Event-Guided Procedure Planning from Instructional Videos with Text Supervision

An-Lan Wang*, Kun-Yu Lin*, Jia-Run Du, Jingke Meng†, Wei-Shi Zheng†
School of Computer Science and Engineering, Sun Yat-sen University, China
Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
{wanganlan, linky5, dujr6}@mail2.sysu.edu.cn, mengjke@gmail.com, wszheng@ieee.org

Abstract

In this work, we focus on the task of procedure planning from instructional videos with text supervision, where a model aims to predict an action sequence to transform the initial visual state into the goal visual state. A critical challenge of this task is the large semantic gap between observed visual states and unobserved intermediate actions, which is ignored by previous works. Specifically, this semantic gap refers to that the contents in the observed visual states are semantically different from the elements of some action text labels in a procedure. To bridge this semantic gap, we propose a novel event-guided paradigm, which first infers events from the observed states and then plans out actions based on both the states and predicted events. Our inspiration comes from that planning a procedure from an instructional video is to complete a specific event and a specific event usually involves specific actions. Based on the proposed paradigm, we contribute an Event-guided Prompting-based Procedure Planning (E3P) model, which encodes event information into the sequential modeling process to support procedure planning. To further consider the strong action associations within each event, our E3P adopts a mask-and-predict approach for relation mining, incorporating a probabilistic masking scheme for regularization. Extensive experiments on three datasets demonstrate the effectiveness of our proposed model.

1. Introduction

In this work, we focus on the procedure planning task from instruction videos [7, 5, 38, 53]. Given the current state (a frame or a clip), procedure planning aims to predict a sequence of actions to reach a desired goal state. This goal-driven decision-making capability comes naturally to humans but is difficult for machine learning systems to acquire. Therefore, due to its wide real-world applications, *e.g.*, Autopilot [48] and Robotic systems [3], solving procedure planning is of great significance.

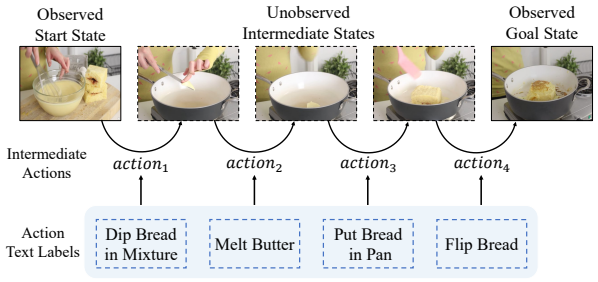

Figure 1: Illustration of the *semantic gap*, *i.e.*, the contents in the observed visual states are semantically different from the elements of some action text labels in a procedure. We show a four-action procedure as an example. As shown above, it is difficult to predict that we should "Melt Butter" by observing the start and goal states, since we see no butter but some other things (*e.g.*, bread, pan, mixture) from the observed states. However, if we take the event of procedure "Make French Toast" into thinking, "Melt Butter" is an indispensable step. Best viewed in color.

Early works [7, 5, 38] typically address the procedure planning task in an auto-regressive manner, following traditional sequential modeling works [15, 27, 8, 41]. Specifically, given both intermediate action labels and intermediate visual states as supervision, these works adopt two-branch networks to predict the action labels and representations of states separately, based on the input start and goal states. These methods mainly differ in the feature extractor for sequential modeling, *e.g.*, DDN [7] uses RNNs [27], Plate [38] uses Transformers [42]. However, all these methods need access to intermediate visual states for supervision. In such a setting, it is necessary to precisely identify the start and end timestamps of all actions in training videos, which is time-consuming and labor-intensive for annotation.

* indicates equal contribution. † indicates the corresponding author.

Recent work [53] provides a way to reduce annotation efforts. They study a weakly-supervised setting that removes the need for intermediate visual states as supervision, named Procedure Planning from instructional videos with Text Supervision (PPTS). In PPTS, text representations of intermediate action labels are introduced for supervision, leveraging the power of a pre-trained vision-language model [28]. To tackle PPTS, P3IV [53] proposes a memory-augmented Transformer for sequential modeling.

In previous PPTS methods, the prediction of intermediate actions is conditioned on only the observed start and goal visual states. However, it is challenging to build a *direct* connection between *observed* visual states and *unobserved* intermediate actions, due to a large semantic gap between them. This semantic gap refers to that the contents in the observed visual states are semantically different from the elements of some action text labels in a procedure. For example, as shown in Figure 1, it is difficult to directly predict the action **"Melt Butter"** by observing the start and goal states, since we see no butter but some other things (*e.g.*, bread, pot, mixture) from the observed states.

To bridge the semantic gap, we propose a novel event-guided paradigm (as shown in Figure 2), which is not explored by previous works. Our proposed event-guided paradigm first infers the events of procedures based on the observed visual states and then predicts a sequence of actions based on both the states and predicted events. Our inspiration comes from the fact that planning a procedure from an instructional video is to complete a specific event (*i.e.*, a procedure matches a clear intention). And, since a specific event usually involves specific actions, we can use the event information to support the procedure planning. For example, as shown in Figure 1, after identifying the event **"Make French Toast"** from the observed visual states, we can plan out the action **"Melt Butter"**, since melting butter is essential to attain crispy French toast. In addition, there are usually strong associations between actions within an event, which can be utilized for planning a reasonable procedure. Also shown in Figure 1, suppose we already know that this procedure is to make French toast and the first three actions are **"Dip Bread in Mixture**→ **Melt Butter**→ **Put Bread in Pan"**, we can deduce that the fourth action should be **"Flip Bread"** because no one will make French toast with only one side fried.

We contribute an Event-guided Prompting-based Procedure Planning (E3P) model based on our proposed event-guided paradigm. Given event labels as supervision, our proposed E3P uses an Event-aware Prompt Generator to encode event information into the hand-crafted prompts of intermediate actions. We find that the events can generally be inferred from the observed start and goal visual states. After sequential modeling based on event-aware prompts, we propose an Action Relation Mining module to model the associations between actions within each event. Our Action Relation Mining module adopts a mask-and-predict approach and incorporates a probabilistic masking scheme for regularization, aiming to fully consider the action associations during training. We conduct extensive experiments on three datasets, and the results demonstrate that our proposed E3P outperforms previous state-of-the-art methods by a large margin.

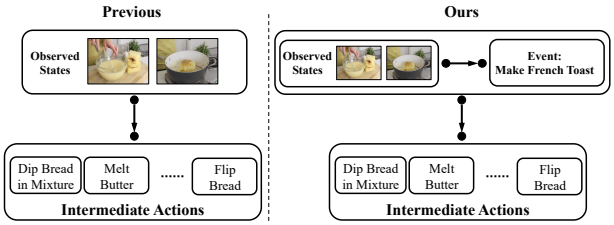

Figure 2: Previous methods build a *direct* connection between the *observed* start and goal visual states and *unobserved* intermediate actions, ignoring a large semantic gap between them. In contrast, our work proposes a novel event-guided paradigm to bridge the semantic gap.

2. Related Works

2.1. Procedure Planning

Procedure planning from instructional videos aims to predict a reasonable plan conditioned on the start and goal visual states. Early works on procedure planning adopt a two-branch auto-regressive method (*i.e.*, action and visual branch) to predict actions and visual representation of intermediate states. These works involve different network architectures for modeling, varying from recurrent neural networks [7], transformers [38] to adversarial networks [5]. Recently, Zhao *et al*. [53] proposes Procedure Planning from instructional videos with Text Supervision (PPTS), where text representations of intermediate actions are introduced as supervision. To address PPTS, P3IV [53] proposes a memory-augmented Transformer for sequential modeling. Different from previous works, we propose a novel event-guided paradigm for PPTS, aiming to bridge the semantic gap between observed visual states and unobserved intermediate actions.

2.2. Action Recognition

With the success of deep learning, effective video classification architectures for action recognition have been proposed, including RNNs [9, 50, 40], 2D CNNs [17, 24, 35, 36] and 3D CNNs [28, 6, 39, 49]. Recently, with the success of Vision Transformer [10, 26], many works adopted Vision Transformer for action recognition [13, 31, 51, 52]. Different from the traditional action recognition task, the procedure planning task studied in our work is much more

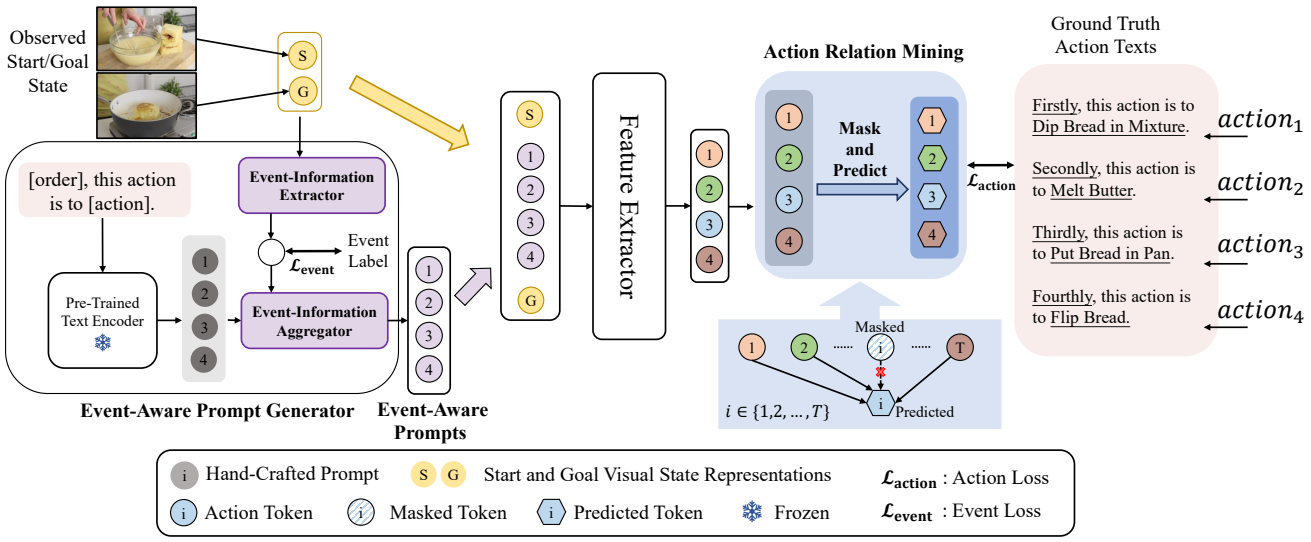

Figure 3: Overview of our Event-guided Prompting-based Procedure Planning (E3P) model. Our proposed E3P model follows a novel event-guided paradigm to bridge the semantic gap between observed visual states and unobserved intermediate actions. In this figure, we take a four-action procedure as an example. First of all, we use a pre-trained text encoder to extract the representations of hand-crafted action prompts. Supervised by event labels, we extract event information from the given visual states and integrate them into the prompts to generate Event-Aware Prompts. After sequential modeling by the feature extractor, the Action Relation Mining module exploits the action associations adopting a "mask-and-predict" approach. Best viewed in color.

challenging, since the relation between actions should be taken into account beyond predicting individual actions.

2.3. Action Anticipation

Different from procedure planning based on start and goal states, action anticipation aims to predict future actions based on past states. Early action anticipation works [20, 44] focus on predicting a single future action within a few moments. Farha *et al*. [1] proposed long-term action anticipation to predict a sequence of future actions by using two networks (*i.e*., a RNN and a CNN). To reduce the error accumulation caused by iterative predictions, Some methods [12, 1, 21] took action labels of past states as input. Gong *et al*. [14] adopted an end-to-end transformer model to anticipate all future actions in parallel. Procedure planning is similar to an action anticipation task with goal guidance, leading to more constraints for modeling.

2.4. Prompting and Regularization

There have been longstanding efforts for prompt design in NLP [25, 34]. Recently, CLIP [33] explored prompting for image understanding by formulating a image-text matching problem. Some other works [19, 30, 46, 23] explored prompting for video understanding.

To alleviate the overfitting problem in deep neural networks, many dropout-like techniques were proposed for regularization [18, 45, 4, 47]. Specific to procedure planning, our proposed Action Relation Mining module involves a probabilistic masking scheme for regularization, aiming to fully consider the action associations.

3. Event-Guided Procedure Planning

In this section, we elaborate on our proposed Event-guided Prompting-based Procedure Planning (E3P) model, which follows a novel event-guided paradigm.

3.1. Problem Formulation and Model Overview

Problem Formulation. Our task is procedure planning from instructional videos with text supervision. Given the start visual state o_s and goal visual state o_g, a model aims to predict a procedure of T action steps, *i.e*., an action sequence $\{a_1, ..., a_T\}$, transforming the visual state from the o_s to o_g. The number of actions T is provided, also known as the prediction horizon. Following Zhao *et al*. [53], we use text representations of action labels as supervision for training.

Model Overview. Figure 3 illustrates the proposed E3P model based on prompting-based feature modeling. Given the state representations o_s and o_g, we extract the event information of the procedure through an event-information extractor and then encode it into hand-crafted text prompts to generate Event-Aware Prompts. Subsequently, the Event-Aware Prompts and visual state representations are then fed

into the feature extractor for sequential modeling, producing T action tokens. These tokens are then fed into an Action Relation Mining module, which uses a "mask-and-predict" approach to model the relation between actions. Finally, we output a sequence of actions, namely, the procedure. Notably, we do not directly predict the distribution over possible actions, and instead by first predicting the feature representations of actions and then predicting the distribution over actions (i.e., by calculating the similarity between predicted action features and all action text features).

3.2. Prompting-based Feature Modeling

First of all, we introduce the prompting-based feature modeling of our approach. Since the PPTS task requires predicting a procedure (*i.e.*, an action sequence), and this sequence is order-sensitive, we leverage the power of a pre-trained vision-language model. Specifically, we use a hand-crafted text prompt in the format of *"[order], this action is to [action]"* as input, which contains an order blank and an action blank. Then, we obtain the representations of prompts using a pre-trained vision-language model.

We use the representations of ground truth action texts as supervision. In specific, we construct a sentence in the same format as the above hand-crafted prompts, with the two blanks filled in. For example, if the first action is "Melt Butter", the constructed sentence would be "Firstly, this action is to Melt Butter". Then we use the vision-language model to encode it into text representations. Following P3IV [53], we use the same loss function [16] to supervise the model, which is formulated as follows:

$$\mathcal{L}_{action} = -\sum_{t=1}^{T}\left[\log\frac{\exp\left(l_{+}\cdot\widetilde{a}_{t}\right)}{\sum_{j=1}^{N}\exp\left(l_{j}\cdot\widetilde{a}_{t}\right)}\right], \quad (1)$$

where l_+ is the ground truth action text presentation, l_j is the text representation of the j-th action, N is the number of actions in the dataset and $\widetilde{a}_t$ is the t-th action token (the final output of our model).

3.3. Event-aware Prompt Generator

To bridge the semantic gap between the observed visual states and unobserved intermediate actions, we propose an Event-aware Prompt Generator that encodes the event information to guide the procedure planning process. Our inspiration comes from that planning a procedure from an instructional video is to complete a specific event and a specific event usually involves specific actions. Our Event-aware Prompt Generator mainly contains two parts, *i.e.*, an event-information extractor and an event-information aggregator, aiming to extract event information and encode event information into the text prompts, respectively.

Specifically, the event-information extractor $\mathbf{E_e}$ extracts the event information $\hat{e}$ from the start state o_s and goal state o_g, which is given as follows:

$$\hat{e} = \mathbf{E}_e(o_s, o_g). \quad (2)$$

To guide the event information extraction, we stack a classification head $h_e(\cdot)$ on top of the $\mathbf{E_e}(\cdot,\cdot)$. And, the event loss is given as follows:

$$\mathcal{L}_{event} = \mathrm{CE}(h_e(\hat{e}), y_e), \quad (3)$$

where CE is the cross-entropy loss and y_e is the ground truth event label provided by the dataset.

Then, we encode the event information $\hat{e}$ into the hand-crafted prompt representations $p_{1:T}$ using the event information aggregator. The event-information aggregator takes T prompt representations and the event information as inputs and produces T event-aware prompts, which is formulated as follows:

$$p^e_{1:T} = \mathcal{F}(p_{1:T}, \hat{e}), \quad (4)$$

where $\mathcal{F}(\cdot)$ is the event-information aggregator and $p^e_{i:T}$ are the event-aware prompts. The introduction of event information would constrain our model to predict actions more related to the event of procedure.

Next, we concatenate generated event-aware prompts with start and goal visual state representations. After positional encoding, we input the event-aware prompts and visual states into the feature extractor for sequential modeling. The input of the feature extractor is given as follows:

$$Q = [o_s, p^e_1, p^e_2, ..., p^e_T, o_g], \quad (5)$$

where p^e_i is the i-th event-aware prompts. After sequential modeling, the feature extractor outputs $T+2$ tokens. We take the middle T action tokens $\hat{a}_{1:T}$ as input to our next module.

3.4. Action Relation Mining within Events

In this section, we propose to model the relation between actions within individual events to support the procedure planning. Usually, there are strong associations between actions within an event, which can be utilized for planning a reasonable procedure. Accordingly, we propose an Action Relation Mining (ARM) module exploiting a "mask-and-predict" approach, which refines the prediction of procedure planning by mining the relation between actions within events.

For our ARM module, we adopt the masked self-attention with a specially designed mask as core. Specifically, the input to the ARM module is a list of action tokens, *i.e.*, $\hat{Q} = [\hat{a}_1, \hat{a}_i, ..., \hat{a}_T]$. For the masked self-attention, we use a *deterministic* mask $M \in \mathbb{R}^{T\times T}$, where all elements on the main diagonal are manually set to zero. Such a mask

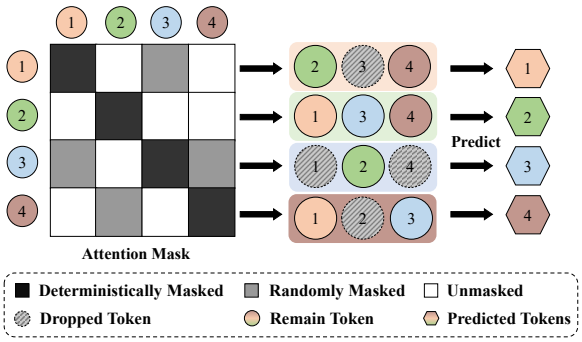

Figure 4: Illustration of the mask in our Action Relation Modeling module. We take a four-action procedure as an example. For the attention mask, we first set the elements of the main diagonal to zero and then randomly set some other elements to zero by a drop rate τ.

design ensures that our ARM model can leverage the information of all other actions for the prediction of the one masked action. The *deterministic* mask is defined as:

$$M_{i,j} = \begin{cases} 0, & \text{if } i = j, \\ 1, & \text{otherwise}, \end{cases} \tag{6}$$

where i and j represent the row and column of the attention mask matrix respectively. After feature modeling by our ARM module, we obtain the final prediction of action sequence by a residual connection, *i.e.* $\widetilde{a}_{1:T} = \hat{a}_{1:T} + \check{a}_{1:T}$, where $\check{a}_{1:T}$ is the prediction of ARM module and $\widetilde{a}_{1:T}$ is the final prediction. Intuitively, our ARM module conducts a refinement on the basic procedure prediction $\hat{a}_{1:T}$.

DropRelation. In the above relation mining process, we adopt a "mask-and-predict" approach with only one token masked out. However, such a modeling focuses on modeling the relation between the one masked action token and all other action tokens (*e.g.*, one masked action token and three unmasked action tokens in a four-action procedure) during training, thus it may miss some associations between actions. For example, suppose we have a four-action sequence "Action 1 $\rightarrow$ Melt Butter $\rightarrow$ Action 3 $\rightarrow$ Flip Bread", where the first and third actions are unknown. We are still able to infer that the third action is "Put Bread in Pan" according to the known two actions, without knowing the first action. Therefore, to fully consider the action associations, we equip our ARM module with a regularization technique named DropRelation.

Specifically, in addition to dropping tokens in the main diagonal, we randomly drop some of the other tokens during the "mask-and-predict" process. For each item in the mask, there is a random variable following a uniform distribution, denoted by $\alpha_{i,j}$. According to the $\alpha_{i,j}$, we obtain a *probabilistic* mask $\widetilde{M}$ as follows:

$$\widetilde{M}_{i,j} = \begin{cases} 1, & \text{if } i \neq j \text{ and } \alpha_{ij} > \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where τ refers to the drop rate. In this way, we randomly drop some connections (relation) between action tokens to regularize the relation mining process. In addition, we ensure that dropping at most $T-2$ tokens in a single row in the $\widetilde{M}_{i,j}$. An illustration of the masking scheme in our ARM is shown in Figure 4.

3.5. Traning and Inference

During training, the overall objective is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{action}} + \mathcal{L}_{\text{event}}. \tag{8}$$

During inference, following previous methods [53], our model only accesses to the start and goal visual states. Notably, the DropRelation regularization is only used during training and turned off during inference. Moreover, following previous work [53], we adopt the Viterbi [43] post process as well.

4. Experiments

We conduct experiments on three datasets and use three metrics to verify the effectiveness of our proposed Event-guided Prompting-based Procedure Planning (E3P) method.

4.1. Datasets and Metrics

Datasets: We conduct experiments on the following three datasets: (1) **CrossTask** [54] contains instructional videos collected for 83 different events, which are divided into 18 primary and 65 related events. Following previous works [7, 38, 53], we use the primary events, containing 2750 videos with an average of 7.6 actions per video. (2) **Narrated Instructional Videos (NIV)** [2] is a dataset collected from real-world instruction videos from the Internet. This dataset contains 150 videos of five events, with an average of 9.5 actions per video. (3) **COIN** is a large labeled instructional video dataset, which is collected from YouTube and consists of 11827 videos related to 778 different actions and on average 3.6 actions per video. For all three datasets, following previous works [7, 53], we adopt 70%/30% to create train/test splits and use a shift window to curate the dataset into plans with different time horizons.

Metrics: We use three evaluation metrics: (1) **Success Rate (SR)** considers a procedure successful only if it exactly matches the ground truth. (2) **mean Accuracy (mAcc)** considers the match of single action between predicted and ground truth action sequences, where action matches the ground truth at the same timestamp is considered correct. (3) **mean Intersection over Union (mIoU)** treats the predicted and ground truth action sequences as two sets, and measures the overlap between them. Note that mIoU is agnostic to the order of actions, which is only used as an auxiliary metric (to measure whether a model predicts the correct action set for the procedure). For more details about these metrics, please refer to DDN [7].

Table 1: Comparison with the state-of-the-art methods on CrossTask for prediction horizon $T \in \{3, 4\}$. SR, mAcc, and mIoU indicate Success Rate, mean Accuracy and mean Intersectin over Union, respectively. The numbers in bold-faced and in underline indicate the highest and the second-highest result, respectively. The column *Supervision* indicates the type of supervision used in training, *i.e.*, visual state, and action text. † indicates that data augmentation is in usage during training.

Methods	Year	Supervision	$T = 3$			$T = 4$		
			SR↑	mAcc↑	mIoU↑	SR↑	mAcc↑	mIoU↑
Random	-	-	<0.01	0.94	1.66	<0.01	0.83	1.66
Retrieval-Based	-	Visual State	8.05	23.30	32.06	3.95	22.22	36.97
WLTDO [11]	2018	Visual State	1.87	21.64	31.70	0.77	17.92	26.43
UAAA [12]	2019	Visual State	2.15	20.21	30.87	0.98	19.86	27.09
UPN [37]	2018	Visual State	2.89	24.39	31.56	1.19	21.59	27.85
DDN [7]	2020	Visual State	12.18	31.29	47.48	5.97	27.10	48.46
Ext-GAIL w/o Aug. [5]	2021	Visual State	18.01	43.86	57.16	-	-	-
Ext-GAIL [5] †	2021	Visual State	21.27	49.46	61.70	<u>16.41</u>	43.05	60.93
P3IV w/o Adv. [53]	2022	Text	22.12	45.57	67.40	-	-	-
P3IV [53]	2022	Text	<u>23.34</u>	<u>49.96</u>	<u>73.89</u>	13.40	<u>44.16</u>	<u>70.01</u>
Ours	-	Text	**26.40**	**53.02**	**74.05**	**16.49**	**48.00**	**70.16**

4.2. Implementation Details

For a fair comparison, we follow the previous approach [53] to extract the representations of start and goal visual states using the S3D network [28] pretrained on the HowTo100M [29] dataset. The text encoder is adopted from the pre-trained CLIP because the text model of P3IV [53] cannot encode prompt sentences. We use a Transformer as feature extractor.

We use Adam [22] optimizer with a weight decay of 0.4 and set the learning rate as 7e-4. Our model is trained for 200 epochs with a batch size of 32. We report the average results over three random trials. The method is implemented in PyTorch [32]. Please refer to the Appendix for more implementation details.

4.3. Comparison with State-of-the-Arts

On the CrossTask dataset, we compare our E3P with two types of methods in procedure planning, and our proposed E3P outperforms previous methods in all metrics as shown in Table 1. Compared with previous text-supervised methods, our E3P obtains significant improvement, *e.g.*, 3.06% for $T = 3$ and 3.09% for $T = 4$ in terms of Success Rate (SR). The results demonstrate that our E3P effectively captures the relation of actions and predicts more accurate procedures, which is attributed to our proposed event-guided paradigm. Our model performs slightly better than P3IV [53] in mIoU. This is because P3IV adopts an adversarial strategy during training and samples 1500 procedures in the inference phase to make the final prediction for each procedure, while we make only one prediction. By removing the adversarial strategy from P3IV (*i.e.*, P3IV w/o Adv), our model outperforms it by 6.65% ($T = 3$) in terms of mIoU. In addition, we compare our E3P with methods that use intermediate visual states as supervision, and our E3P still outperforms all these methods.

Table 2: Comparison with the state-of-the-art methods on a large dataset COIN for prediction horizon $T \in \{3, 4\}$. † indicates using the visual state as supervision. The bold-faced and underlined numbers indicate the highest and the second-highest performance, respectively.

Methods	$T = 3$			$T = 4$		
	SR↑	mAcc↑	mIoU↑	SR↑	mAcc↑	mIoU↑
Random	<0.01	<0.01	2.47	<0.01	<0.01	2.32
Retrieval-Based†	4.38	17.40	32.06	2.71	14.29	36.97
DDN [7]†	13.90	20.19	64.78	11.13	17.71	68.06
P3IV [53]	<u>15.40</u>	<u>21.67</u>	<u>76.31</u>	<u>11.32</u>	<u>18.85</u>	<u>70.53</u>
Ours	**19.57**	**31.42**	**84.95**	**13.59**	**26.72**	**84.72**

Table 3: Comparison with the state-of-the-art methods on NIV dataset for prediction horizon $T \in \{3, 4\}$. † indicates using the visual state as supervision. The bold-faced and underlined numbers indicate the highest and the second-highest performance, respectively.

Methods	$T = 3$			$T = 4$		
	SR↑	mAcc↑	mIoU↑	SR↑	mAcc↑	mIoU↑
Random	2.21	4.07	6.09	1.12	2.73	5.84
DDN [7]†	18.41	32.54	56.56	15.97	27.09	53.84
Ext-GAIL [5]†	22.11	42.20	65.93	19.91	36.31	53.84
P3IV [53]	<u>24.68</u>	<u>49.01</u>	<u>74.29</u>	<u>20.14</u>	<u>38.36</u>	<u>67.29</u>
Ours	**26.05**	**51.24**	**75.81**	**21.37**	**41.96**	**74.90**

We also conduct experiments on the COIN and NIV datasets. The results are reported in Table 2 and Table 3. Our E3P outperforms all previous methods on both datasets. On the COIN dataset, the performance of our model far exceeds the latest state-of-the-art P3IV [53] by 4.17% ($T = 3$) and 2.27% ($T = 4$) in terms of Success Rate (SR). On the NIV dataset, our model improves the performance by up to 1.37% ($T = 3$) and 1.23% ($T = 4$) over the state-of-the-art method [53] in terms of SR. The consistent results on

Table 4: Comparison with the state-of-the-art methods on CrossTask dataset for different prediction horizon $T \in \{3, 4, 5, 6\}$. † indicates using visual states as supervision. The bold-faced and underlined numbers indicate the highest and the second-highest performance, respectively. The results are evaluated on Success Rate (SR).

Methods	$T = 3$	$T = 4$	$T = 5$	$T = 6$
Retrival-Based	8.05	3.95	2.40	1.10
DDN [7]†	12.18	5.97	3.10	1.20
P3IV [53]	<u>23.34</u>	<u>13.40</u>	<u>7.21</u>	<u>4.40</u>
Ours	**26.40**	**16.49**	**8.96**	**5.76**

Table 5: Ablation study of our method on the CrossTask dataset for prediction horizon $T \in \{3, 4\}$ in terms of Success Rate (SR) and mean Accuracy (mAcc).

Model	PFE	EPG	ARM	$T = 3$		$T = 4$	
				SR↑	mAcc↑	SR↑	mAcc↑
baseline				22.56	46.17	12.97	43.97
+ PFE	✓			23.55	48.33	13.53	45.20
+ EPG	✓	✓		25.62	52.28	14.85	47.44
Full	✓	✓	✓	26.40	53.02	16.49	48.00
Full w/o EPG	✓		✓	25.25	52.59	14.23	47.61

all three datasets demonstrate the effectiveness of our E3P, which attributes to our novel event-guided paradigm.

We further verify the effectiveness of our method for different prediction horizons on the CrossTask dataset. The results are reported in Table 4, our model shows significant improvement compared with P3IV [53] in the more difficult long-time horizon prediction, *i.e.*, 1.75% ($T = 5$) and 1.36% ($T = 6$) in terms of Success Rate (SR).

4.4. Ablation Study

Effect of main components in E3P. In Table 5, we analyze the effect of each component of our proposed method. Following the P3IV [53], we apply an action classifier on top of the backbone as our baseline, using action text labels as supervision. By adopting the Prompting-based Feature Extractor (PFE), we obtain an improvement over the baseline. Then, by introducing Event-aware Prompt Generator (EPG), our model obtains significant improvements, *e.g.*, 2.07% when $T = 3$ and 1.32% when $T = 4$ in terms of SR, which demonstrates the effectiveness of our proposed event-guided paradigm. Noteworthy, we evaluate the performance of event classification based the start and goal visual states and find that the event of procedure can generally be inferred from the observed states (*e.g.*, the event classification accuracy is 99.5% when $T = 3$). Then, by introducing the Action Relation Mining (ARM) module, our model obtains further performance improvement, *e.g.*, 0.78% when $T = 3$ and 1.64% when $T = 4$ in terms of SR, which demonstrates the effectiveness of mining action associations within each event. In addition, if the Event-aware Prompt Generator is removed from the full E3P model, the performance drops but still achieves state-of-the-art. We mainly attribute this to the proposed ARM module, as there are strong associations between actions even without knowing the event. In summary, the ablation study demonstrates the effectiveness of the proposed event-guided model.

Table 6: Quantitative analysis of DropRelation regularization for prediction horizon $T \in \{3, 4, 5\}$ on CrossTask dataset in terms of Success Rate (SR).

Prediction Horizon	Drop Rate (%)					
	0	5	10	20	30	40
$T = 3$	26.04	26.40	26.33	26.26	25.87	25.76
$T = 4$	15.69	15.87	16.18	16.49	16.12	15.81
$T = 5$	7.95	8.23	8.54	8.77	8.96	8.68

Table 7: Effect of the pre-trained text Model (*i.e.*, CLIP) on CrossTask dataset for prediction horizon $T \in \{3, 4\}$. SR and mIoU indicate Success Rate (SR) and mean Accuracy (mAcc), respectively.

Model	$T = 3$		$T = 4$	
	SR↑	mAcc↑	SR↑	mAcc↑
Full	26.40	53.02	16.49	48.00
w/o CLIP	25.83	52.39	14.70	46.18
w/o CLIP & w/o EPG	24.67	49.88	13.93	44.33

Analysis of DropRelation Regularization. In Table 6, we conduct a quantitative analysis of the DropRelation regularization. In general, for all prediction horizons (*i.e.*, $T \in \{3, 4, 5\}$), the performance of our method follows a trend of first increasing and then decreasing. Compared with a short prediction horizon (*e.g.*, $T = 3$), the best drop rate is higher for a longer prediction horizon (*e.g.*, $T = 5$). This is because the longer prediction horizon requires modeling more diverse action relations, and a relatively large drop rate ensures that our Action Relation Mining module adequately covers different combinations of the action tokens during the training, thus enabling the model to adequately consider the action associations. A drop rate of 20% is a recommended choice for all prediction horizons.

Effect of the Pre-trained Text Model. In our method, we use the pre-trained CLIP for the text representation extraction. For a fair comparison, we report the result of a variant of our E3P. Specifically, we replace the handcrafted text prompt (*i.e.*, based on CLIP) with learnable tokens (the same as P3IV [53]) and use one-hot action and order labels as supervision. As shown in Table 7, even if without the CLIP (*i.e.*, w/o CLIP), our method still achieves the state-of-the-art performance, *i.e.*, 25.83% when $T = 3$ and 14.70% when $T = 4$ in terms of Success Rate (SR). Furthermore, we remove the Event-aware

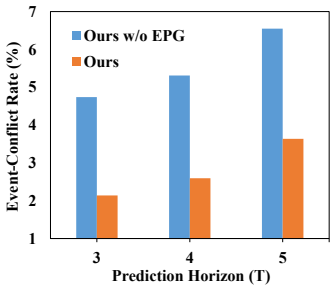

Figure 5: Analysis by event-conflict rate (*i.e.*, the proportion of event-conflict procedures) on the CrossTask dataset.

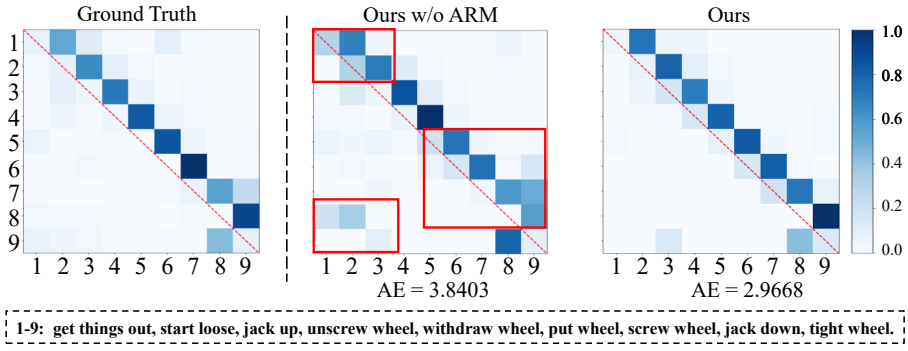

Figure 6: Visualizations of the action transition matrix for the event "Change a Tire" on the CrossTask dataset. The *i-row-j-column* depicts the probability of the transition from *i*-th action to *j*-th action and there are nine actions in total. Darker color indicates higher probability. Best viewed in color.

Prompt Generator from the above variant (*i.e.*, w/o CLIP & w/o EPG), our method still outperforms the latest state-of-the-art P3IV [53] (*i.e.*, 1.33% and 0.53% on SR of $T = 3$ and $T = 4$). These results demonstrate the effectiveness of our approach again.

4.5. More Analysis

Quantitative Analysis by Event-Conflict Rate. Here, we verify the effectiveness of our Event-aware Prompt Generator (EPG) by calculating an event-conflict rate, *i.e.*, the proportion of event-conflict procedure in all predicted procedures. An event-conflict procedure is defined as a procedure with actions that belong to different events. As shown in Figure 5, our full model predicts procedures with fewer event-conflicts, compared to our model without the Event-aware Prompt Generator (*i.e.*, Ours w/o EPG). These results demonstrate our event-guided paradigm helps exclude some impossible actions in transforming a start state to the goal state, bridging the semantic gap.

Analysis by the Action Transition Matrix. To verify the effectiveness of our Action Relation Mining (ARM) module, we compute the ground truth action transition matrix and the action transition matrix learned by a model. The i-row-j-column element in the transition matrix depicts the probability of the transition from i-th action to j-th action (*i.e.*, two successive actions). For an intuitive comparison, we first focus on the action transition within an event "Change a Tire" and visualize the transition matrix. As shown in Figure 6, the action transition matrix learned by our full model is more consistent with the Ground Truth, compared with our E3P without Action Relation Mining ("Ours w/o ARM"). The results demonstrate the effectiveness of our proposed "mask-and-predict" approach for relation mining.

For a quantitative analysis, we introduce a quantitative metric, namely Absolute Error (AE), to measure the difference between the learned transition matrix and the corresponding ground truth. The AE is defined as $AE = \sum_{i=1}^{n}\sum_{j=1}^{n}|F_{ij} - E_{ij}|$, where n is the number of actions, F is the learned matrix and E is the ground truth matrix. As shown in Figure 6, our full model achieves a lower AE (2.96) compared with "Ours w/o ARM" (3.84) in the "Change a Tire" event. Furthermore, we calculate the mean Absolute Error (mAE) to measure the difference of action matrices for all events. As shown in Table 8, our full model achieves a much lower mAE compared with "Ours w/o ARM", which attributes to our proposed relation mining scheme.

Table 8: Quantitative analysis of the learned transition matrix on CrossTask dataset in terms of mean Absolute Error (mAE) for prediction horizon $T = 4$.

Method	Ours	Ours w/o ARM	P3IV [53]
mAE ↓	2.55	3.21	3.56

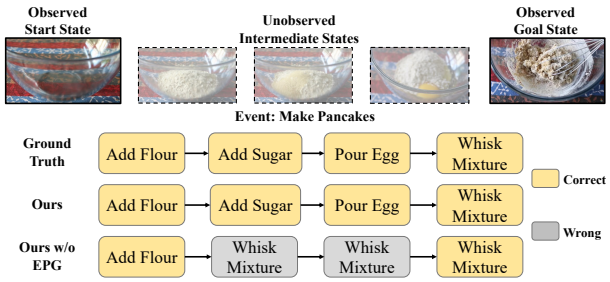

Figure 7: Quantitative analysis of the Event-aware Prompt Generator (EPG). In this figure, we take a four-action procedure planning as an example. Best viewed in color.

Qualitative analysis of the Event-aware Prompt Generator (EPG). In Figure 7, we give an example to demonstrate the effect of our EPG. Due to the semantic gap between the observed start-goal states and unobserved inter-

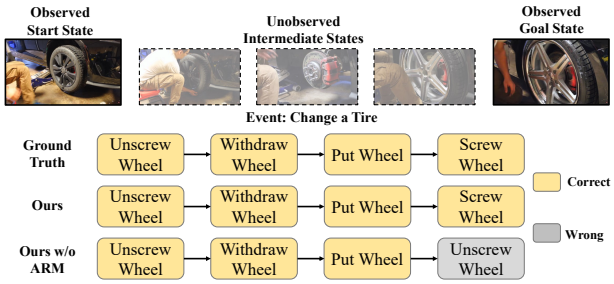

Figure 8: Quantitative analysis of the Action Relation Mining (ARM) module. In this figure, we take a four-action procedure as an example. Best viewed in color.

mediate actions, it is hard to predict "Add Sugar" and "Pour Egg" without Event-aware Prompt Generator, as shown by "Ours w/o EPG". In contrast, our full model captures the necessity of adding some sugar and eggs to make a fluffy pancake, which shows the effectiveness of our event-guided paradigm.

Qualitative Analysis of the Action Relation Mining. In Figure 8, we give an example to demonstrate the effect of our ARM. We find that "Ours w/o ARM" incorrectly predicts the "Screw Wheel" as "Unscrew Wheel", since it does not capture the action association that "Screw Wheel" should follow "Put Wheel". In contrast, our full model predicts the correct procedure, which shows the importance of modeling action relations and the effectiveness of our proposed ARM.

5. Conclusion

In this work, we propose a novel event-guided paradigm to bridge the semantic gap between the observed visual states and unobserved intermediate actions, aiming at solving procedure planning from instructional videos with text supervision. Based on the paradigm, we proposed an Event-guided Prompting-based Procedure Planning (E3P) model, which encodes event information into the sequential modeling to support procedure planning. A mask-and-predict approach is adopted to fully consider the strong action associations within each event. Extensive experiments on three datasets demonstrate the effectiveness of our event-guided paradigm, and our E3P achieves a new state-of-the-art performance. One limitation of both previous works and our work is that, they perform poorly when being evaluated on events that do not belong to the training set (*i.e.*, Success Rate $< 1\%$), the future effort could be devoted to such cross-event procedure planning from instructional videos.

Acknowledgement. This work was supported partially by the NSFC (U21A20471,U1911401,U1811461,62206315), Guangdong NSF Project (No. 2023B1515040025, 2020B1515120085), Fundamental Research Funds for the Central Universities, SYSU (23ptpy112), China Postdoctoral SF (2022M713574)

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *CVPR*, 2018. 3

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 5

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *IJRR*, 2020. 1

[4] Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. *NeurIPS*, 2013. 3

[5] Jing Bi, Jiebo Luo, and Chenliang Xu. Procedure planning in instructional videos via contextual modeling and model-based policy learning. In *ICCV*, 2021. 1, 2, 6

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2

[7] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *ECCV*, 2020. 1, 2, 5, 6, 7

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*, 2014. 1

[9] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020. 2

[11] Kiana Ehsani, Hessam Bagherinezhad, Joseph Redmon, Roozbeh Mottaghi, and Ali Farhadi. Who let the dogs out? modeling dog behavior from visual data. In *CVPR*, 2018. 6

[12] Yazan Abu Farha and Juergen Gall. Uncertainty-aware anticipation of activities. In *ICCV*, 2019. 3, 6

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019. 2

[14] Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future transformer for long-term action anticipation. In *CVPR*, 2022. 3

[15] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 2012. 1

[16] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[18] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*, 2012. 3

[19] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022. 3

[20] Hirokatsu Kataoka, Yudai Miyashita, Masaki Hayashi, Kenji Iwata, and Yutaka Satoh. Recognition of transitional action for short-term action prediction using discriminative temporal cnn feature. In *BMVC*, 2016. 3

[21] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *CVPR*, 2019. 3

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 6

[23] 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 *CVPR*, 2022. 3

[24] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 2

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 2023. 3

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[27] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 2001. 1

[28] 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 *CVPR*, 2020. 2, 6

[29] 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. 6

[30] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *ECCV*, 2022. 3

[31] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, 2021. 2

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[33] 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 supervision. In *ICML*, 2021. 3

[34] Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv*, 2020. 3

[35] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. In *AAAI*, 2020. 2

[36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NeurIPS*, 2014. 2

[37] Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *ICML*, 2018. 6

[38] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *RA-L*, 2022. 1, 2, 5

[39] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[40] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE access*, 2017. 2

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv*, 2017. 1

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 1

[43] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *TIT*, 1967. 5

[44] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 3

[45] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013. 3

[46] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv*, 2021. 3

[47] Sida Wang and Christopher Manning. Fast dropout training. In *ICML*, 2013. 3

[48] Markus Wulfmeier, Dominic Zeng Wang, and Ingmar Posner. Watch this: Scalable cost-function learning for path planning in urban environments. In *IROS*, 2016. 1

[49] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv*, 2017. 2

[50] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 2

[51] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. Token shift transformer for video classification. In *ACM MM*, 2021. 2

[52] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *ICCV*, 2021. 2

[53] 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 *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[54] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 5