A Language First Approach to Procedure Planning

Anonymous ACL submission

Abstract

Procedure planning, or the ability to predict a series of steps that can achieve a given goal conditioned on the current observation, is critical for building intelligent embodied agents that can assist users in everyday tasks. Encouraged by the recent success of language models (LMs) for zero-shot (Huang et al., 2022a; Ahn et al., 2022) and few-shot planning (Micheli and Fleuret, 2021), we hypothesize that LMs may be equipped with stronger priors for planning compared to their visual counterparts. To this end, we propose a language-first procedure planning framework with modularized design: we first *align* the current and goal observations with corresponding steps and then use a pre-trained LM to *predict* the intermediate steps. Under this framework, we find that using an image captioning model for alignment can already match state-of-the-art performance and by designing a double retrieval model conditioned over current and goal observations jointly, we can achieve large improvements (19.2% - 98.9% relatively higher success rate than state-of-the-art) on both COIN (Tang et al., 2019) and CrossTask (Zhukov et al., 2019) benchmarks. Our work verifies the planning ability of LMs and demonstrates how LMs can serve as a powerful "reasoning engine" even when the input is provided in another modality.[1]

1 Introduction

Developing autonomous agents of versatility and flexibility requires the ability to produce plans on-the-fly for a given task based on observations of the current state. Procedure planning, as proposed by (Bi et al., 2021), tests whether an agent can predict the steps needed to bring a given initial state into a given goal state, where both states are specified with visual observations, as shown in Figure 1. Compared to planning in a closed-world with structured environments, procedure planning with instructional videos provides an unstructured, visually complex, and highly-detailed observation of the world (i.e., *visual observation space*, presented as video instances) while asking the model to predict high-level actions (i.e., *action space*, highlighted in the green box).

To handle such a mismatch between the observation space and the action space, previous methods (Bi et al., 2021; Chang et al., 2020) have focused on learning a *latent visual feature space* from visual observations that is more suitable for planning. However, learning the ideal latent space is challenging since visual observations can differ greatly due to changes in the background, actor, or tools, even for the same task. For example, the two observations in Figure 1 are highly dissimilar although they are part of the same task *making salad*. This makes it inherently difficult for models to *align* visual observations to high-level actions, not to mention *reason* and *predict* over multiple steps to produce a plan.

Meanwhile, pre-trained language models (LMs) show strong planning ability, as demonstrated by their excellent performance for zero-shot (Huang et al., 2022a) and few-shot text planning tasks (Micheli and Fleuret, 2021). This inspires us to think if planning in *text feature space* is a better alternative to planning in *visual feature space* used in prior work. Apart from the strong prior from language model pretraining, the actions in procedure planning have the dual representation of text and labels (Zhao et al., 2022), which makes text space more easily aligned with the action space, both of which are more abstract than visual observations.

While the idea of converting visual input into text and relying on language models has been effective in a series of multimodal tasks such as image captioning and visual question answering (VQA) (Zeng et al., 2022; Wang et al., 2022), the case is different for procedure planning as (1) proce-

[1] Our code is provided as part of the supplementary materials.

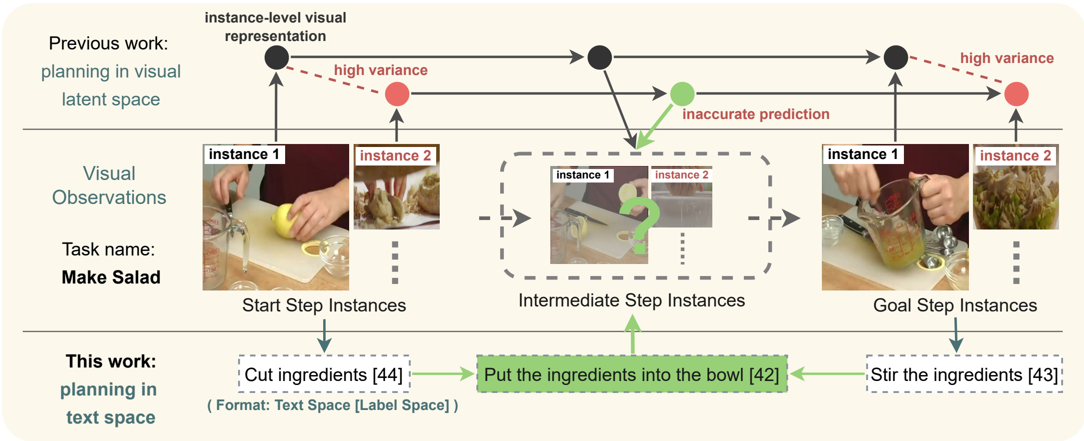

Figure 1: Overview of our language first approach for procedural planning. Previous work performs planning in the visual latent space, which can be difficult to learn due to the high variance of image features in the same step. We propose to perform planning in the existing language latent space, which is more generalized and robust compared to the visual variance.

dure planning was originally proposed as a vision-only task instead of being inherently multi-modal; (2) we attempt the transfer of the procedure reasoning and prediction ability of the LM instead of simply extracting information from the images. As shown in Figure 1, LM helps us predict the hardest intermediate steps (`Put the ingredients into the bowl`) which have little support from either start or end observations.

The major challenge of employing language models for procedure planning is how to map the start and goal observations into text space without losing salient information for planning. If the mapping is largely inaccurate, then even with the strong reasoning ability of LMs, it might not be worth the trouble of converting the problem into text space.

As the first exploration, we validate the effectiveness of a simple baseline model in our language-first planning framework, i.e., using image captioning to convert visual observations into text to prompt LMs. We find that by using image captioning we can already achieve performance comparable to state-of-the-art models. However, closer examination shows that image captioning is not sufficient to capture visual details across the current and goal observation (especially those related to movement and state change) and in turn does not effectively leverage the planning power of LMs.

Rooted in this observation, we propose to perform direct alignment from observations to steps by retrieving the most relevant step from the dataset-wide candidate step pool. Since visual observations can be highly diverse for the same step, for the modularized framework, we design a double retrieval model that jointly retrieves the first and the last steps corresponding to the start and goal observation respectively. Using both the visual observations (such as the video input of the start step and goal step in Figure 1) and the task name (such as *make salad*), we can further constrain the search space and identify the steps with higher accuracy.

Experiments on two benchmark datasets COIN (Tang et al., 2019) and Crosstask (Zhukov et al., 2019) show that our proposed language-first framework can improve procedure planning effectiveness under all settings. In particular, our best model, which represents each observation by a montage of multiple frames and utilizes the double retrieve model, achieves the best results and yields 19.2% - 98.9% relatively higher success rate than the state-of-the-art. This demonstrates the strong planning ability of pre-trained LMs and shows the potential of using LMs as a general "reasoning engine" or "planning engine", even in tasks where images are provided as input.

In summary, our contributions are as follows:

1. We verify the effectiveness of planning in text space compared to visual space by employing language models for procedure planning.

2. We design two models for adapting language models for procedure planning: an image captioning based baseline model performs explicit conversion to generate prompts and a modularized framework which split the prediction into two stages.

3. On two instructional video datasets COIN and Crosstask, we show that our proposed text space planning approach can significantly outperform prior methods, in certain cases doubling the plan success rate.

2 Related Work

Instructional Procedure Planning Introduced by (Chang et al., 2020), the procedure planning task aims at predicting the intermediate steps (actions) given a start visual observation and a goal visual observation. The key challenge of this task lies in its unstructured, highly diverse observations which are unsuitable for directly planning over. To tackle this challenge, most previous approaches (Bi et al., 2021; Chang et al., 2020; Srinivas et al., 2018; Sun et al., 2022) attempt to learn a latent space from visual observations by a supervised imitation learning objective over both the actions and the intermediate visual observations. More recently, P3IV(Zhao et al., 2022) observes that actions can be treated as both discrete labels and natural language. By using a pretrained vision-language model to encode the actions as text, P3IV achieves higher planning success rate using only action-level supervision. P3IV can be seen as an attempt to map the action text into visual space to provide more stable supervision. In comparison, our model maps visual observations into text space.

Pre-trained Language Models for Planning Recent work has shown the potential of language models for text-based planning tasks. Language models pre-trained on a large internet-scale corpus encodes rich semantic knowledge about the world and are equipped with strong low-shot reasoning abilities. In the effort of connecting language models with embodied AI, pioneering work on text-based planning (Côté et al., 2018; Shridhar et al., 2020; Micheli and Fleuret, 2021) shows that learning to solve tasks using abstract language as a starting point can be more effective and generalizable than learning directly from embodied environments. More recently, (Ahn et al., 2022; Huang et al., 2022b; Yao et al., 2022; Huang et al., 2022a) further show that using large language models as out-of-the-box planners brings significant benefits to a wide range of embodied tasks, such as navigation and instruction following.

In this paper, we utilize language model's planning ability to solve cross-modal planning tasks. We finetune a pre-trained BART model (Lewis et al., 2019) as a planning expert.

3 Method

In this section, we introduce our language-first approach to procedure planning. We first investigate whether language models can be applied for the task of procedure planning using text-only input (Section 3.2). Building upon this model, we explore two different methods to map the visual observations to their corresponding steps.

In Section 3.3 we introduce our baseline model which incorporates a pre-trained image-captioning model and a language model to do procedure planning task. This baseline yields results comparable to the state-of-the-art approaches, we identified its deficiencies by giving examples.

In Section 3.4 we introduce our modularized framework which first utilizes a conditional double retrieval model to retrieve the most similar step for the start and goal visual observations jointly. Then the retrieved steps will be plugged into the language model to predict all the intermediate steps.

3.1 Task Formulation

As shown in Figure 1, given a current visual observation o_0, and a goal visual observation o_T, procedure planning requires the model to plan a sequence of actions $\{a_1, \cdots, a_T\}$ that can turn the current state into the goal state, where T is the planning horizon. Additionally, every task has an overall goal, or task name, g such as `Replace a lightbulb`.

During training, two types of supervision are available: visual supervision and action supervision. Visual supervision refers to the visual observations at each intermediate timestep $\{o_1, ..., o_T\}$. Action supervision refers to the corresponding action labels $\{a_1, ..., a_T\}$. In particular, a_i is the action that transforms the observed state from o_{i-1} into o_i. Each action can be interpreted as a discrete label (`Action 33`) or a short piece of text (`Remove the lampshade`). In this paper, we use the terms *action* and *step* interchangeably. Following P3IV (Zhao et al., 2022), in our work, we only use action supervision during training.

3.2 Text-Based Planning Model

Language models are trained with the self-supervised objective of recovering the original text given a partial or corrupted text sequence. To adapt language models for our use case where the out-

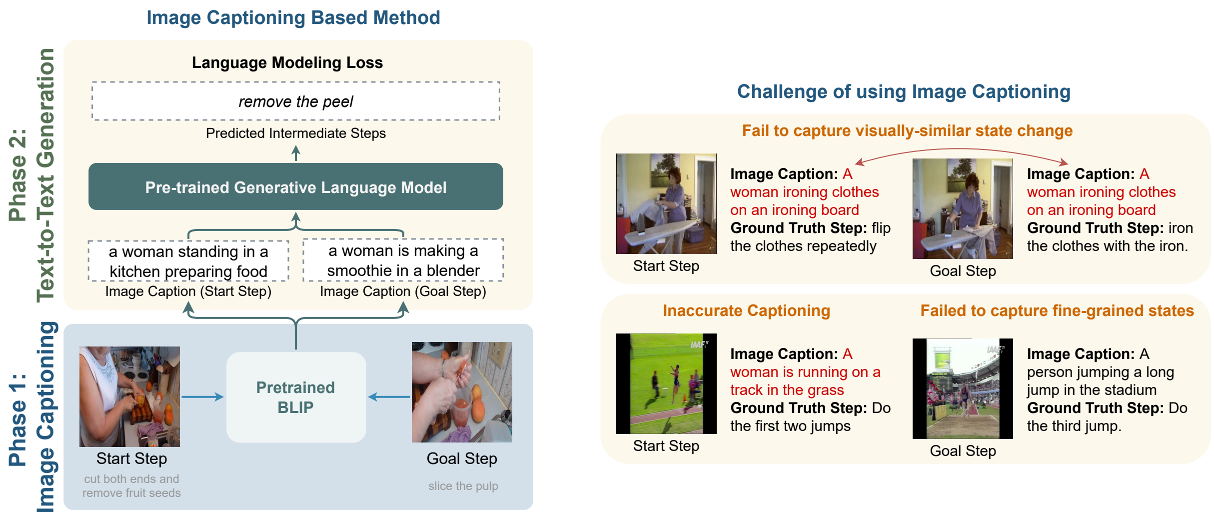

Figure 2: In the left we show the architecture of our language-first baseline model, which uses image captioning to transform images into the text space. In the right we show the example challenging cases for this approach: (a) the generated caption may not be able to capture fine-grained details of an image; (b) the generated caption can hardly relate to target steps/actions.

put action descriptions are of variable token length, we employ a pretrained encoder-decoder model BART (Lewis et al., 2019).

Assuming that we can perfectly map the input visual observations to actions, the input $\mathbf{x}$ to the BART model will be a prompt containing the task g, the first action a_1, the last action a_T, and the prediction horizon T. Here, the actions are interpreted as a short piece of text. The model will then be fine-tuned to sequentially predict all of tokens $a_i^1, \cdots, a_i^m$ that comprise each of the intermediate action descriptions a_i. This factorization allows us to train the language model using cross-entropy loss over each token a_i^j.

During inference, we face two challenges: (1) restricting the language model's output to the set of feasible actions and (2) allowing for diversity in the generated plans.

The first challenge is due to the fact that the language model predicts a distribution over the entire vocabulary at each decoding step, which makes the output domain essentially the space of all possible text strings. We experiment with two methods, namely *projection* and *constrained decoding*. In the projection method, similar to (Huang et al., 2022a), we first generate the entire action sequence using beam search and then for each predicted action, we project it to the most similar viable action based on SentenceBERT (Reimers and Gurevych, 2019), embedding cosine similarity between predicted steps and all the candidate steps. In the constrained decoding approach, we first construct a Trie of tokens using all of the viable actions. During decoding, we look up the Trie to check which tokens are valid and suppress the probability of the other tokens, effectively reducing the possible output space.

3.3 Baseline Model

A straightforward way to use LMs for procedure planning is to first convert the visual observations into text. We adopted a pre-trained image captioning model to do this. As shown in Figure 2, we first conduct image captioning for both the start and goal images. Then, the captions are converted into a prompt to be fed into a generative language model to predict the intermediate steps.

3.4 Modularized Framework

Our baseline model yields results comparable to state-of-the-art models. However, large amounts of inaccurate captions are found as shown in the right part of Figure 2. This leads to the design of our modularized model, where we first employ a pretrained vision-language model to align the visual observation to the most similar step, directly mapping it to the text space and label space.

We formulate the first step as a retrieval problem over all possible actions in the dataset. Initially, we tried to retrieve the start and goal actions independently conditioned on the corresponding observations:

$$\hat{a}_1 = f(o_0), \hat{a}_T = f(o_T) \quad (1)$$

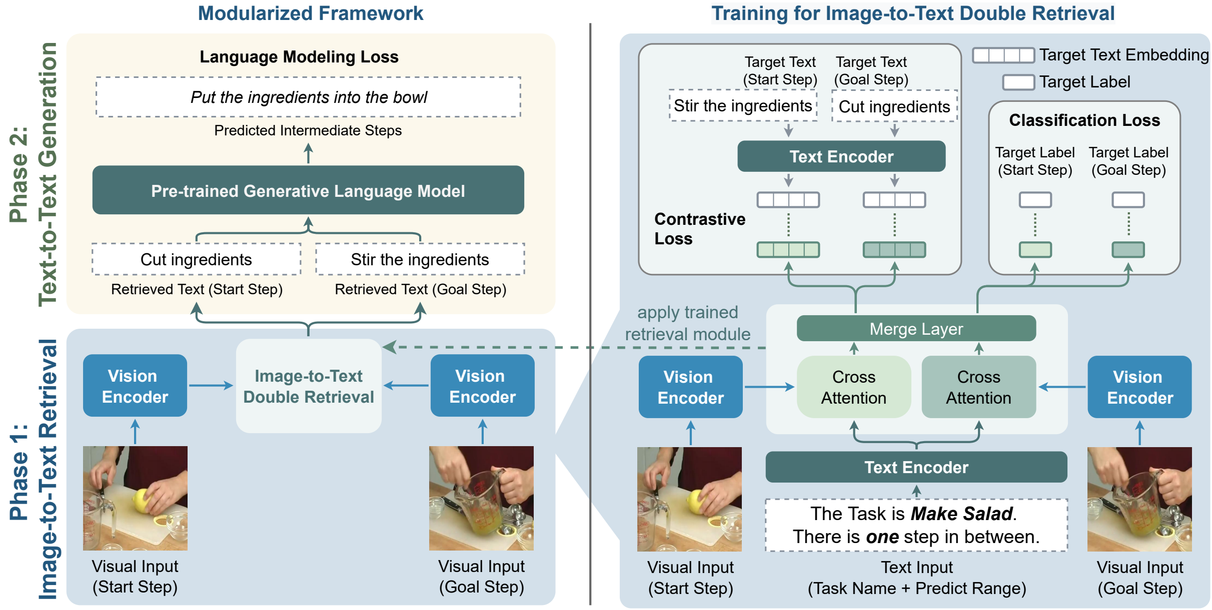

Figure 3: The architecture of our modularized framework. The right part is a double retrieval model, whose input includes both the start step and the end step (presented as images), as well as a textual prompt. The left side is based on a language model finetuned on ground truth steps, which is designed to predict the intermediate steps. By integrating these two models, we are able to perform procedure planning task.

However, the retrieval performance using an off-the-shelf vision-language model is far from satisfactory even after fine-tuning on our target dataset. This is due to the high visual variance within the same action class (same action can happen in different backgrounds and involving visually dissimilar objects) and relatively low visual variance within the same observation trajectory (frames of the same actor in the same environment).

Thus we propose to make the retrieval problem less ambiguous and more constrained by retrieving the start and goal actions jointly, namely the double retrieval model.

$$\hat{a}_1, \hat{a}_T = f(o_0, o_T) \tag{2}$$

An illustration of the model is shown in Figure 3.

Double retrieval input The input to the model is a pair of visual observations (o_0, o_T) and a text prompt specifying the task name d and the planning horizon T: `The task is` g `and there are` $T-2$ `steps in between.`

Vision-Language cross-attention model We use pre-trained BLIP (Li et al., 2022) as the basis for our retrieval model. The input observations and prompt are first encoded by the image encoder and text encoder respectively and then passed through a cross-attention module to model their interaction. Then, the fused representation for the start observation and the goal observation will be passed to a merging layer to combine the information from both images. This merging layer is implemented as a single linear projection which maps the concatenated features into 768 dimensions.For each of the observations, we use a classification head and a language embedding head to output the predicted action as a probability over a candidate set $p(\mathbf{a})$, and as a text embedding $\hat{h}$, respectively. The loss function is a combination of the cross-entropy action classification loss $\mathcal{L}_a$and the text embedding contrastive loss $\mathcal{L}_l$.

$$\mathcal{L}_a = -\sum_{i=0}^{N} a_i \log p(a_i) \tag{3}$$

$$\mathcal{L}_l = -\log \frac{\exp(l_i \cdot \hat{h})}{\sum_{j=0, j\neq i}^{N} \exp(l_j \cdot \hat{h})} \tag{4}$$

where N is the number of the valid actions in the dataset, l_i is the text embedding of the ground truth label for this instance and l_j are the text embeddings of all the other labels, which serve as negative examples.

4 Experiments

4.1 Experiment Setup

Datasets We evaluate on two mainstream datasets of instructional videos including COIN(Tang et al., 2019), CrossTask(Zhukov et al., 2019). COIN is a dataset containing 11827 videos with 180 different tasks and 46354 annotated segments. Following previous attempts (Zhao et al., 2022; Chang et al., 2020), we adopt the 70%/30% split to create our training and testing set. We use 20% of training data for validation.

We followed the data preprocessing steps of the procedure planning task(Chang et al., 2020) to select the start and goal visual observations, while at the same time, we also adopt a multi-frame dataset curation approach to boost our model's ability. Apart from the original approach of getting the start image / goal image of the video segment directly, we also use a uniform sampling of nine frames across the video and concatenate them into one single image to represent the visual observation. Details about our data pre-processing and parameter setting can be found in Appendix A We report the results of both methods in our ablation study Section 4.3.

Metrics Previous efforts regard the step prediction for procedure planning tasks as a classification task. Instead, we focus on generating each step with a language model. It is certainly possible for the language model to generate steps that have same meaning as the ground-truth steps but of different textual descriptions. For example, the language model may produce an output as "put all the bed boxes together" while the correct prediction is "put all bed boxes together". However, we only consider predictions that are identical to ground truth as successful. As a result of this evaluation protocol, we are able to use similar metrics as previous work to ensure our results comparable. Generally, our model will generate a sequence containing several steps. The sequence is separated by a separator "." to distinguish different steps. We use the first K steps as our final output for predictions that have more steps than we want. In the case of predictions with fewer steps than we would like, we regard the last few predictions as empty strings. The metrics that we adopt include:

- Success Rate (SR) considers a plan successful only if it exactly matches the ground truth.

Dataset	LM	Steps.	**SR**	**mAcc**	**mIoU**
COIN	BART	3	67.37	67.37	67.37
COIN	BART	4	35.43	51.12	62.89
Crosstask	BART	3	60.04	60.04	60.04
Crosstask	BART	4	33.27	48.28	61.37

Table 1: Finetuning intermediate steps on BART: For a given prediction horizon T, we show the prediction result (%) for the intermediate $T-2$ steps.

- Mean accuracy (mAcc) treats each step prediction independently, so the order of the predicted steps matters.

- Mean Intersection over Union (mIoU). In this evaluation, if one step is successfully predicted at anywhere in the procedure, this step will be considered as correct.

Baselines We adopt state-of-the-art models as baselines, including DDN (Chang et al., 2020), PlaTe (Sun et al., 2022), Ext-GAIL (Bi et al., 2021), P3IV (Zhao et al., 2022). As ablation studies, we include three variants of our proposed approach: "Ours(base)" uses single frames as model input and applies our image captioning baseline model; "Ours(multi-frame)" and "Ours(single-frame)" employ our double retrieval model and use multiple frames and single frames as input respectively.

4.2 Quantitative Results

The main results of our modularized framework are shown in Table 2 and Table 3. Note that we use neither *projection* nor *constrained-decoding* here. Our performance on COIN is astonishing and doubles the success rate of previous works. According to the result tables and the independent modular result shown in Table 1 and 4, we draw the following conclusions:

1. The language first approach brings significant accuracy improvement to procedure planning tasks.

2. Our modularized framework outperforms the base model which considers vision-to-text transformation and text planning independently. It demonstrates that two sub-modules are complimentary and mutually beneficial.

3. LMs demonstrate strong ability in planning while the mapping from visual observations to the text space remains a challenge. Also, the

	$T = 3$		
Model	**SR**	**mAcc**	**mIoU**
Random	<0.01	0.94	1.66
DDN(Chang et al., 2020)	12.18	31.29	47.48
PlaTe(Sun et al., 2022)	16.00	36.17	65.91
Ext-GAIL (Bi et al., 2021)	21.27	49.46	61.70
P3IV(Zhao et al., 2022)	23.34	49.96	73.89
Ours(multi-frame)	**30.55**	**59.59**	**76.86**
Ours(single-frame)	**25.01**	**53.79**	**75.43**
	$T = 4$		
Model	**SR**	**mAcc**	**mIoU**
Random	<0.01	1.83	1.66
DDN(Chang et al., 2020)	5.97	27.10	48.46
PlaTe(Sun et al., 2022)	14.00	35.29	55.36
Ext-GAIL(Bi et al., 2021)	**16.41**	43.05	60.93
P3IV(Zhao et al., 2022)	13.40	44.16	70.01
Ours(multi-frame)	15.97	**50.70**	**75.30**
Ours(single-frame)	14.11	**47.93**	**73.21**

Table 2: Procedure planning results (%) on CrossTask.

	$T = 3$		
Model	**SR**	**mAcc**	**mIoU**
Random	<0.01	<0.01	2.47
DDN(Chang et al., 2020)	13.90	20.19	64.78
P3IV(Zhao et al., 2022)	15.40	21.67	76.31
Ours(base)	12.27	**33.29**	59.76
Ours(multi-frame)	**30.64**	**54.72**	**80.64**
Ours(single-frame)	**28.35**	**53.14**	**78.56**
	$T = 4$		
Model	**SR**	**mAcc**	**mIoU**
Random	<0.01	<0.01	2.32
DDN(Chang et al., 2020)	11.13	17.71	68.06
P3IV(Zhao et al., 2022)	11.32	18.85	70.53
Ours(base)	3.52	**24.81**	52.48
Ours(multi-frame)	**18.52**	**49.31**	**80.32**
Ours(single-frame)	**15.43**	**45.04**	**78.07**

Table 3: Procedure planning results (%) on COIN.

Visual Form	Steps	**SR-COIN**	**SR-CrossTask**
Multi-frame	3	37.83	47.48
Single-frame	3	35.22	39.37
Multi-frame	4	31.03	40.95
Single-frame	4	30.38	36.44

Table 4: Step retrieval accuracy (%) for both start and end steps.

performance of BART drops with increasing horizon due to variable executable plans.

4.3 Ablation Studies

We conduct two categories of ablation studies: (1) on the language model fine-tuning, including evaluating the impact of different prompts, as well as different approaches to constrain the generation; (2) on the vision-to-text transformation, with different transformation settings adopted.

Impact of language model prompts We use three types of language model prompts to obtain the intermediate steps from the start step and the end step. The prompts are:

- Prompt 1: "Taking $T-2$ steps from + a_1 to a_T + we need to."
- Prompt 2: "You start from a_1. Your goal is a_T. List $T-2$ steps to do this."
- Prompt 3: "For Task d , given the first step and the last step, a_1, a_T. Predict the intermediate $T-2$ steps."

Note that all the actions here are interpreted as their textual expression. The results of predicting the intermediate steps with the given three prompts are shown in Table 5. The first two prompts are very different but of the same amount of information (including two steps plus a count of interval steps) while the third prompt add in the task description label. Experiments show that the prompts do not have a major impact on the language planning performance. And adding in the task name will bring a visible increase. This increase is mainly brought by some overlapped step names. For example, the task `PractiseTripleJump` contains a sequence of steps of {"begin to run up", "do the first two jumps", "do the third jump", "begin to run up"}, while the task `PractisePoleVault` contains a sequence of steps of {"begin to run up", "begin to jump up", "fall to the ground", "begin to run up"}. The "task name" label can help the language model distinguish between this two samples.

Impact of projection The result of using *projection* and *constrained-decoding* is shown in Table 7. We witness only marginal increase in the overall accuracy when adding constrained decoding, which proves that LMs adapt well to the new data domain.

	$T = 3$			$T = 4$		
Method	**SR**	**mAcc**	**mIoU**	**SR**	**mAcc**	**mIoU**
Prompt1	66.03	66.03	66.03	34.87	49.95	61.63
Prompt2	65.96	65.96	65.96	34.83	49.72	61.41
Prompt3	**67.37**	67.37	67.37	**35.43**	51.12	62.89

Table 5: Evaluation (%) of different language prompts on COIN dataset.

Retrieval Model	Prec@1 (%)
BLIP	<1.00
BLIP-finetuned	21.30
Double Retrieval	**37.83**
w/o language loss	24.81
w/o task name	33.32

Table 6: Retrieval performance of different models to get both start image and end image predicted right on COIN.

Approach	$T = 3$		
Constraining Method	**SR**	**mAcc**	**mIoU**
No constraint	28.35	53.14	78.56
Sentence-BERT	29.11	53.45	80.07
Constraint decoding	29.02	53.30	79.67
Approach	$T = 4$		
Constraining Method	**SR**	**mAcc**	**mIoU**
No constraint	15.43	45.04	78.07
Sentence-BERT	16.95	45.82	79.92
Constraint decoding	16.86	46.02	79.43

Table 7: Evaluations on how different approaches to constrain our generation result will influence the final accuracy.

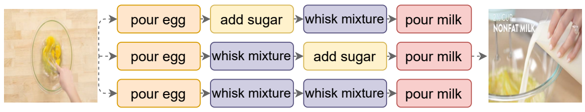

Figure 4: Probabilistic modeling results. We enable language models to generate different outputs via sampling.

Impact of retrieval model design As shown in Table 4, we further evaluate the performance of our double retrieval model by presenting the retrieval performance of the first step and the last step (rather than retrieving the intermediate steps in the planning task). The success rate is determined by the retrieval correctness of both the first and last steps.

To verify that our design of double retrieval is effective in transforming visual details into language, we compare it with the state-of-the-art visual-language transformation approaches in Table 6. We observe that directly finetuning a BLIP retrieval model does not work well. This is due to the difficulty of predicting two steps independently from the visual input.

We also present the ablation studies of removing language loss and task name in Table 6. The performance drop indicates the importance of the language loss term and the additional task name term to the success of our double retrieval model.

Probabilistic modeling ability LMs inherently have the ability of probabilistic modeling. As a result of experimenting with different decoding methods (greedy search, beam search, and sampling) for LMs, we found that the overall accuracy difference is less than 1%. We recognize, however, that the model is capable of generating multiple reasonable plans for a given input. For example, in Figure 4, alternative planning results can be produced through sampling. All alternative predictions are tagged as correct in the test set. It matches the observation that multiple alternative plannings can exist given the same start step and the same goal.

5 Conclusion and Future Work

We introduce a new language-first perspective for the procedure planning task, and propose two models to construct a text planning space and transfer the generalization ability of LMs to vision-based planning. Different from previous approaches that derive a latent space from visual features to perform planning, we propose that a language model with sufficient priors can serve as a better planning space. The key challenge is enabling LMs to capture appropriate visual details for planning purposes. We transform visual input into language and propose a double-retrieval mechanism to force the model to align salient visual details with actions. The superior performance of our approach prove that using language models with strong priors is a promising and powerful paradigm to procedure planning over visual observations.

In the future, we would like to explore the domain generalizability of LM-based planning models and extend our model to handle longer planning horizons, possibly with the help of sub-goal prediction.

6 Limitation

We reflect on the limitations of our model as below:

1. Our experiments are based on large everyday household datasets (i.e. COIN and Crosstask). Our language model is pretrained with web data, which helps it handle such household-related procedures well. However, when applied to other more specialized domains like medical procedures, language models might suffer from the domain gap and impact overall model performance.

2. The language model has excellent planning ability given the ground truth start and goal steps. However, it is still hard for the language model to generate very long sequences of steps. When the planning horizon T increases, the performance of our model drops quickly just as other methods do.

3. In real-world applications (i.e planning task for robots), a good model should be able to dynamically adjust the plan given external feedback. For example, when the execution of one step fails, the model will need to re-plan as soon as possible. Our model does not possess such an ability so far, since our planning approach is offline. We leave this direction for future research.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Jayant Joshi, Ryan C. Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego M Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, and Mengyuan Yan. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *ArXiv*, abs/2204.01691.

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

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

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022a. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Vincent Micheli and Francois Fleuret. 2021. Language models are few-shot butlers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9312–9318, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

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

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

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

Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, Shih-Fu Chang, Mohit Bansal, and Heng Ji. 2022. Language models with image descriptors are strong few-shot video-language learners.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Andy Zeng, Adrian S. Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Peter R. Florence. 2022. Socratic models: Composing zero-shot multimodal reasoning with language. *ArXiv*, abs/2204.00598.

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

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

A Appendix

A.1 Experiment Settings

We trained and evaluated our approach on a single RTX3090 GPU. For COIN and Crosstask dataset processing, we transform the visual observations of a video segment into images. Under our single image setting, we followed previous works and used the first frame of the video segment for the start visual observation while using the last frame to represent the goal visual observation. Under our multiple-image setting, we uniformly sampled 9 images from the videos. The image size is 384*384 under the single image setting while the 9 images are concatenated and then resized to 384*384 under the multiple image setting.

For the baseline model, we used the original image captioning model of Blip. We used the prompt "A picture of" for all the captioning samples. We set the min-length and the max-length of generation to 5 and 20 independently and set the number of beams to 3.

For the language planning side, we employed BART language model (Lewis et al., 2019). During the fine-tuning process, we set the batch size to 16 and used the Adam optimizer with $lr = 10^{-5}$ and weight decay as 0.02. For the double retrieval side, we initialize the model with a BLIP pretrained model checkpoint. During training, we set the batch size to 4 and used an Adam optimizer with a learning rate of 10^{-5} and 0.05 weight decay.

To get our main results on the COIN dataset, it costs about 12 hours to independently fine-tune the language model and train the double retrieval model.

Examples of output We give more examples of our Modularized Framework output in this section. In Figure 5, we provide an example where our model makes a successful prediction. In Figure 6, we show an example where the language model fails. In Figure 7, we show an example where using the multi-image input gets the right prediction while using the single-image variant makes mistakes. It shows that the alignment ability from visual observations to step(action) space is still our model's bottleneck.

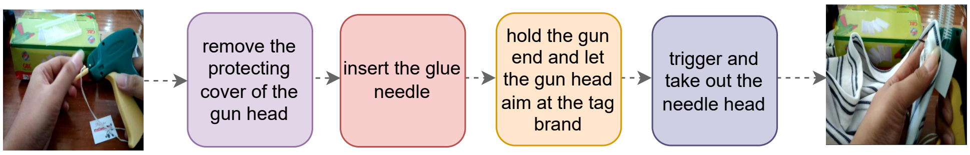

Figure 5: We present a perfect prediction example in this figure. We used single image as input and generate a plan of Horizon $T = 4$. We get all the steps right in this example.

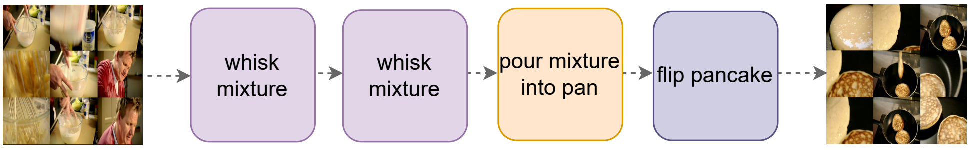

Figure 6: We present prediction example where the double retrieval model works well while the language model fail to predict the right sequence. In this figure. We used Multiple image as input and generate a plan of Horizon $T = 4$. We get one intermediate step predicted wrong in this example. The Right sequence (Ground Truth for this input) is: "**Step1** : whisk mixture", "**Step2** : pour milk", "**Step3** : pour mixture into pan", "**Step4** : flip pancake"

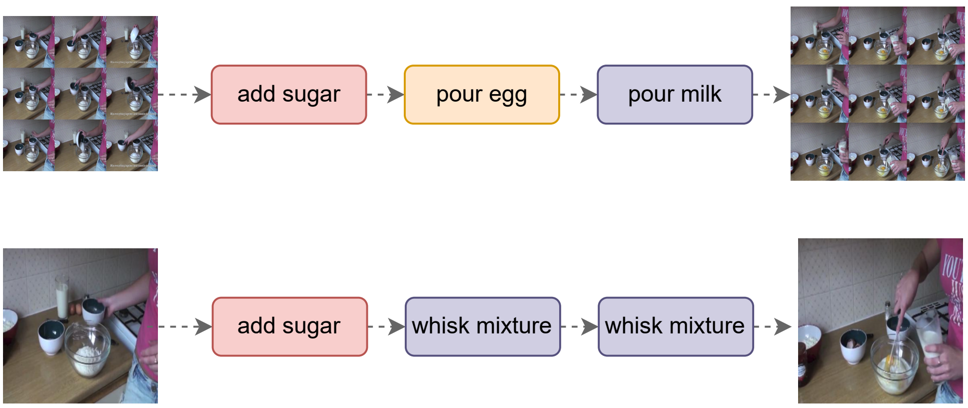

Figure 7: The multi-image setting provides more detailed visual information which helps with the prediction. As is shown in the figure, the multi-image setting has a right prediction(i.e. add sugar, pour egg, pour milk). Using single images, it's easy for us to ignore that the last step is actually pouring milk instead of whisk misture.