TIDM: Temporal Interpolation Diffusion Model for Procedure Planning

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we study the problem of procedure planning in instructional videos, which involves making goal-directed plans based on current visual observations in unstructured real-life videos. Prior research has approached this as a distribution fitting problem, utilizing diffusion models to represent the entire sequence of actions, thereby transforming the planning challenge into one of sampling from this distribution. Building on this foundation, we introduce a novel approach by incorporating temporal diffusion model, where the temporal interpolation expands the previously non-existent temporal logical relationships. In terms of details, we employ an interpolating predictor to guide the intermediate process within U-Net, using the start and end frames as inputs. This involves extracting potential features through an encoder and applying an interpolation strategy to derive potential features for the intermediate frames. Furthermore, to make sure the accuracy pf actions in outputs, we also add mask strategy both in inference and loss calculation. Results across these three datasets of varying scales demonstrate that our TIDM model achieves state-of-the-art performance on several key metrics. The code and trained model are available at https://www.example.com.

1 Introduction

In recent years, the computer vision community has increasingly focused on the role of instructional video comprehension for model training [xxxxxx]. Instructional videos are Strong knowledge carriers, which contain rich scene variations and various actions. Learning procedural knowledge from instructional videos may be a natural ability for humans, but it is a major challenge for AI. This task requires AI to not only understand dynamic and complex scenes, but also to effectively segment and recognize key events, accurately identify and predict actions, and perform in-depth causal reasoning [xx,xx,xx]. Constructing such models can analyze complex human behaviors and provide effective assistance in problematic situations with clear goal orientation, such as home assistants, autonomous driving, and medical assistance tasks [xxxxxx]. To address the program planning problem in instructional videos, Han,Wu [PDPP] et al. proposed a solution to the PDPP model, which achieved good results through xxx, we we followed this work and made improvements, specifically xxxx, as shown in Fig. 1.

Previous instructional videos on program planning methods have typically treated it as a sequence planning problem, and most of the early work on program planning relied on two-branch autoregressive methods for stepwise prediction of intermediate states and actions [xx, xx, xx],along with the use of different network architectures for modeling probabilistic processes. A limitation of these methods is related to the autoregressive process, which is slow and subject to error propagation, and is prone to accumulate errors during complex planning. In contrast, zhao (to change the example) et al. proposed a single-branch non-autoregressive model that predicts intermediate steps in parallel, achieving good performance. However, this approach involves a complex training process of multiple loss functions to manage large design spaces. In contrast, we take inspiration from the work of [StoryDiffusion: Consistent Self-Attention for Long-Range Image and Video Generation] and others by using an interpolating predictor that uses the start and end frames as inputs, uses an encoder to extract the potential features of the intermediate frames to guide the intermediate process of Unet using the interpolation strategy, meanwhile we used LLM to guide the logical order of actions in order to enhance the temporal logic between actions, and achieved good results.

In this study, we propose an LLM-based mask interpolator that uses mask as the action mask, which limits the action range generation and directly restricts the search space for action prediction. Meanwhile, we use an interpolating predictor to extract potential features using encoder by using start and end frames as inputs, and utilize an interpolation strategy to get the potential features of intermediate frames to guide the intermediate process of Unet. At the same time, considering the possible adverse effects of inaccurate classification of task combinations in order to enhance the temporal logic between actions, we use use LLM to guide the logical order of actions. Finally, we performed experimental validation on three different types of datasets and obtained good results.

The main contributions of this paper are as follows:

1. An LLM-based mask interpolator While inheriting the basic model of PDPP, this interpolator absorbs the experience of Mask Diffusion and innovatively uses masks to limit the range of actions, thus improving the accuracy of generated actions.
2. Using interpolated predictors and latent feature extraction methods The model generation is enhanced by inputting start and end frames, extracting potential features using encoder, and generating potential features for intermediate frames through interpolation strategy to further guide the intermediate process of Unet.
3. Introduction of LLM to enhance the temporal logic of actions In this paper, LLM is uniquely used to guide the logical sequence of actions, ensuring that the generated actions are rational and consistent in the time series.
4. Experimental validation was performed on different types of datasetsIn this paper, we have conducted extensive experimental validation on CrossTask, COIN, and NIV datasets, and the results show that the method proposed in this paper significantly improves the model performance on several tasks.

The remainder of the paper is organized as follows: Section 2 presents some preliminary knowledge; Section 3 contains the main results; Section 4 demonstrates the practicality of the results; Section 5 provides numerical examples; Section 6 presents the proofs; and finally, Section 7 concludes the paper.

2 Related Work

Procedural video understanding.The issue of program video comprehension is a growing concern.In [20](,Zhao et al. 2022),Zhao et al proposes a weakly supervised probabilistic procedure planning method for learning execution procedures from instructional videos. The method uses a Transformer-based model with a memory module to map the start and goal states to a sequence of possible actions.In [21](PDPP),Wang at all proposed PDPP method which effectively solves the process planning problem in instructional videos, and requires only task labels as supervision, avoiding reliance on intermediate visual or verbal annotations.In this paper, we follow this work by studying the problem of procedural video comprehension through learning goal-directed action planning and we improved the previous work by using LM-based mask interpolator to limit the range of actions, thus improving the accuracy of generated actions.

Diffusion probabilistic models. Nowadays, many reasearchers proved that utilizing diffusion models to represent the entire sequence of actions, thereby transforming the planning challenge into one of sampling have achieved great success[11,22,33,44].Further more, diffusion probabilistic models have achieved great success in many research areas,such as image generation and editing[111],speech generation and processing[222],text generation [333] and many other domains.In this work, we apply diffusion process to procedure planning in instructional videos.Additionally, we employ an interpolating predictor to guide the intermediate process within Unet, using the start and end frames as inputs. This involves extracting potential features through an encoder and applying an interpolation strategy to derive potential features for the intermediate frames.

Visual representation learningRecently,the latest models usually use knowledge from the language domain (e.g., wikiHow) as distant supervision signals (Zhong et al. 2023; Zhou et al. 2023; Lin et al. 2022). However, the computational cost of training/fine-tuning large VLMs is usually prohibitively high. Alternatively, efforts have also been made to use pretrained large language models (LLMs) as a visual planner (Patel et al. 2023; Wang et al. 2023b), leveraging the zero- shot reasoning ability of powerful foundation models (Ge et al. 2023; Kim et al. 2022; OpenAI 2023; Touvron et al. 2023). However, significant performance gaps remain due to lack of domain knowledge.In this paper,we introduce LLM to guide the logical sequence of actions, ensuring that the generated actions are rational and consistent in the time series.

need a figure

3 Method

In this section, we present the details of our projected diffusion model for procedure planning (PDPP). We will first introduce the setup for this problem in Sec. 3.1. Then we present the diffusion model used to model the action sequence distribution in Sec. 3.2. To provide more precise conditional guidance both for the training and sampling process, a simple projection method is applied to our model, which we will discuss in Sec. 3.3. Finally, we show the training scheme (Sec. 3.4) and sampling process (Sec. 3.5) of our PDPP. An overview of PDPP is provided in Fig. 2.

3.1 Problem formulation

We follow the problem set-up of (PDPP),Wang at all:given an initial visual observation o_s and a target visual state o_g, the model is tasked with generating a sequence of actions $a_{1:T}$ that transforms the environment state from o_s to o_g. Here, T represents the planning horizon, indicating the number of action steps the model must take, while $\{o_s, o_g\}$ denotes two distinct states of the environment as shown in an instructional video.

We break down the procedure planning problem into two distinct sub-problems, as illustrated in Eq. (1). The first sub-problem involves learning task-related information c from the given pair $\{o_s, o_g\}$. This step serves as an initial inference in the procedure planning process. The second sub-problem focuses on generating action sequences using the obtained task-related information and the given observations. It's important to note that Jing et al. [2] also decompose the procedure planning problem into two parts; however, their first sub-problem aims to provide long-horizon information for the second stage since they plan actions sequentially. In contrast, our approach seeks to establish conditions for sampling to facilitate more efficient learning.

$$p\left(a_{1:T} \mid o_s, o_g\right) = \int p\left(a_{1:T} \mid o_s, o_g, c\right) p\left(c \mid o_s, o_g\right) \, dc \tag{1}$$

During the training phase, we begin by training a basic model, represented by multi-layer perceptrons (MLPs), using the provided observations $\{o_s, o_g\}$ to predict the task category. The task labels from the instructional videos, denoted as c, are used to supervise the model's output. Following this, we evaluate $p(a_{1:T} \mid o_s, o_g, c)$ in parallel with our model and utilize the ground truth (GT) intermediate action labels as supervision for training. Unlike the visual and language supervision used in previous studies, task label supervision is easier to obtain and results in a simpler learning process. During the inference phase, we use the initial and goal observations to predict the task class information c, and then sample action sequences $a_{1:T}$ from the learned distribution based on the given observations and the predicted c, where T represents the planning horizon.

need figure 3

3.2 Projected diffusion for procedure planning

Our approach is composed of two main stages: predicting the task class and modeling the distribution of action sequences. The first stage involves solving a standard classification problem, which we address using a simple MLP model. The core component of our approach is the second stage, which focuses on modeling $p(a_{1:T} \mid o_s, o_g, c)$ to effectively handle the procedure planning task. In contrast to Jing et al. [2], who frame this as a Goal-conditioned Markov Decision Process and use a policy $p(a_t \mid o_t)$ alongside a transition model $\tau_\mu(o_t \mid c, o_{t-1}, a_{t-1})$ to plan actions incrementally—an approach that is complex to train and slow during inference—we treat this problem as one of direct distribution fitting using a diffusion model.

Diffusion model.A diffusion model [19, 29] addresses the data generation challenge by representing the data distribution $p(x_0)$ as a denoising Markov chain over a sequence of variables $\{x_N, \ldots, x_0\}$, where x_N is assumed to follow a random Gaussian distribution. In the forward process, Gaussian noise is gradually added to the initial data x_0, which can be described by $q(x_n \mid x_{n-1})$. This allows for the generation of all intermediate noisy latent variables $x_{1:N}$ across N diffusion steps. During the sampling phase, the diffusion model iteratively performs a denoising procedure $p(x_{n-1} \mid x_n)$ over N iterations to approximate samples from the target data distribution. The forward and reverse processes of diffusion are depicted in Fig. 3.

In a typical diffusion model, the proportion of Gaussian noise introduced to the data at each diffusion step n is predetermined and denoted by $\{\beta_n \in (0,1)\}_{n=1}^N$. Each step of adding noise can be parameterized as:

$$x_n \mid x_{n-1} \sim \mathcal{N}\left(\sqrt{1-\beta_n}\, x_{n-1}, \beta_n \mathbf{I}\right). \tag{2}$$

Since the hyper-parameters $\{\beta_n\}_{n=1}^N$ are predefined, the noise-adding process does not involve any training. Based on the discussion in [19], we can re-parameterize Eq. (2) to obtain:

$$x_n = \sqrt{\bar{\alpha}_n}\, x_0 + \sqrt{1-\bar{\alpha}_n}\, \epsilon \quad (1) \tag{3}$$

where $\bar{\alpha}_n = \prod_{s=1}^n (1-\beta_s)$, and $\epsilon \sim \mathcal{N}(0, I)$.

In the denoising process, each step is parameterized as:

$$p_\theta\left(x_{n-1} \mid x_n\right) = \mathcal{N}\left(x_{n-1}; \mu_\theta\left(x_n, n\right), \Sigma_\theta\left(x_n, n\right)\right). \quad (1) \tag{4}$$

3.3 Footnotes

Indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).[2]

3.4 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction; art work should not be hand-drawn. The figure number and caption always appear after the figure. Place one line space before the figure caption, and one line space after the figure. The figure caption is lower case (except for first word and proper nouns); figures are numbered consecutively.

Make sure the figure caption does not get separated from the figure. Leave sufficient space to avoid splitting the figure and figure caption.

You may use color figures. However, it is best for the figure captions and the paper body to make sense if the paper is printed either in black/white or in color.

[1] Sample of the first footnote

[2] Sample of the second footnote

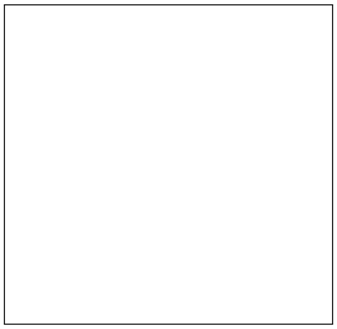

Figure 1: Sample figure caption.

Table 1: Sample table title

PART	DESCRIPTION
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

3.5 Tables

All tables must be centered, neat, clean and legible. Do not use hand-drawn tables. The table number and title always appear before the table. See Table ??.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

4 Conclusion

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

A Appendix

You may include other additional sections here.