdot)$. $t(\cdot)$ is trained and evaluated on the smaller-scale downstream dataset to perform downstream tasks. We experiment with two options for $t(\cdot)$.
MLP. MLP with only 1 hidden layer is the classifier of the downstream tasks, given the input of mean aggregated sequence features. Since a shallow MLP has a limited capacity, performance of a MLP downstream task model heavily relies on the quality of the input segment features.
Transformer. Since context and temporal reasoning is crucial for the downstream TR and SF tasks, we follow [38] to use a one-layer Transformer [66] to allow the downstream task model the capability to automatically learn to reason about segment and step relations. Transformer is a relatively stronger downstream task model compared to MLP.

4.3. Implementation Details

We used the version of $\mathbb{B}$ that has $10,588$ step headlines from T=$1,053$ task articles. We used Agglomerative Clustering given the features of step headlines, which resulted in $10,038$ step nodes. Length of segments was set to be 9.6 seconds. The pre-trained MIL-NCE [43] was used as $e(\cdot)$. $f(\cdot)$ was a MLP with a bottleneck layer that has a dimension of 128 as the only hidden layer. The refined segment feature shares the same dimension as the input segment feature (i.e., 512). Our pre-training objectives were cast to a multi-label classification problem with Binary Cross Entropy as the loss function. We used the Adam optimizer [29], a batch size of 256, and 8 NVIDIA A100 GPUs. Interested readers may refer to Supplementary Material for more details.

4.4. Quantitative Results

4.4.1 Ablation Studies

We train Paprika utilizing each of our pre-training objectives from Sec. 3.3; the results are in Table 1. We also compute a performance matrix with color-based visualization (Fig. 3) to compare the overall performance of the different pre-training objectives more easily.

The performance ranking of the pre-training objectives is NRL > TCL > VTM > VNM. VNM is the least powerful because it only focus on learning the simpler knowledge of matching single video segments to step nodes.

VTM (w+h) > VTM (w) > VTM (h) where 'w' denotes wikiHow and 'h' for HowTo100M. This ranking suggests that if the pre-training video corpus has the annotation of video's task name, our method can well utilize such annotation to further improve performance. Utilizing the wikiHow task names is better than HowTo100M because the mapping between step headlines and HowTo100M tasks would *not* be as clean as the mapping between step headlines and wikiHow tasks, because the former depends on the quality of the matching between a video segment to a step headline.

Pre-training Method		**Downstream Transformer**						**Downstream MLP**					
		COIN			**CrossTask**			**COIN**			**CrossTask**		
		SF	**SR**	**TR**	**SF**	**SR**	**TR**	**SF**	**SR**	**TR**	**SF**	**SR**	**TR**
MIL-NCE* [43] ($e(\cdot)$)		36.55	41.98	76.62	57.96	59.90	61.71	3.16	1.17	21.06	27.71	24.98	5.27
DS [38]		38.13	42.54	79.94	56.29	57.11	59.49	32.54	34.07	72.65	49.95	50.23	57.28
DS* [38]		39.54	45.97	82.66	**61.23**	**61.91**	64.24	30.88	32.74	77.66	52.97	53.69	61.08
VSM		39.29	44.37	82.23	57.94	58.92	62.24	31.45	32.66	76.51	49.59	50.01	58.76
Paprika (ours)	**VNM**	41.98	49.80	82.88	59.45	61.00	64.77	37.56	42.32	82.23	57.08	58.23	64.14
	VTM (*wikiHow*)	42.05	49.89	84.45	60.27	61.26	66.25	38.13	42.56	82.41	58.48	59.02	65.82
	VTM (*HT100M*)	41.97	48.59	83.44	60.19	60.64	65.08	36.87	40.07	81.52	56.45	57.42	65.61
	VTM (*wikiHow* + *HT100M*)	42.10	50.02	**84.73**	60.63	61.14	66.14	38.12	42.68	82.77	58.87	59.30	66.14
	TCL (*wikiHow*)	42.42	50.12	84.48	60.27	61.40	**66.67**	39.04	44.16	82.84	58.48	59.59	65.93
	TCL (*HT100M*)	42.05	48.68	83.20	60.49	61.86	66.03	38.86	43.56	82.55	58.38	58.63	64.66
	TCL (*wikiHow* + *HT100M*)	42.53	49.79	83.95	60.19	61.67	66.14	38.61	43.27	82.95	58.40	59.26	65.08
	NRL (*1 hop*)	**42.60**	**50.23**	84.66	60.68	61.36	**66.67**	**39.58**	**45.38**	**83.45**	59.12	59.59	65.95
	NRL (*2 hops*)	42.53	50.13	84.31	60.68	61.60	**66.67**	**40.55**	**45.82**	**83.84**	**60.13**	**60.23**	**66.98**
	VNM + VTM + TCL + NRL	**42.65**	**50.48**	**85.31**	**61.42**	**62.38**	**67.09**	**39.82**	**44.78**	**83.88**	**59.53**	**60.16**	**67.41**
	Gains to DS [38]	+4.52	+7.94	+5.37	+5.13	+5.27	+7.60	+7.28	+10.71	+11.23	+9.58	+9.93	+10.13
Paprika (ours)*	**VNM + VTM + TCL + NRL**	**43.22**	**50.99**	**85.84**	**62.63**	**63.53**	**68.35**	38.38	42.95	83.41	**60.38**	**61.21**	**68.35**
	Gains to DS* [38]	+3.68	+5.02	+3.18	+1.40	+1.62	+4.11	+7.50	+10.21	+5.75	+7.41	+7.52	+7.27

SF: Step Forecasting; **SR**: Step Recognition; **TR**: Task Recognition.
The top 3 performance scores of each downstream evaluation setting are highlighted with green cells (the darker green, the better).
* denotes the model was pre-trained on the *full* HowTo100M (HT100M) dataset; otherwise, a subset of HowTo100M containing 85K videos was used.
"**VNM + VTM + TCL + NRL**" represents "**VNM + VTM** (*wikiHow* + *HT100M*) + **TCL** (*wikiHow*) + **NRL** (*1 hop*)". Please see Supplementary Material for results when K=2.
DS* [38] reported results on COIN are SF: 38.2 (39.4 from 'Transformer w/ KB Transfer'), SR: 54.1 , and TR: 88.9 (90.0 from 'Transformer w/ KB Transfer'). Our downstream experimental configurations are different from that in [38] (e.g., w.r.t. temporal length of segments, downstream Transformer model – ours has less parameters, etc.).

Table 1. **Accuracies (% ↑) of the downstream procedure understanding tasks under the 12 evaluation settings**. Paprika that exploits the PKG outperforms the SOTA methods. Among our pre-training objectives, NRL is the most effective one, because it exploits the structural information of the PKG and elicits the procedural knowledge on the *order* and *relation* of *cross-task* steps to Paprika.

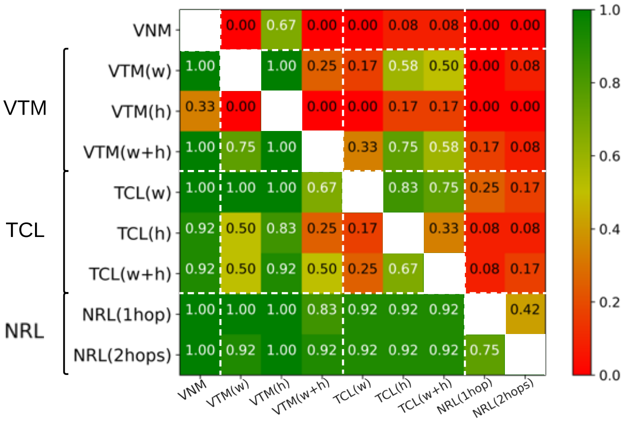


Figure 3. **Overall Performance Comparison.** This matrix compares the *overall* performance of our proposed pre-training objectives. The value in **entry (i, j)** is the **ratio of evaluation settings** in which **the accuracy of method i ≥ the accuracy of method j**. Here, 1 indicates method i outperforms method j in all 12 evaluation settings. The more **green** entries in the row of a method, the better its overall performance. NRL is the most effective method.

Comparing the three variants of TCL, TCL (w) > TCL (w+h) > TCL (h). Overall, TCL (w+h) is worse than TCL (w) because TCL depends on the quality of the pseudo labels of VTM. As utilizing the HowTo100M task names already leads to probably problematic matched tasks, asking $f(\cdot)$ to further identify the step nodes that these matched tasks need would introduce additional noise, which eventually undermines the overall downstream performance.

NRL (2 hops) > NRL (1 hop) overall. NRL (2 hops) has a worse or close performance than NRL (1 hop) only when the downstream task model $t(\cdot)$ is Transformer (Table 1). When $t(\cdot)$ is MLP, NRL (2 hops) is always clearly better. This is because when the capacity of $t(\cdot)$ is limited, it desires the input video representations to encode more comprehensive information. NRL with more hops indicates a larger exploration on the local graph structure of the PKG that a video segment belongs to; it can provide more related neighboring node/step information, and allow the learned video representations to excel at the downstream tasks.

We train Paprika using all pre-training objectives without tuning coefficient of each loss term. Paprika trained using all pre-training objectives yields the best result on 8 out of 12 evaluation settings, which suggests the four pre-training objectives can collaborate to lead to better results. Compared with NRL (1 hop), the performance gains brought by VNM, VTM and TCL are relatively small. This variant also fails to outperform NRL (2 hops) on the SF and SR tasks when $t(\cdot)$ is MLP. These results highlight the superiority of NRL.

We also experiment with the full HowTo100M data. Increasing the size of the pre-training dataset, for both DS and Paprika, accuracies are dropped on the COIN dataset when $t(\cdot)$ is MLP (due to MLP's limited capacity to exploit the features pre-trained on the large dataset and scale well), but we observe performance improvement of Paprika on the rest 9 out of 12 evaluation settings.

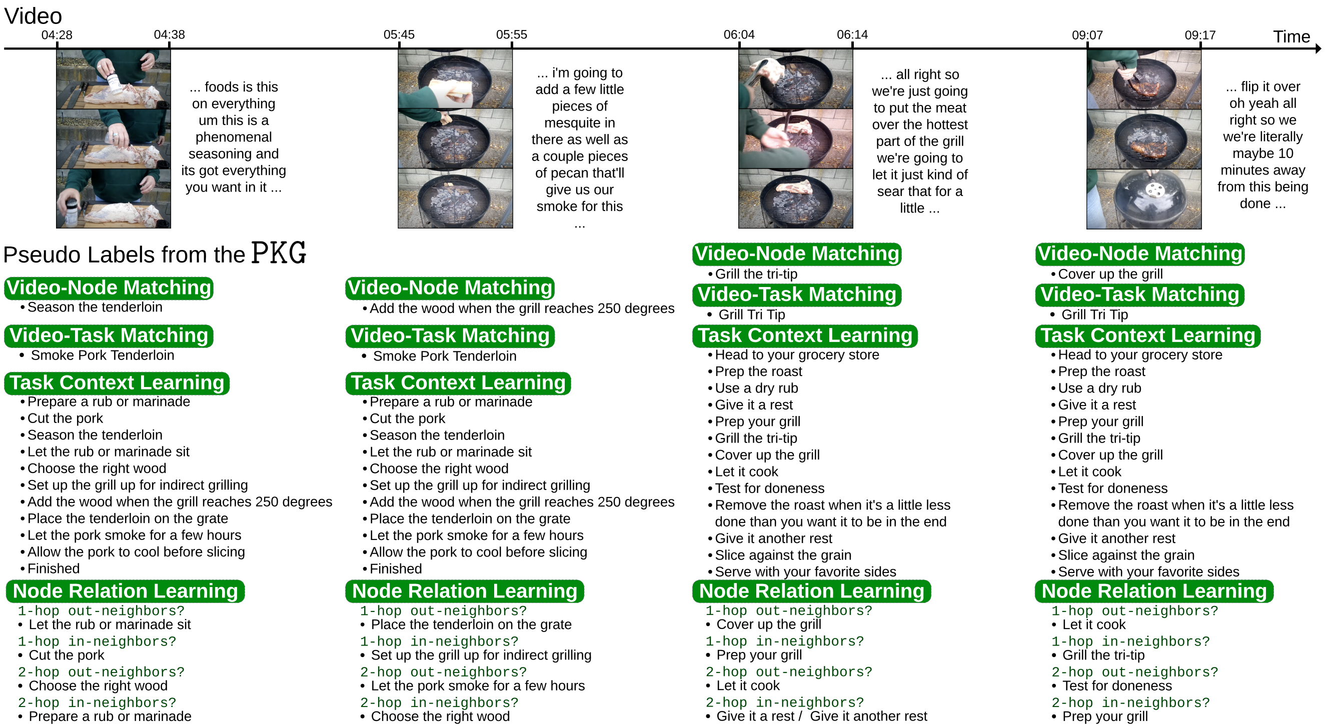

Video
04:28
04:38
05:45
05:55
06:04
06:14
09:07
09:17
Time
... foods is this on everything um this is a phenomenal seasoning and its got everything you want in it ...
... i'm going to add a few little pieces of mesquite in there as well as a couple pieces of pecan that'll give us our smoke for this ...
... all right so we're just going to put the meat over the hottest part of the grill we're going to let it just kind of sear that for a little ...
... flip it over oh yeah all right so we we're literally maybe 10 minutes away from this being done ...
Pseudo Labels from the PKG
Video-Node Matching
• Season the tenderloin
Video-Task Matching
• Smoke Pork Tenderloin
Task Context Learning
• Prepare a rub or marinade
• Cut the pork
• Season the tenderloin
• Let the rub or marinade sit
• Choose the right wood
• Set up the grill up for indirect grilling
• Add the wood when the grill reaches 250 degrees
• Place the tenderloin on the grate
• Let the pork smoke for a few hours
• Allow the pork to cool before slicing
• Finished
Node Relation Learning
1-hop out-neighbors?
• Let the rub or marinade sit
1-hop in-neighbors?
• Cut the pork
2-hop out-neighbors?
• Choose the right wood
2-hop in-neighbors?
• Prepare a rub or marinade
Video-Node Matching
• Add the wood when the grill reaches 250 degrees
Video-Task Matching
• Smoke Pork Tenderloin
Task Context Learning
• Prepare a rub or marinade
• Cut the pork
• Season the tenderloin
• Let the rub or marinade sit
• Choose the right wood
• Set up the grill up for indirect grilling
• Add the wood when the grill reaches 250 degrees
• Place the tenderloin on the grate
• Let the pork smoke for a few hours
• Allow the pork to cool before slicing
• Finished
Node Relation Learning
1-hop out-neighbors?
• Place the tenderloin on the grate
1-hop in-neighbors?
• Set up the grill up for indirect grilling
2-hop out-neighbors?
• Let the pork smoke for a few hours
2-hop in-neighbors?
• Choose the right wood
Video-Node Matching
• Grill the tri-tip
Video-Task Matching
• Grill Tri Tip
Task Context Learning
• Head to your grocery store
• Prep the roast
• Use a dry rub
• Give it a rest
• Prep your grill
• Grill the tri-tip
• Cover up the grill
• Let it cook
• Test for doneness
• Remove the roast when it's a little less done than you want it to be in the end
• Give it another rest
• Slice against the grain
• Serve with your favorite sides
Node Relation Learning
1-hop out-neighbors?
• Cover up the grill
1-hop in-neighbors?
• Prep your grill
2-hop out-neighbors?
• Let it cook
2-hop in-neighbors?
• Give it a rest / Give it another rest
Video-Node Matching
• Cover up the grill
Video-Task Matching
• Grill Tri Tip
Task Context Learning
• Head to your grocery store
• Prep the roast
• Use a dry rub
• Give it a rest
• Prep your grill
• Grill the tri-tip
• Cover up the grill
• Let it cook
• Test for doneness
• Remove the roast when it's a little less done than you want it to be in the end
• Give it another rest
• Slice against the grain
• Serve with your favorite sides
Node Relation Learning
1-hop out-neighbors?
• Let it cook
1-hop in-neighbors?
• Grill the tri-tip
2-hop out-neighbors?
• Test for doneness
2-hop in-neighbors?
• Prep your grill

Figure 4. **Pseudo labels generated by the PKG of one video** (title is “Grilling A Tri-Tip ...”). Frames and temporally overlapped subtitles of four segments sampled from this video were shown. For a succinct visualization, for each pre-training objective, we only show the result of the most confident pseudo label. TCL and NRL provide more procedure-level context information than VNM and VTM. Our pseudo labels entail a much higher relevance to each segment than the subtitle and allow Paprika to leverage cross-task information sharing.

4.4.2 Comparison to the State of the Art (SOTA)

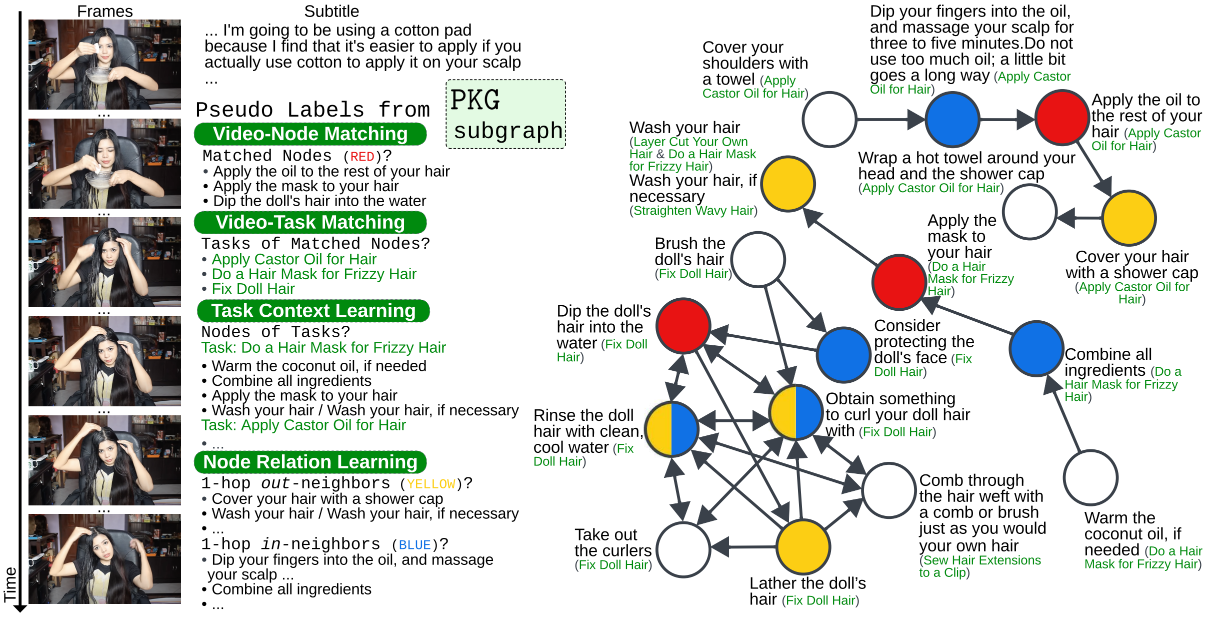
We have kept the model architectures and experimental setups the same between Paprika and the SOTA baselines. **MIL-NCE** [43] is a pre-training objective based on video-subtitle matching, and the subtitle can be weakly aligned to the video segment. This pre-training objective is widely used by video foundation models. We use the frozen S3D model released by the authors as $e(\cdot)$ in our framework (and to build the PKG). Our reported MIL-NCE results can be interpreted as the results of removing $f(\cdot)$ in our framework. **DS** [38] proposes to match a video segment's subtitle to a step *headline* in wikiHow by leveraging a pre-trained language model, i.e., MPNet [57]; and the matching results are used as the pre-training supervisory signals. We use their proposed objective to train $f(\cdot)$–the same MLP-based architecture used by Paprika–in our experiments.

Our Paprika outperforms the SOTA (Table 1). The large performance improvement of ours compared to MIL-NCE highlights the ability of our Paprika model in adapting the inferior video features to be instead competent at the procedure understanding tasks. Paprika also outperforms DS. Among our proposed pre-training objectives, VNM has the closest results to DS because both focus on learning step knowledge; the better results of VNM attribute to multimodal matching – matching the video *frames* to the step *nodes* that summarize and unite different step headlines in the same action. We perform ablation to match video frames to the wikiHow step headlines (VSM). VSM has a slightly better overall performance than DS, but worse than VNM. VTM, TCL and NRL learn more advanced procedural knowledge from the PKG, and therefore their gains over DS are even more obvious.

Paprika pre-trained with all four pre-training objectives obtains the highest gain over DS, which is $\mathbf{11.23\%}$ improvement in accuracy on the COIN task recognition task when $t(\cdot)$ is MLP and the HowTo100M subset is the pre-training dataset. Overall, the gains are larger when $t(\cdot)$ is MLP than Transformer. A shallow MLP downstream model, learned with features from Paprika pre-trained using our full pre-training objectives, even outperforms the Transformer downstream model learned with input features from the SOTA pre-trained models. This is because our proposed method allows the video feature to early encode relation information to address the limitation of a MLP model in lacking the relational reasoning capability.

4.5. Qualitative Results

We present the pseudo labels generated by the PKG of one long video in Fig. 4. Compared to the subtitles, the source of information that prior pre-training methods often use for supervision, pseudo labels generated by the PKG entail a higher relevance to each segment. Subtitles are noisy



... I'm going to be using a cotton pad because I find that it's easier to apply if you actually use cotton to apply it on your scalp ...

Pseudo Labels from PKG subgraph

Video-Node Matching

Matched Nodes (RED)?
- Apply the oil to the rest of your hair
- Apply the mask to your hair
- Dip the doll's hair into the water

Video-Task Matching

Tasks of Matched Nodes?
- Apply Castor Oil for Hair
- Do a Hair Mask for Frizzy Hair
- Fix Doll Hair

Task Context Learning

Nodes of Tasks?
Task: Do a Hair Mask for Frizzy Hair
- Warm the coconut oil, if needed
- Combine all ingredients
- Apply the mask to your hair
- Wash your hair / Wash your hair, if necessary

Task: Apply Castor Oil for Hair
- ...

Node Relation Learning

1-hop *out*-neighbors (YELLOW)?
- Cover your hair with a shower cap
- Wash your hair / Wash your hair, if necessary
- ...

1-hop *in*-neighbors (BLUE)?
- Dip your fingers into the oil, and massage your scalp ...
- Combine all ingredients
- ...

Cover your shoulders with a towel (Apply Castor Oil for Hair)
Dip your fingers into the oil, and massage your scalp for three to five minutes.Do not use too much oil; a little bit goes a long way (Apply Castor Oil for Hair)
Apply the oil to the rest of your hair (Apply Castor Oil for Hair)
Wash your hair (Layer Cut Your Own Hair & Do a Hair Mask for Frizzy Hair)
Wash your hair, if necessary (Straighten Wavy Hair)
Wrap a hot towel around your head and the shower cap (Apply Castor Oil for Hair)
Apply the mask to your hair (Do a Hair Mask for Frizzy Hair)
Cover your hair with a shower cap (Apply Castor Oil for Hair)
Brush the doll's hair (Fix Doll Hair)
Dip the doll's hair into the water (Fix Doll Hair)
Consider protecting the doll's face (Fix Doll Hair)
Combine all ingredients (Do a Hair Mask for Frizzy Hair)
Rinse the doll hair with clean, cool water (Fix Doll Hair)
Obtain something to curl your doll hair with (Fix Doll Hair)
Comb through the hair weft with a comb or brush just as you would your own hair (Sew Hair Extensions to a Clip)
Warm the coconut oil, if needed (Do a Hair Mask for Frizzy Hair)
Take out the curlers (Fix Doll Hair)
Lather the doll's hair (Fix Doll Hair)

Figure 5. **Pseudo labels of one segment and the subgraph of the PKG that this segment belongs to.** The PKG encodes the procedural knowledge of the general order and relation of steps from multiple tasks. This is because a node's k-hop neighbors can come from multiple tasks, and the edge direction encodes the general execution order of steps (the order that was observed in data – not in one specific video).

because the narrator may not directly describe the step. E.g., in the 1st segment, the narrator only mentions "this is a phenomenal seasoning" without explicitly describing the step "seasoning the tri-tip". When the camera records how a narrator is performing a step, the narrator may omit verbally or formally describing the step. Out of this observation, we leverage a multimodal matching function.

Assigning wikiHow steps to a video, allows one video to leverage cross-task information sharing. As shown in the pseudo labels of VNM, the matched step headlines can come from another task. E.g., "Season the tenderloin" and "Add the wood ..." are step headlines of the task "Smoke Pork Tenderloin", but the task of the video is "Grill Tri-Tip". In the wikiHow article of the task "Grill Tri-Tip" (shown in the TCL blocks of the 3rd and 4th segments), the step headline corresponds to the action "seasoning" is "Prep the roast", which is vague, and no step headlines describe the action "adding wood". Instead, "seasoning" and "adding wood" have a clearer step headlines to describe them in the wikiHow article of "Smoke Pork Tenderloin".

TCL and NRL provide more procedure-level context information as shown in Fig. 4. The procedural knowledge conveyed by TCL and NRL is the *general* prior knowledge about the step and task of the current segment, and the knowledge is not constrained to the current step, task, or video. In other words, steps shown in the TCL or the NRL blocks can be absent in this video demonstration.

In Fig. 5, we show pseudo labels of one video segment and the subgraph of the PKG that the segment belongs to. The top 3 matched nodes's step headlines come from different tasks, and especially the top 2 well describe the step of the video segment. NRL allows Paprika to learn the knowledge on order and relation of cross-task steps because pseudo labels of NRL are led by the structure of the PKG.

5. Conclusion

We show how to learn a video representation for procedure understanding in instructional videos that encodes procedural knowledge. The key is to leverage a Procedural Knowledge Graph (PKG) to inject procedural knowledge into the video representation, which improves the state-of-the-art performance on several tasks.

Limitations & Future Directions: Our model is built on top of frozen video and language encoders. Future work should explore jointly updating these deep visual and text representations while also learning the procedural knowledge model. Future work should also extend our methodology beyond the existing downstream tasks to more complex procedure understanding benchmarks.

Social Impact: The final models may also be limited to perform video understanding on tasks not represented in training. These datasets primarily reflect the culture of only a portion of the world's population, and may contain that culture's socioeconomic biases on gender, race, ethnicity, or other features. These biases may be present in the generated pseudo labels, subgraphs, and/or overall video understanding capabilities of the resulting system.

Acknowledgments: Prof. Mubbasir Kapadia was supported in part by NSF awards: IIS-1703883, IIS-1955404, IIS-1955365, RETTL-2119265, and EAGER-2122119.

References

[1] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021. 1, 2, 3

[2] 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. 2

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 3

[4] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *Advances in Neural Information Processing Systems*, 33:25–37, 2020. 2

[5] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *Advances in Neural Information Processing Systems*, 33:9758–9770, 2020. 2

[6] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9922–9931, 2020. 2

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 5

[8] 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. 2

[9] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 2, 3

[10] 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. 2

[11] Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, et al. Multimodal clustering networks for self-supervised learning from unlabeled videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8012–8021, 2021. 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] Nikita Dvornik, Isma Hadji, Hai Pham, Dhaivat Bhatt, Brais Martinez, Afsaneh Fazly, and Allan D Jepson. Graph2vid: Flow graph to video grounding for weakly-supervised multi-step localization. *arXiv preprint arXiv:2210.04996*, 2022. 2

[14] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. In *European Conference on Computer Vision*, pages 557–573. Springer, 2020. 2

[15] Ehsan Elhamifar and Zwe Naing. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6341–6350, 2019. 2

[16] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017. 2

[17] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridging video-text retrieval with multiple choice questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16167–16176, 2022. 2

[18] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12046–12055, 2019. 2

[19] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, Chiho Choi, and Behzad Dariush. Weakly-supervised online action segmentation in multi-view instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13780–13790, 2022. 2

[20] Reza Ghoddoosian, Saif Sayed, and Vassilis Athitsos. Hierarchical modeling for task recognition and action segmentation in weakly-labeled instructional videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1922–1932, 2022. 2

[21] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13515, 2021. 2

[22] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 3

[23] 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

[24] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al.

Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 2

[25] Lei Ji, Chenfei Wu, Daisy Zhou, Kun Yan, Edward Cui, Xilin Chen, and Nan Duan. Learning temporal video procedure segmentation from an automatically collected large dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1506–1515, 2022. 2

[26] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. *arXiv preprint arXiv:2210.03929*, 2022. 2

[27] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. *arXiv preprint arXiv:2111.01024*, 2021. 2

[28] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8545–8552, 2019. 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[30] Mahnaz Koupaee and William Yang Wang. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*, 2018. 2, 4

[31] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE international conference on computer vision*, pages 667–676, 2017. 2

[32] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. 2

[33] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963, 2022. 1, 2, 3

[34] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065, 2020. 1, 2

[35] Muheng Li, Lei Chen, Yueqi Duan, Zhilan Hu, Jianjiang Feng, Jie Zhou, and Jiwen Lu. Bridge-prompt: Towards ordinal action understanding in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19880–19889, 2022. 2

[36] Muheng Li, Lei Chen, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Order-constrained representation learning for instructional video prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 2

[37] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Hongfa Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. *arXiv preprint arXiv:2206.01670*, 2022. 2

[38] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. 2, 3, 4, 5, 6, 7

[39] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 2

[40] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8085–8095, 2021. 2

[41] Zijia Lu and Ehsan Elhamifar. Set-supervised action learning in procedural task videos via pairwise order consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19903–19913, 2022. 2

[42] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[43] 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. 1, 2, 3, 4, 5, 6, 7

[44] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[45] 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 *ICCV*, 2019. 2, 5

[46] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12486, 2021. 2

[47] AJ Piergiovanni, Anelia Angelova, Michael S Ryoo, and Irfan Essa. Unsupervised discovery of actions in instructional videos. *arXiv preprint arXiv:2106.14733*, 2021. 2

[48] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2021. 2

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language super-

vision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[50] Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1255–1265, 2021. 2

[51] Ludan Ruan and Qin Jin. Survey: Transformer based video-language pre-training. *AI Open*, 2022. 2, 3

[52] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *arXiv preprint arXiv:2207.00419*, 2022. 2

[53] Fadime Sener, Rishabh Saraf, and Angela Yao. Learning video models from text: Zero-shot anticipation for procedural actions. *arXiv preprint arXiv:2106.03158*, 2021. 2

[54] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17959–17968, 2022. 2

[55] Yuhan Shen and Ehsan Elhamifar. Semi-weakly-supervised learning of complex actions from instructional task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3344–3354, 2022. 2

[56] 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

[57] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020. 7

[58] 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

[59] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015. 2

[60] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 2

[61] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 2

[62] 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. 2

[63] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237, 2022. 3

[64] 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

[65] Yansong Tang, Jiwen Lu, and Jie Zhou. Comprehensive instructional video analysis: The coin dataset and performance evaluation. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3138–3153, 2020. 2, 5

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[67] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106, 2016. 2

[68] Dong Wang, Di Hu, Xingjian Li, and Dejing Dou. Temporal relational modeling with self-supervision for action segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2729–2737, 2021. 2

[69] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4006–4015, 2019. 2

[70] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European conference on computer vision*, pages 504–521. Springer, 2020. 2

[71] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2

[72] Qingyun Wang, Manling Li, Hou Pong Chan, Lifu Huang, Julia Hockenmaier, Girish Chowdhary, and Heng Ji. Multimedia generative script learning for task planning. *arXiv preprint arXiv:2208.12306*, 2022. 2

[73] Shaojie Wang, Wentian Zhao, Ziyi Kou, Jing Shi, and Chenliang Xu. How to make a blt sandwich? learning vqa towards understanding web instructional videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1130–1139, 2021. 2

[74] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 2

[75] 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

[76] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. Vlm: Task-agnostic video-language model pre-training for video understanding. *arXiv preprint arXiv:2105.09996*, 2021. 2

[77] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, 2021. 1, 2

[78] 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

[79] Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. Visual goal-step inference using wikihow. *arXiv preprint arXiv:2104.05845*, 2021. 2

[80] Zhengyuan Yang, Jingen Liu, Jing Huang, Xiaodong He, Tao Mei, Chenliang Xu, and Jiebo Luo. Cross-modal contrastive distillation for instructional activity anticipation. *arXiv preprint arXiv:2201.06734*, 2022. 2

[81] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022. 2

[82] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021. 2

[83] Li Zhang, Qing Lyu, and Chris Callison-Burch. Intent detection with wikihow. *arXiv preprint arXiv:2009.05781*, 2020. 2

[84] Li Zhang, Qing Lyu, and Chris Callison-Burch. Reasoning about goals, steps, and temporal ordering with wikihow. *arXiv preprint arXiv:2009.07690*, 2020. 2

[85] Luowei Zhou, Chenliang Xu, and Jason J Corso. Procnets: Learning to segment procedures in untrimmed and unconstrained videos. *arXiv preprint arXiv:1703.09788*, 2(6):7, 2017. 2

[86] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[87] Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. Show me more details: Discovering hierarchies of procedures from semi-structured web data. *arXiv preprint arXiv:2203.07264*, 2022. 2

[88] 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. 2, 5