Procedure Planning in Instructional Videos
via Contextual Modeling and Model-based Policy Learning

Jing Bi   Jiebo Luo   Chenliang Xu

University of Rochester

jing.bi@rochester.edu   jiel@cs.rochester.edu   chenliang.xu@rochester.edu

Abstract

Learning new skills by observing humans' behaviors is an essential capability of AI. In this work, we leverage instructional videos to study humans' decision-making processes, focusing on learning a model to plan goal-directed actions in real-life videos. In contrast to conventional action recognition, goal-directed actions are based on expectations of their outcomes requiring causal knowledge of potential consequences of actions. Thus, integrating the environment structure with goals is critical for solving this task. Previous works learn a single world model will fail to distinguish various tasks, resulting in an ambiguous latent space; planning through it will gradually neglect the desired outcomes since the global information of the future goal degrades quickly as the procedure evolves. We address these limitations with a new formulation of procedure planning and propose novel algorithms to model human behaviors through Bayesian Inference and model-based Imitation Learning. Experiments conducted on real-world instructional videos show that our method can achieve state-of-the-art performance in reaching the indicated goals. Furthermore, the learned contextual information presents interesting features for planning in a latent space.

1. Introduction

Humans can learn new skills by watching demo videos. Although this seems natural to human, it is challenging for AI. We have seen rich works on modeling human behaviors from videos with the majority focusing on recognizing actions [21, 18, 26]. However, solely perceiving what actions are performed without modeling the underlying decision-making process is insufficient for AI to learn new skills. The next-generation AI needs to figure out what actions are necessary to achieve the desired goals [5] with the consideration of actions' potential consequences. In this paper, we focus on learning the goal-directed actions from instructional videos. Recently, Chang *et al*. [6] proposed a new problem known as procedure planning in instructional videos.
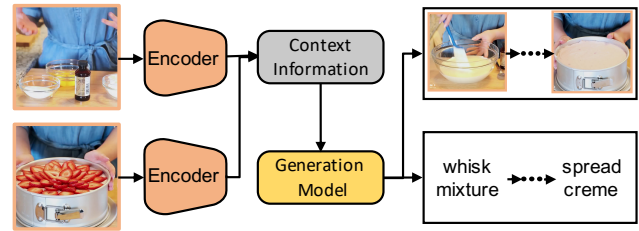

Figure 1: **Overview of our proposed method.** Given a starting observation (top-left image) and a desired visual goal (bottom-left image), we extract the contextual information of the planning trajectory upon which the Generation Model outputs a sequence of actions. The model is responsible for learning plannable latent representations with a focus on procedures and action consequences. Thus, we can retrieve images of intermediate steps (top-right images).

It requires a model to 1) plan a sequence of verb-argument actions and 2) retrieve the intermediate steps for achieving a given visual goal in real-life tasks such as making a strawberry cake (see Fig. 1). This task is different from the typical image-language translation problem in the way that certain actions can be exchanged to achieve the same goal (*e.g*., the order of *adding salt* and *sugar* usually does not matter), making it difficult to predict the same action sequence as ground-truth using sequence mapping.

Moreover, sequence-to-sequence based structure, suitable for modeling events that tend to occur in sequence with high probability, is thought to involve no consideration of the likely outcome [9]. Therefore, we formalize this task as a planning problem with focus on two different sequential patterns that can be easily observed in Fig. 4: In the context of *making a cake*, *mixing ingredients* and *washing cherries* are interchangeable, *i.e*., short-term action separation, but both should be ahead of the action *putting cherries on the top*, *i.e*., long-term action association.

Inspired by Raab *et al*. [9], we think that when performing goal-directed tasks, it is beneficial to consider both the task contextual information and the potential action consequences. Contextual information here refers to the time-invariant knowledge (not changed during planning) that dis-
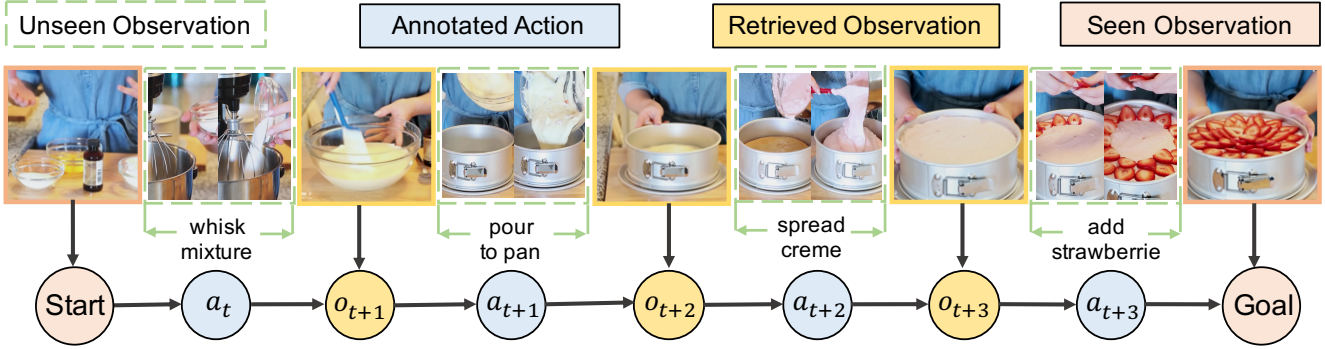
Figure 2: **Procedure planning example.** Given a starting observation (picture of food ingredients) and a visual goal (picture of a made cake), the model needs to learn how to complete real-world tasks such as making a cake by planning a sequence of actions $a_{1:T}$ (blue circles) and retrieving the intermediate observations $o_{2:T-1}$ (yellow circles).

tinguishes a particular task from the others. For example, if we know the goal is to *make a cake* as shown in Fig. 1, it is less likely to plan an action like *putting it on the grill*.

Therefore, we model the dependency between the actions and different goals as the long-term action association in a Bayesian framework. As we show later in the experiments, this serves a few purposes: a) it provides a more structured representation for the subsequent policy learning; b) we can sample from the posterior distribution for more diverse trajectories to facilitate the action exploration; and c) compared with the noisy pixel space, feature distances in the learned latent space are more meaningful. To achieve short-term action separation, we model the action sequence as a Markov Decision Process (MDP) as shown in Fig. 2, where the future action depends only upon the present state. Besides, because goal-directed actions are often selected based on expectations of their consequent outcomes [7], we propose to incorporate a transition model into the Imitation Learning (IL) framework [16, 19] so that we can explicitly model the environment jointly with policy learning.

This approach brings following advantages: a) it helps policy to flexibly pursue a goal by leveraging causal knowledge of the actions' potential consequences; b) when model-based simulations produce states with alternative actions, the discrimination and selection between actions allow an agent to find the currently most desired outcome [24, 34]; and c) it bypasses the need of an interactive environment that is required by classic planning algorithms [31, 30], making it suitable for modelling the web videos.

We demonstrate the effectiveness of our approach by evaluating it on a real-world instructional video dataset [43] (an example is shown in Fig. 2). The results on the procedure planning task show that our learned model can uncover the underlying human decision-making processes. Furthermore, the results on the challenging walk-through planning task [22] confirm that our model learns meaningful representations of the environment dynamics, which is crucial for efficient plannings in the latent space. Finally, the vi-

sualization of contextual information indicates that our proposed encoder structure can learn a concise representation to capture distinct knowledge of different real-world tasks. The main contributions of our work are summarized as follows: a) we propose a novel method to address the procedure planning problem, which combines Bayesian Inference with Model-based Imitation Learning; b) we propose a neural network structure based on variational inference that learns to embed sufficient information to convey the desired task, incorporating the visual observations' uncertainty; and c) we propose two model-based IL algorithms that explicitly learn the environment dynamics (in either a stochastic or deterministic way) and integrate with the transition model to simultaneously learn a plannable latent representation for accurate planning.

2. Related Work

Vision-based Human Behavior Understanding. Our planning tasks are highly related to a popular AI research area: building a machine that can accurately understand humans' actions and intentions. The intention can be seen as the sequence of actions needed to be taken to achieve an objective [5]. To understand human attention, Zhang *et al.* [41] proposed the Deep Future Gaze model to predict the gaze location in multiple future frames conditioned on the current frame. Forthermore, Wei *et al.* [38] utilized a hierarchical graph that jointly models attention of the gaze and intention of the performing task from a RGB-D video. Rhinehart *et al.* [30] proposed an online inverse reinforcement learning method to discover rewards for modeling and forecasting first-person camera wearer's long-term goals, together with locations and transitions from streaming data. Merel *et al.* [27] extended the Generative Adversarial Imitation Learning (GAIL) [16] framework to learn human-like movement patterns from demonstrations consisting of only partial observations. Unlike these previous works that predict the future, we try to understand human behaviors by learning their goal-direction actions.

Deep Reinforcement Learning. Reinforcement learning (RL) is often employed to learn and infer the MDP model simultaneously, which is a natural way to understand how humans learn to optimize their behaviors in an environment [35]. Recently, combined with deep learning, DRL is leveraged to solve several vision problems such as Visual Tracking [33], Video Summarization [42], Stroke-based rendering [18], and Vision-based Navigation [28]. For semantic-level video understanding, DRL can also play an important role. For instance, it is utilized for Activity Localization [36], Natural Language Grounding [13], and Video Description [37]. However, these works often require expert knowledge to design a useful reward function, whose goal is to learn a behavior that maximizes the expected reward. In contrast, we work on the IL problem without explicit usage of a hand-craft reward. Our work is most closely related to inverse RL [30, 1] and contextual RL[39]. However, the major difference is that we focus on learning from collected dataset which is crucial for applications when online interaction is not permitted, *e.g.*, safety-critical situation.

Planning in Latent Space. Planning is a natural and powerful approach to decision-making with known dynamics, such as game playing and simulated robot control. To plan in unknown environments, the agent needs to learn the environment dynamics from previous experiences. Recent model-based RL schemes have shown promise that deep networks can learn a transition model directly from low-dimensional observations and plan with the learned model [40, 6, 11]. A closely related method is Universal Planning Networks (UPN) [32] that learns a plannable latent space with gradient descent by minimizing an imitation loss, i.e., learned from an expert planner. Plannable means the learned representations are structured to perform a classic planning algorithm [22]. Our method further incorporate the contextual knowledge of assigned task to the latent space and remove the assumption of differentiable action space. Another line of work is causal InfoGAN [22], which tries to capture the relations between two sequential images and models the causality of the simulation environment in an unsupervised learning manner. Similarly, our Ext-MGAIL model also focus on the stochastic transition model. However, making predictions in raw sensory space is unnecessarily hard [10], we predict low dimensional latent representations for future state and plan upon it.

3. Methods

We consider a set-up similar to Chang *et al.* [6]: we have access to K trajectories $\{(o_{1:T}^j, a_{1:T}^j)\}_{j=0}^K \sim \pi_E$ collected by an expert trying to achieve different tasks. Given a starting visual observation o_1 and a visual goal o_T that indicates for a particular task, we want to learn a plannable representation upon which goal-directed actions are planned to perform two complex planning tasks (Fig. 2): a) proce-

dure planning: generate a valid sequence of actions $a_{1:T}$ to achieve the indicated goal; and b) walk-through planning: retrieve the intermediate observations $o_{2:T-1}$ between the starting o_1 and the goal. Our key insight is that by decomposing the procedure planning problem in Eq. 1 into two sub-problems, we can decouple representation learning into two parts: a) inferring the time-invariant contextual information that conveys the task to achieve; and b) learning the time-varying plannable representations related to the decision-making process and environment dynamics. In this way, both representations can be further used to retrieve $o_{2:T-1}$ for solving the walk-through planning.

As shown in the overall architecture Fig. 3, we assume that the contextual information contains all the details an agent need for achieving the desired goal. Hence, we formulate the procedure planning problem $p(a_{1:T}|o_1, o_T)$ as:

$$p(a_{1:T}|o_1, o_T) = \iint p(a_{1:T}, s_{1:T}|z_c)p(z_c|o_1, o_T)\mathrm{d}s_{1:T}\mathrm{d}z_c, \tag{1}$$

where we donate z_c as the context variable that conveys the desired task, $p(z_c|o_1, o_T)$ as the inference model for modeling posterior distribution over the context variable and given observations and $p(a_{1:T}, s_{1:T}|z_c)$ as the generation model that plans a sequence of actions and hidden states that transfer the initial state to the desired outcome. In the following sections, we will first discuss how to infer the contextual information. We will then solve the second sub-problem by imitating human behaviors with consideration of offline policy evaluation [23], and utilize Hindsight Experience Replay (HER) [3] to better leverage the expert demonstrations. Lastly, we will discuss how to solve the walk-through planning problem with learned model.

3.1. Inference Model

As visualized in Fig. 3 the action a_t at time-step t is solely governed by the current state s_t that contains the information of current observation and the information regarding the desired goal. We want z_c to represent the contextual information for achieving the goal, which should be time-invariant, and the hidden state s_t to contain the time-varying information for the decision-making process. To achieve this separation, the hidden states are only allowed to condition on z_c; thus, all the information about the goal must pass through z_c to avoid a shortcut from the observations to actions. In this way, s_t will be the only time-dependent hidden variable used to recover the actions, and we reserve z_c for compressing everything else. However, the true posterior distribution $p_\phi(z_c|o_1, o_T)$ from video frames is analytically intractable; thus, we use variational inference to approximate posterior distribution from given observations. Note here we use the raw pixel observations o_1 and o_T for planning, which is different from compared methods which use pre-computed visual features.
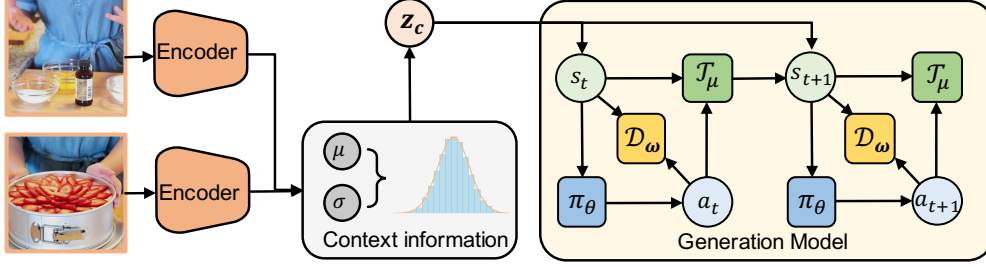
Figure 3: **The overall architecture:** given the initial and the goal observations, two parallel encoders will parameterize the mean and log-variance of the Gaussian distribution. The context variable will then be sampled from this distribution and fed into the generation model to roll out a trajectory. We use a discriminator that tries to distinguish the state-action pairs from the expert or the learned policy, which serves as the local reward function.

The model involves an encoder (shown in Fig. 3) that models the approximation distribution $q_\varphi(z_c|o_1, o_t)$ and a decoder (omitted from Fig. 3 for simplicity) that models the prior $p_\phi(o_1, o_t|z_c)$. It can be seen as a two-head Variational Auto-Encoder (VAE) [20] with one head encoding o_1 and the other for o_T, and we call it a predictive VAE. We jointly optimize ϕ and φ by maximizing the evidence lower bound:

$$l(\phi, \varphi) = \mathbb{E}_{q_\varphi}[\log p_\phi(o_1, o_T|z_c)] - \text{KL}(q_\varphi(z_c|o_1, o_T)||p(z_c)), \quad (2)$$

where we assume $p(z_c)$ is a Gaussian prior parameterized by the context variable. By training in this way, the encoder $q_\varphi(z_c|o_1, o_t)$ is enforced to learn a compact representation from the given observations, o_1 and o_T, to convey the desired task, which serves as the contextual information.

3.2. Generation Model

After inferring the context variable, the remaining question is: how to model $p(a_{1:T}, s_{1:T}|z_c)$ to solve the planning problem? We assume the underlying process in Fig. 2 is a fully observable Goal-conditioned Markov Decision Process $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{C})$, where $\mathcal{S}, \mathcal{A}$ is the state and action space. We denote $p(a_t|s_t)$ as policy π_θ and $p(s_t|z_c, s_{t-1}, a_{t-1})$ as transition model $\mathcal{T}_\mu$. In this way, the generative model $p(a_{1:T}, s_{1:T}|z_c)$ can be factorized as:

$$p(a_{1:T}, s_{1:T}|z_c) = \prod_{t=1}^{T} \pi_\theta(a_t|s_t)\mathcal{T}_\mu(s_t|z_c, s_{t-1}, a_{t-1}), \quad (3)$$

where we use the convention that $s_0, a_0 = 0$.

A popular way to solve the MDP problem is using RL algorithms. However, we only have access to expert trajectories $\{(o_{1:T}^j, a_{1:T}^j)\}_{j=0}^{K} \sim \pi_E$ without a well-defined reward function, making it infeasible to directly apply RL algorithms. Therefore, we adopt an IL approach and use the expert trajectories as demonstrations. However, there are still several key difficulties: a) typical IL algorithm is model-free algorithm that is ideal for learning *habitual* behavior without thought for actions' consequences [24], making it imperfect to learn goal-directed actions. b) the static dataset

cannot provide feedback signals or the transferred states as the learning agent interacts with it; and c) each demonstration trajectory is performed by the expert to reach a specific goal and thus might not be sufficiently explored under different situations. Below, we address these difficulties.

Effective Imitating with Transition Model. Instead of short-term environment learning as in [6], we optimize model with the whole trajectory. Inspired by GAIL [16], we formulate the IL problem as an occupancy measure matching problem [17], where the goal is to minimize the Jenson-Shanon divergence of trajectory distributions induced by the learned policy π_θ and the expert policy π_E respectively. In order to learn goal-directed actions, and bypass the need of an interactive environment which is required the original GAIL, we employ a transition model to roll out and jointly optimize it with policy learning. There are two important reasons for the joint optimization: a) during the training, the action policy is not stationary, which means a pre-trained transition model will not help the action policy explore better decisions; and b) the transition model can interact with action policy so that the learned latent space is optimized on the entire state-action pairs induced by the expert policy π_E, which helps it incorporate information over multiple time steps. The model can be either deterministic or stochastic; thus, we introduce two versions of the transition model.

Int-MGAIL: In Interior-Model GAIL, the transition model is built inside the LSTM cell, which can be seen as a fully deterministic model. We modify the LSTM cell and treat the long-term cell state as the state s_t in Eq. 3 and the short-term hidden state as our action a_t, so that we can enforce the action a) to interact with the hidden state to roll out the next state b) only depends on the current state. At each time-step, the input to the cell is the previous cell's long-term and short-term state s_{t-1} and a_{t-1}, as shown in Eq. 4.

$$
\begin{aligned}
f_t &= \sigma(W_f * a_t + U_f * s_t + b_f), \\
i_t &= \sigma(W_i * a_t + U_i * s_t + b_i), \\
a_t &= \text{Tanh}(W_a * s_t + b_a), \\
s_{t+1} &= f_t * [s_t, z_c] + i_t * a_t.
\end{aligned}
\quad (4)
$$

Ext-MGAIL: Int-MGAIL provided a deterministic solution for modeling the unknown environment, but it will underestimate the uncertainty of the environment. Therefore, we further take the transition model as an external module to explicitly model the environment transition in a stochastic way, meaning different observations can follow the same state. To model the uncertainty, we designed the action policy as a stochastic model with Bernoulli probability vector of a_t because of the discrete action space. The stochastic modeling is crucial for successful planning when we have the same start and goal states but different procedures.

Hindsight Relabeling. The problem with the static dataset is that each episode only shows one possible way to reach the specified goal, which limits the agent's ability to explore what would happen had the circumstance been different. Inspired by HER [3], we utilized the relabeling method that tries to alleviate this problem by augmenting the demonstrations with "fake" goals that were attained in the episode, allowing the agent to sufficiently explore the state-action space and make better decisions for the future. Formally, we have one valid trajectory $\{(o_{1:T}^j, a_{1:T}^j)\}$ of an expert attempting to reach the goal o_T at jth episode from the start. Then, the portion of this trajectory between any two non-adjacent observations, o_m and o_n, can also be seen as a valid trajectory as the expert attempts to reach o_n starting from o_m. Therefore, for every trajectory in the original dataset, we select two non-adjacent observations and augment the dataset with $D \leftarrow D \cup (o_{m:n}, a_{m:n})$. The intuition behind this process is that we can replay each episode with a different goal than the one the expert was initially trying to achieve.

3.3. Learning

We have three main components to be optimized: a) the transition model $\mathcal{T}_\mu(s_{t+1}|s_t, a_t, z_c)$ that uses previous state-action pair and context variable to predict the next state; b) the policy model $\pi_\theta(a_t|s_t)$ that models the distribution over the set of action under current state; and c) the discriminator $\mathcal{D}_\omega$, parameterized by ω, tries to distinguish the $\{(s_t, a_t)\}$ from the expert or the learned policy π_θ.

We refer the expert trajectory as $\tau^E = \{(s_t^E, a_t^E)\}$ and trajectory $\tau = \{(s_t, a_t)\}$ as state-action pairs visited by the current learned policy. We first randomly sample τ^E from the dataset and roll out τ^E accordingly, then we optimize the discriminator by ascending the gradient in Eq. 5:

$$\mathbb{E}_{\pi_\theta}[\nabla_\omega \log(1 - \mathcal{D}_\omega(s_t, a_t))] + \mathbb{E}_{\pi_E}[\nabla_\omega \log(\mathcal{D}_\omega(s_t^E, a_t^E))] . \tag{5}$$

We further let the discriminator gradient back-propagate into the previous time-step, helping the transition model to learn the further consequences related to current action. However, we observe a high variance problem during the training. Therefore, we employ an additional loss to help the generated states s_t quickly move to regions close to the

---

Algorithm 1 Walk-through Planning

Input: All observations $\{o_i\}_{i=1}^N$, set of action $\{a_i\}_{i=1}^M$, models $\mathcal{T}_\mu, \pi_\theta$ planning length T

1: Initialize observation list $\beta \leftarrow \emptyset$
2: **for** $i = 1, 2, \cdots, N$ **do**
3: $\quad s_i = \phi(o_i)$
4: **for** $i = 1, 2, \cdots, N$ **do**
5: $\quad s_{\text{next}} = \mathcal{T}_\mu(s_i, \pi_\theta(s_i))$
6: $\quad \triangleright$ Find the index of the nearest state
7: $\quad k = \arg\min_k \|s_k - s_{\text{next}}\|_2^2$
8: $\quad \triangleright$ Increment the transition probability over all action
9: $\quad S_{i,k} \mathrel{+}= \sum_{m=0}^M \pi_\theta(a_m|s_i)$
10: $\beta \leftarrow \arg\max_{\rho \in Perm(T)} \sum_{i=1}^T S_{i,\rho(i)}$

---

expert-visited states. Hence, the transition model is optimized by descending the gradient in Eq. 6:

$$\mathbb{E}_{a_t \sim \pi_\theta}[\nabla_\mu \log(1 - \mathcal{D}_\omega(\mathcal{T}_\mu(s_{t-1}, a_{t-1}, z_c), a_t^E))] + \mathbb{E}_{s_t^E \sim \pi^E}[\nabla_\mu \mathcal{L}(\mathcal{T}_\mu(s_{t-1}, a_{t-1}, z_c), s_t^E)] , \tag{6}$$

where $\mathcal{L}$ measures the distance between two latent vectors.

The last component is the action policy π_θ. After optimizing the discriminator, we can interpreted it as a local reward function and we optimize policy to maximum the reward $r(s_t, a_t) = log(D_\omega(s_j, a_j))$. In order to imitate the expert rather than mimicking, the action policy needs the ability to intentionally explore actions that the expert did not perform. We adopted the offline policy evaluation in offline-RL and follow the classical evaluation method [12, 8], re-weighting the rewards by the importance sampling ratio (Eq. 7) to select a better policy during training. Concretely, we first learn a classification network as the behavior policy $\beta(a_t|s_t)$ from demonstrations via behavioral cloning. Then optimize π_θ with policy gradient which tries to maximize the accumulated reward along the whole trajectory:

$$\mathbb{E}_\beta[\frac{\pi_\theta(a_t|s_t)}{\beta(a_t|s_t)} \nabla_\theta \log \pi_\theta(a_t|s_t) Q(s_t, a_t)] - \lambda \mathcal{H}(\pi_\theta) , \tag{7}$$

where $\mathcal{H}(\pi_\theta) = \mathbb{E}_{\pi_\theta}[-\log \pi_\theta(a|s)]$ is the policy entropy.

3.4. Walk-through Planning with Transition Model

Given the start and goal observations, we first infer the contextual information by sampling from $q_\varphi(z_c|o_1, o_T)$. Based on the sampled z_c, the generation model will roll out subsequent actions and hidden states as the sampled trajectory. Given the pool of visual observations $\{o_i\}$, we first construct the score matrix $S_{i,j}$ to capture the transition probability between o_i and o_j with the sampled trajectory,

as shown in Alg. 1. After constructing the rank score table, we can then perform walk-through planning to retrieve the intermediate observations that lead to the goal. As suggested in [6], this problem can be seen as finding a permutation function $b : \{1, 2, \cdots, T\} \rightarrow \{1, 2, ..., T\}$ that maximizes the transition probability along the permutation path, subject to the constraints that $b(1) = 1, b(T) = T$.

4. Experiments

We choose CrossTask [43] to conduct our experiments, which consists of 2,750 video (212 hours in total). Each video depicts one of the 18 primary long-horizon tasks such as *Grill Steak* or *Make French Strawberry Cake*. For the videos in each task, we randomly select 70% for training and 30% for testing. Different tasks have various procedure steps: less complex tasks include *jack up a car* (3 steps); more complex ones include *pickle cucumbers or change tire* (11 steps), and the steps do not necessarily appear in the same order as the task description as shown in Fig. 4.

Each video has densely annotated boundaries with caption labels that describe the person's actions in the video. We treat each video as a sequence of images $I_{1:N}$ having annotated description $v_{1:M}$ with temporal boundaries $(s_{1:M}, e_{1:M})$. For i-th video clip, we choose frames around the beginning of the captions $I_{s_i-\delta:s_i+\delta}$ as o_i, caption description v_i as the semantic meaning of action, and images nearby the end $I_{e_i-\delta:e_i+\delta}$ as the next observation o_{i+1}. Here, δ controls the duration of each observation, and we set $\delta = 1$ for all experiments. We further use the relabeling technique introduced in Section 3.2 to augment the data with randomly selected 30% of the expert trajectories.

To construct our state-space $\mathcal{S}$, we use pre-computed features provided in CrossTask as our state estimation: One second of the video is encoded into a 3,200-dimensional feature vector which is a concatenation of the I3D, Resnet-152, and audio VGG features [15, 14, 4]. Note here we do not use the state estimations for testing; we only use them for training the Generation model. Lastly, we construct the action space $\mathcal{A}$ by enumerating all combinations of the caption description's predicates and objects, which provides 105 action labels and are shared across all 18 tasks. **Implementation Details.** For computing context variables, we use the DCGAN architecture [29] as the image encoder and decoder in our model. The behavior policy is a classification network that takes state estimation as input and generates the probability over the action space. The policy network for both Int-MGAIL and Ext-MGAIL share a similar structure as the off-policy Actor-Critic network [8], which is two-headed: one for computing an action based on a state and another one producing the expected return values of the action. In the Ext-MGAIL, we assume our hidden state s_t to be Gaussian; thus, the transition model is Gaussian with mean and variance parameterized by a feed-forward neu-
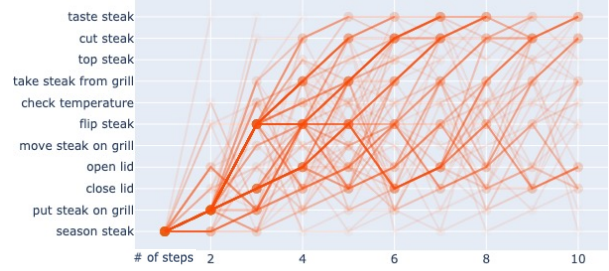

Figure 4: **Expert trajectories of *Grill Steak* task.** Heavier color indicates more frequently visited path

ral network. The discriminator networks for both models share the same architecture, which is a similar network in the original GAIL [16]. Further implementation details can be found in the supplementary material.

4.1. Evaluating Procedure Planning

We compared with the following methods:
- **Uniform Policy.** At each step, the algorithm will uniformly sample one action from all actions. This method serves as the empirical lower bound of performance.
- **Universal Planning Networks (UPN)** [32]. Like our method, UPN learned a plannable latent representation where gradient descent can be used to compute a plan that minimizes a supervised imitation loss. We expand the original UPN to discrete action space by using a softmax layer to output probability over discrete actions.
- **Dual Dynamics Networks (DDN)** [6]. DDN is the first work proposing procedure planning in instructional video problem. Similar to UPN, it learns the dual dynamic of the state-action transition and perform sample-based planning upon the learned latent representation.

When evaluating with the pre-collected dataset, a common way is the re-weighted rewards [23]. But there is no defined rewards here. To keep consistent with state-of-the-art methods, we use three different matrices to evaluate the performance and limit the experiment to length 3-5 even our method is applicable for longer trajectory modeling.
- **Success Rate** is designed to evaluate the long-term action association, *i.e.*, the correctness of *action sequence*. Only if every action matches, this plan is considered as a success. The result is the percentage of the successes in test dataset.
- **Accuracy** is used to evaluate the correctness of *individual time step action*, which serves as a constraint relaxation of the success rate metric. The individual action is considered as a success only if it matches the ground truth at the same time step, which is written in terms of percentages.
- **mIoU** is used to capture the cases where the model can output the right actions but *fail to preserve the actions' order*. We adopt this metric from [6] that computes IoU $\frac{|\{a_t\} \cap \{\hat{a}_t\}|}{|\{a_t\} \cup \{\hat{a}_t\}|}$ between the set of ground-truth $\{a_t\}$ and the planned actions $\{\hat{a}_t\}$.

Table 1: **Results of Procedure Planning.** Our models significantly outperform the baselines by $\sim 10\%$ improvement in terms of the success rate. Our Ext-MGAIL has a marginal improvement compared with Int-MGAIL; this shows that introducing a stochastic process in the policy can help the policy explore and thus improve the performance.

		Uniform	UPN	DDN	Int	Ext
T=3	Succ.	0.01	2.89	12.18	17.03	**21.27**
	Acc.	0.94	24.39	31.29	44.66	**49.46**
	mIoU	1.66	31.56	47.48	58.08	**61.70**
T=4	Succ.	0.01	1.19	5.97	9.47	**16.41**
	Acc	0.83	21.59	27.10	37.16	**43.05**
	mIoU	1.66	28.85	48.46	57.24	**60.93**

As illustrated in Table 1, UPN can learn representations that perform reasonably well compared to the uniform baseline. However, as instructional videos' action space is not continuous, the gradient-based planner cannot work well. The proposed Int-MGAIL outperforms baseline DDN at two different time-scales. The reason is that we perform RL training, which maximizes the accumulated reward alone the whole trajectory. By introducing the stochastic process into the action policy, our Ext-MGAIL has a better performance. This is because, given the same beginning and goal observation, there is more than one valid sequence of actions. By designing a model with both stochastic and deterministic components, we show that our agent successfully learns plannable representations from real-world videos to outperform all the baseline approaches on all metrics.

Figure 5: **Walk-through Planning qualitative results for *Grill Steak*.** Given the starting and goal observations, our model can output the correct order for each step.

4.2. Evaluating Walk-through Planning

Different from procedure planning, given the the observations o_1 o_T, the model needs to generate the intermediate observations $\{o_2, \cdots, o_T\}$. Directly predicting the raw RGB image is unnecessarily hard [10]. Similar to the setup in [6], instead of generating frames, we retrieve the intermediate video clips from dataset in the correct order. To find the path from o_1 to o_T, a rank table is constructed to evaluate the transition probability between two clips.

Table 2: **Results of Walk-through Planning.** Our model outperforms the baselines by explicitly modeling the transition dynamics between temporally adjacent observations.

		Uniform	UPN	DDN	Int	Ext
T=3	Ham.	1.06	0.57	0.26	0.19	**0.13**
	Pacc.	46.85	71.55	86.81	86.98	**93.66**
T=4	Ham.	1.36	1.36	0.88	0.70	**0.57**
	Pacc.	52.23	68.41	81.21	86.42	**89.74**

In addition to the **Uniform** policy and **DDN**, we include the **Causal InfoGAN (CIGAN)** into comparison. Like our approach, they plan the trajectory in latent space but use the generative model to transform the trajectory to observations directly. The advantage of CIGAN is that it can be trained to perform walk-through planning without action supervision.

The evaluations are conducted on the following metrics.
- Hamming. As described earlier, we are finding the best permutation of the observation index. Then the distance is defined as $d(y, \hat{y}) = \sum_{i=1}^{T} I(1|y_i \neq \hat{y}_i)$, which is good for evaluating the *single step observation order*.
- Pair Accuracy. To compare the distance between two permutation sequence, we use pairwise accuracy to calculate the distance along the planned and ground truth observation orders. This is defined as $\frac{2}{T(T-1)} \sum_{i<j, i \neq j}^{T} I(1|y_i < \hat{y}_j)$.

The results are shown in Table 2 and Fig. 5. CIGAN can learn reasonable models beyond Uniform without using action supervision. However, the complexity of the instructional videos requires explicit modeling of the forward dynamics conditioned on the semantic actions. Our two methods outperform all baseline models, which means both models are applicable to both planning and walk-through planning. We also show that effective IL requires learning a transition model and optimizing policy on multi-step transitions instead of individual state-action pairs.

4.3. Visualization of Contextual Information

In this section, we aim to answer the following two questions: a) Can the proposed Inference model learn the useful contextual information of different tasks from the demonstrations? b) Why does the contextual information help the subsequent action learning? To this end, we use t-SNE [25] to reduce the dimension of the context variable z_c to 2 and visualize z_c all 18 tasks as shown in Fig. 6. For every task, we randomly sampled 100 pairs of start and goal clips (1800 pairs in total) and extracted their contextual information.

As we can see, all the samples are grouped by the tasks' labels, which suggests the Inference model has learned roughly distinct regions in the hidden space to correspond to each task in the dataset. Further task descriptions can be found in the supplementary material. Considering we

never used the task labels in the learning process, this result indicates an underlying relationship between the different tasks' observations. This suggests that the Inference model can encode sufficient information to convey the desired task. Furthermore, the Generation model can benefit from this concise embedding on modeling decision-making because irrelevant deviations from the raw pixel space will create exponentially diverging trajectories.

However, there still exist overlaps between different clusters, and some samples drift from the majority. This reflects the fact that we cannot entirely rely on the contextual information to recover the expert's decision process, which further validates the effectiveness of the Generation model.

4.4. Ablation Study

We conducted experiments with three variations: **w/o r:** The model is trained with both sequence mapping loss and discriminator loss but w/o maximizing accumulated reward. In this way, the model tries to match the short-term actions w/o considering the trajectory as whole. **w/o dis:** We further drops the discriminator loss in Eq. 6, making it a supervised seq2seq learning model. As shown in Table 3, learning the model solely w/o dis will significantly hurt the overall performance because the model simply optimized on expert trajectory will be over-fitted to the regions that the expert traversed and thus make it hard to generalize to other areas that might be helpful for action policy learning. We think the main reason for the low success rates with sequence mapping is that pure supervised learning will excessively focus on recovering the same action sequence as ground-truth, ignoring the fact that actions can be exchanged to achieve the same goal. **w/o HER:** We observed that combining HER always brings performance boosts. Our stochastic $Ext_{w/o\ HER}$ has a lower performance than Int-MGAIL; we suspect that it is because, without HER, the original dataset is insufficient for stochastic model optimizing the whole trajectory. This result is consistent with our observation on the experiment with the additional dataset.

T=3	Int w/o r	Int w/o dis	Ext w/o r	Ext w/o dis	Int w/o HER	Ext w/o HER
Succ.	7.18	5.89	15.18	11.42	14.39	18.01
Acc.	18.74	11.66	27.29	23.46	37.43	43.86
mIoU	27.51	20.66	37.48	30.97	54.18	57.16

Table 3: **Results of Ablation Study.** The performance decreases significantly when optimizing without sequence modeling or only with a supervised learning loss, which shows the importance of learning over the whole trajectory.

4.5. Addtional Experiment

We further compare our model with previous best performing method DDN on a second dataset [2]. Both of our models outperformed the DDN, but the increase of
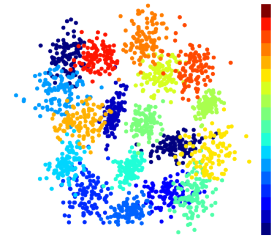

Figure 6: **Visualization of the contextual information** of the starting and goal observations of all 18 tasks in the CrossTask dataset. The colorbar indicates the ground-truth task labels, where different colors indicate different tasks.

the accuracy on procedure planning (Succ.∼4%, T=3) is smaller than one obtained on the CrossTask, and the performance between Int-MGAIL and Ext-MGAIL are very similar (Succ. 20.19/22.11%, T=3). We suspect that it is because the new dataset does not provide sufficient samples for optimizing long trajectories. More details can be found in the supplementary material.

5. Conclusion, Application, and Future Work

In this paper, we present a new method to address the procedure planning problem focusing on learning goal-directed actions. Concretely, we propose a predictive VAE structure that learns to embed the contextual information of the desired task. Moreover, we propose two novel model-based imitation-learning algorithms to solve the formulated decision-making problem in unknown environments. Results on real-world instructional videos show that our approach can learn a meaningful representation for planning and uncover the human decision-making process.

Being able to learn goal-directed actions from the pixels, the proposed method enables the AI system to extract useful information from expert demonstrations. Moreover, learning policy from the offline dataset avoids online interaction with the environment, making our method practical in real-world applications, *e.g.*, service robots.

A direction of future work is investigating different ways to combine contextual information with environment dynamics, such as through self-supervision. Another important future direction is to consider the collected data as policy constraints such that robots can act safely in the real world and continuously improve themselves by accumulating data of the environment interactions, making robotic agents more capable of solving challenging real-life tasks.

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004. 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. 8

[3] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in neural information processing systems*, pages 5048–5058, 2017. 3, 5

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6

[5] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956, 2018. 1, 2

[6] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 3, 4, 6, 7

[7] Nathaniel D Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12):1704–1711, 2005. 2

[8] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012. 5, 6

[9] Raab .et al. Ch. 13. the development of goal-directed decision-making. In *Goal-Directed Decision Making*. Academic Press, 2018. 1

[10] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016. 3, 7

[11] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565, 2019. 3

[12] Josiah Hanna, Scott Niekum, and Peter Stone. Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, pages 2605–2613. PMLR, 2019. 5

[13] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8393–8400, 2019. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[15] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017. 6

[16] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4565–4573. Curran Associates, Inc., 2016. 2, 4, 6

[17] Jonathan Ho, Jayesh Gupta, and Stefano Ermon. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning*, pages 2760–2769, 2016. 4

[18] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8709–8718, 2019. 1, 3

[19] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019. 2

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[21] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*, 2018. 1

[22] Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. In *Advances in Neural Information Processing Systems*, pages 8733–8744, 2018. 2, 3

[23] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. 3, 6

[24] Mimi Liljeholm. Instrumental divergence and goal-directed choice. In *Goal-Directed Decision Making*, pages 27–48. Elsevier, 2018. 2, 4

[25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7

[26] Brais Martinez, Davide Modolo, Yuanjun Xiong, and Joseph Tighe. Action recognition with spatial-temporal discriminative filter banks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5482–5491, 2019. 1

[27] Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201*, 2017. 2

[28] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019. 3

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 6

[30] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017. 2, 3

[31] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017. 2

[32] Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks. *arXiv preprint arXiv:1804.00645*, 2018. 3, 6

[33] James Supancic III and Deva Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 322–331, 2017. 3

[34] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991. 2

[35] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 3

[36] Weining Wang, Yan Huang, and Liang Wang. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 334–343, 2019. 3

[37] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4213–4222, 2018. 3

[38] Ping Wei, Yang Liu, Tianmin Shu, Nanning Zheng, and Song-Chun Zhu. Where and why are they looking? jointly inferring human attention and intentions in complex tasks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6801–6809, 2018. 2

[39] Lantao Yu, Tianhe Yu, Chelsea Finn, and Stefano Ermon. Meta-inverse reinforcement learning with probabilistic context variables. *arXiv preprint arXiv:1909.09314*, 2019. 3

[40] Kuo-Hao Zeng, William B Shen, De-An Huang, Min Sun, and Juan Carlos Niebles. Visual forecasting by imitating dynamics in natural sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2999–3008, 2017. 3

[41] Mengmi Zhang, Keng Teck Ma, Joo Hwee Lim, Qi Zhao, and Jiashi Feng. Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[42] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3

[43] 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 Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. 2, 6