Pretrained Language Models as Visual Planners for Human Assistance

Dhruvesh Patel[1,2] Hamid Eghbalzadeh[1] Nitin Kamra[1]
Michael Louis Iuzzolino[1] Unnat Jain[1*] Ruta Desai[1*†]
[1]Meta [2]UMass Amherst

https://github.com/facebookresearch/vlamp

Abstract

In our pursuit of advancing multi-modal AI assistants capable of guiding users to achieve complex multi-step goals, we propose the task of 'Visual Planning for Assistance (VPA)'. Given a succinct natural language goal, e.g., "make a shelf", and a video of the user's progress so far, the aim of VPA is to devise a plan, i.e. *a sequence of actions such as "sand shelf", "paint shelf", etc. to realize the specified goal. This requires assessing the user's progress from the (untrimmed) video, and relating it to the requirements of natural language goal,* i.e. *which actions to select and in what order? Consequently, this requires handling long video history and arbitrarily complex action dependencies. To address these challenges, we decompose VPA into video action segmentation and forecasting. Importantly, we experiment by formulating the forecasting step as a multi-modal sequence modeling problem, allowing us to leverage the strength of pre-trained LMs (as the sequence model). This novel approach, which we call* ***V****isual* ***La****nguage* ***M****odel based* ***P****lanner (VLaMP), outperforms baselines across a suite of metrics that gauge the quality of the generated plans. Furthermore, through comprehensive ablations, we also isolate the value of each component – language pre-training, visual observations, and goal information. We have open-sourced all the data, model checkpoints, and training code.*

1. Introduction

Imagine assembling a new piece of furniture or following a new recipe for a dinner party. To achieve such a goal, you might follow a manual or a video tutorial, going back and forth as you perform the steps. Instead of fumbling through a manual, imagine an assistive agent capable of being invoked through natural language, having the ability to understand human actions, and providing actionable multi-step guidance for achieving your desired goal. Such multi-modal assistive agents should be able to reason human activities from visual observations, contextualize them to the goal at hand, and plan future actions for providing guidance.

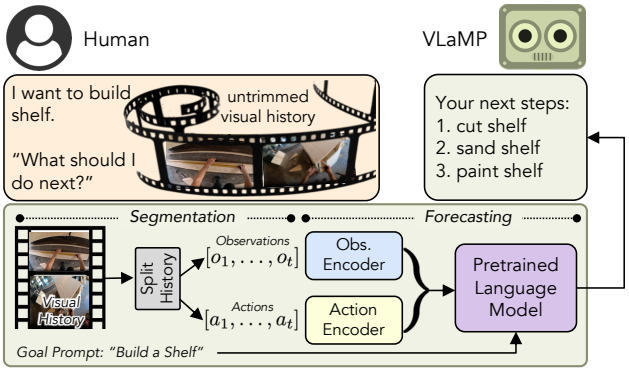

Figure 1: **Visual Planning for Assistance overview (top) and general methodology (bottom).** Given a user-specified, natural language goal ("build a shelf") and corresponding visual history of the user's progress till now, VPA involves predicting a sequence of actions, to assist the user towards the goal ("cut", "sand", "paint"). Our approach is based on multi-modal sequence modeling where we reuse pre-trained video segmentation and language models.

To quantify the progress and aid the development of such multi-modal neural models, we need a intuitive task, as illustrated in the example in Fig. 1 (top). We call this Visual Planning for Assistance (VPA) that we detail next. Given a user-specified goal in natural language ("build shelf") and corresponding video observations of the user's progress towards this goal, the task objective should be to generate the ordered sequence of next actions towards achieving the goal ("cut → sand → paint"). We base VPA off instructional YouTube videos of such procedural activities from large, open-sourced datasets – CrossTask [89] and COIN [73]. This is a natural choice as procedural human activities (cooking, assembly, repair, *etc.*) are a perfect source of multi-step and complex sequence of actions, where humans routinely seek guidance. Realistic nature

*equal mentoring †corresponding author

of VPA makes it particularly challenging. Operating on untrimmed videos requires dealing with potentially many irrelevant background frames, chunking actions in the videos will also be imperative. Another challenge is the validity of the plan of actions *i.e*. they must respect the constraints of the activity – the shelf shouldn't be painted before sanding.

The natural next question is – what's a good way to tackle VPA? Marrying research from video forecasting & anticipation with embodied AI [19, 17], we cast VPA as *goal-conditioned* task planning. We approach VPA by utilizing video action segmentation and transformer-based neural sequence modeling – the former allows us to deal with long video history and the latter can handle arbitrary sequential constraints [86, 39]. The formulation (particularly, transformer-based neural sequence modeling) allows us to tap into the prowess of pre-trained language models (PTLMs). These PTLMs contain useful priors about action-action similarity, action-goal association, and action ordering [7, 30, 58], that our formulation can piggy-back off. Particularly, our model – Visual Language Model Planner (VLaMP) – conditions the generated plan onto the visual history by using a transformer-based mapper network that projects embeddings corresponding to visual history into the input space of the LM. Using VPA as the testbed, we show that VLaMP outperforms a range of standardized baselines. We undertake head-on ablations to quantify effect of each component of VLaMP – language pre-training of the LM, the visual history, and goal description. We believe the modularity of our approach can allow researchers to swap components with their own, to made rapid progress towards VPA.

In summary, our main contributions are: (1) a new task VPA on human interaction videos capturing unique aspects of real-world vision-powered assistive agents; (2) a general-purpose methodology for VPA, which allows leveraging pre-trained multi-modal encoders and sequence models; (3) an instantiation of this methodology (we call VLaMP), where we reuse sequence priors from PTLM and investigate its efficacy across two, well-studied datasets of procedural human activities.

2. Related Work

Forecasting in Videos. VPA entails future action sequence prediction for a video and is closely related to anticipation in video understanding, including future localization [22, 40, 78], future frame prediction [77, 46, 53], next active object estimation [21, 5, 25, 67, 4, 24, 44], as well as short- and long-term action anticipation [66, 15, 20, 23, 25, 65, 64, 57, 45, 1, 52, 50, 11, 79]. While short-term forecasting approaches, such as [66], are limited to predicting the single next action occurring only a few seconds into the future, our work focuses on predicting sequences of actions over longer time horizons (on the order of minutes) into the future. The long-term action (LTA) forecasting benchmarking was recently established on a small subset of Ego4D [25], and although VPA is similarly devised to predict a temporally ordered sequence of actions conditioned on video-based visual observations, our approach differs in that the LTA task does no goal-conditioning. Our task incorporates goal-conditioning on long term sequence prediction as we argue this will drive the development of critical system aspects, such as virtual assistants that afford human interaction via the natural language goals. Additionally, in contrast to LTA, we focus on a wide range of goal-oriented activities available within the CrossTask and COIN datasets. Tab. 1 highlights the differences between VPA and other forecasting tasks.

	Action Prediction	Goal Modality	Visual Reasoning
Action anticipation [14]	Single	None	Video-based
Action forecasting [25]	Multiple, ordered	None	Video-based
Procedural planning [9]	Multiple, ordered	Vision	Image-based
VPA (ours)	Multiple, ordered	Natural Language	Video-based

Table 1: **VPA *vs*. prior tasks related to forecasting of real-world human activities.** We predict an ordered sequence of actions given video history with a focus on natural language goal-conditioning for human-assistive applications.

Procedural Planning Approaches. VPA is similar to the task of procedure planning [9], wherein given a starting and terminating visual observation, the aim is to predict the actions that would transform the state from starting to terminating. However, we argue that due to the unavailability of the terminating visual state, the procedure planning task is not useful in a real-world assistance setting. While several recent works introduce novel models for procedure planning, these models cannot be deployed for VPA either because they rely heavily on the availability of visual goal [70, 87, 6], or assume access to true action history [49].

Transformers for Decision Making. VLaMP autoregressively predicts future states and/or observations and actions and is similar in spirit to sequence models for decision making such as GATO [61], Decision Transformer [12], and Trajectory Transformer [35]. VLaMP extends such models to work with egocentric observations that humans observe in the day-to-day activities.

Planning in Embodied AI using LMs. Past research has leveraged PTLMs for task planning in real world for embodied agents [30, 68, 41, 7, 42, 31, 18, 16, 36]. Many of these works focus on converting a high-level task or instruction into sequence of low-level steps and then ground them in the environment using either affordance functions [7], visual feedback [31], or multimodal prompts [36, 18]. Owing to their focus on robotic agents, these works focus on predominately pick-place tasks. Instead VPA is focused on

complex tasks, in which humans might require assistance in the form of recommended future actions. Consequently, VLaMP requires grounding and reasoning of much more complex states and actions, which it accomplishes by finetuning a pretrained LM on multimodal sequences of (visual) observations and (text-based) actions.

Multi-modal LMs. Recent works have successfully used the transformer architecture for modeling multi-modal sequences that have visual tokens. For instance, [47, 59, 2, 48, 29, 18] train large transformer based multi-modal sequence prediction models. While [3, 74, 56] focus on adapting pre-trained LMs to work with visual tokens by aligning the representation spaces of the two modalities. VLaMP's approach of modeling sequence of visual and textual representations using PTLMs is similar in spirit to these, *i.e*. VLaMP also uses a mapper network to align video representations to LM's token space and jointly learns the mapper with LM finetuning for VPA. However, in contrast to previous works, VLaMP predicts the visual tokens autoregressively at inference time to enable forecasting of the state for planning. Consequently, VLaMP's token prediction loss is multi-modal, unlike most multi-modal LMs that only use token prediction loss for text tokens.

3. Visual Planning for Assistance

Here, we introduce the task of Visual Planning for Assistance (VPA), towards enabling multi-step guidance to humans in their real-world activities. We instantiate VPA for procedural activities, where humans routinely seek assistance. In this section, we include the definition of VPA, and describe the evaluation protocol.

3.1. Task Definition

The following two intuitive inputs are given to any model performing VPA.

Goal Prompt (G). The natural language description (in short phrase) of the user's goal, emulating a typical user's request for assistance for a day-to-day task. Examples include, "build a shelf" and "change a tire".

Visual History (V_t). An untrimmed video that provides context about the user's progress towards a goal from the start till time, say t. We assume that V_t contains k actions or steps $\{a_1, \ldots, a_k\}$ pertaining to the goal. However, VPA doesn't have access to $\{a_1, \ldots, a_k\}$ or k and must work with V_t.

Given these two inputs, V_t and G, the objective of VPA is to generate a *plan* $\mathcal{T}$. The plan is a sequence of actions that should be executed (in the next steps) to assist the user in achieving the goal G. Concretely, the prediction is denoted by $\mathcal{T} = (a_{k+1}, \ldots, a_{k+l})$, where a_i are represented in natural language but come from a closed set $\mathcal{A}$. Here, $l \leq K$ denotes the number of future actions that should be predicted, out of the K number of remaining actions required to accomplish the goal. For our shelf-building example, the correct $\mathcal{T}$ = (`sand shelf`, `paint shelf`, `attach shelf`), capturing the remaining 3 future actions (here, $l = N = 3$).

In day-to-day activities, we request assistance for goals in natural language. However, prior works *procedural planning* [9, 6] assume access to visual goal state. This is not a realistic assumption for an AI agent assisting humans. Hence, we purposefully relax this assumption, making our formulation of the task significantly more practical. Moreover, an agent with access to the visual modality, should be able to improve its plan by filtering out relevant information from the raw video stream of the progress. These are the two central assumptions around which we formulate the task of VPA.

3.2. Evaluation

Open-Sourced Video Data. We leverage existing datasets CrossTask [89] and COIN [73], originally developed to enable video action understanding for VPA, based on the following three requirements:

- *Rich diversity of activities from multiple domains*: The data from different domains such as cooking, assembly etc., enables testing of VPA models in a more generalized manner.
- *Goal-oriented activities consisting of long sequences of actions*: Since the objective is to generate l future actions, activities in these datasets that require diverse sequences of actions e.g., "making a pancake", instead of "running" are more suitable for VPA.
- *Action annotations from a fixed closed set of actions* : Action labels described using verb-noun from a finite set [14], which are temporally aligned with the videos. This makes evaluating the accuracy of $\mathcal{T}$ prediction straightforward. Specifically, free-form, narration-style descriptions of actions that are available in recent video datasets aid in efficient multi-modal representation learning for video understanding [25, 54]. However, evaluating the quality of action *sequences* towards goal achievement, where each action is described in free-form natural language, is non-trivial. We leave the instantiation of VPA with free-form natural language actions as future work.

Table 2 summarizes the features from CrossTask and COIN, aligned with the above requirements.

Metrics. The planning performance of a VPA model is measured by comparing the generated plan $\hat{\mathcal{T}} = (\hat{a}_{k+1}, \ldots, \hat{a}_{k+l})$ to the ground truth plan $\mathcal{T}$ for l actions in the future, given V_t and G. Here $\hat{a}_{k+i}$ denotes the prediction for the $k+i$-th step given history till k-th step. Consistent with community practices [9, 6], we use the following metrics, listed in decreasing order of strictness: *success rate* (SR), *mean accuracy* (mAcc), *mean intersection over union*

Dataset	# train videos	# test videos	# test samples	actions per video	# goals	# domains
CrossTask [89]	1756	752	4123	7.6 ± 4.3	18	3
COIN [73]	9428	1047	2011	3.9 ± 2.4	180	12

Table 2: **VPA datasets.** We evaluate VPA on two existing video datasets containing multiple goal-oriented procedural activities from varied domains. Such activities contain sequences of multiple actions making them ideal for VPA.

(mIOU). Success rate requires an exact match between all actions and their sequence between $\hat{\mathcal{T}}$ and $\mathcal{T}$. Mean accuracy is the accuracy of the actions at each step. Unlike success rate, mean accuracy does not require a 100% match to ground truth. Instead it considers matching at each individual step. Lastly, mean intersection over union captures the cases where the model predicts the steps correctly, but fails to identify the correct order. Concretely,

$$\text{mIOU}_l = \frac{\left|\hat{a}_{\{k+1:k+l\}} \bigcap a_{\{k+1:k+l\}}\right|}{\left|\hat{a}_{\{k+1:k+l\}} \bigcup a_{\{k+1:k+l\}}\right|}, \tag{1}$$

$$\text{mAcc}_l = \frac{1}{l}\sum_{i=1}^{l} \mathbb{1}[\hat{a}_{k+i,} = a_{k+i}], \tag{2}$$

$$\text{SR}_l = \prod_{i=1}^{l} \mathbb{1}[\hat{a}_{k+i} = a_{k+i}], \tag{3}$$

where $\mathbb{1}[\cdot]$ is the identity function, which is 1 when the condition in its input is true, and 0, and $\hat{a}_{\{k+1:k+l\}}$ denotes the *set* of l future actions in $\hat{\mathcal{T}}$, i.e., a sequence but disregarding the order. To complement the above metrics, we also measure the accuracy of predicting the next action *i.e.* nAcc, as defined in (4), where 'n' stands for next. Note that all metrics are averaged over the test set details of which are included in Sec. 5.

$$\text{nAcc} = \mathbb{1}[\hat{a}_{k+1} = a_{k+1}] \tag{4}$$

4. Visual LM Planner

VPA can be viewed as a sequential decision making problem, where the model (say, π) predicting the sequence of next actions is a policy conditioned on the visual history V_t, serving as (partially-observed) state, and goal prompt G. Inspired by the offline learning formulation closest to 'learning from offline demonstrations' [27, 76, 37, 81], in VLaMP, we formulate VPA as a goal-conditioned, multi-modal sequence prediction problem. [1] This formulation allows us to leverage high-capacity sequence models like Transformers [75], which have been successfully applied to sequential decision making [35, 12]. Furthermore, inspired by the recent success of pre-trained transformer language models (PTLMs) on such tasks, [62, 30, 12], we propose to leverage PTLMs as the sequence model in our formulation. In the remainder of this section we describe our approach for VPA, called **V**isual **La**nguage **M**odel based **P**lanner (VLaMP), consisting of a segmentation module and PTLM based sequence prediction module.

4.1. Planning with Segmentation and Forecasting

We define π as a goal-conditioned, multi-modal sequence prediction problem:

$$\pi = P(a_{k+1}, a_{k+2}, \cdots \mid V_t, G). \tag{5}$$

where π models the probability of goal-relevant and valid future actions sequences conditioned on V_t and G.

Modeling the multi-modal sequence in Eq. 5 is computationally expensive and difficult to scale because of the high-dimensional state space of raw untrimmed video V_t.[2] Also, there is limited data to learn the distribution over valid action sequences for goals in real-world applications. Tackling these challenges, we argue that the latent space of factors influencing the future plan can indeed be expressed in lesser dimensions. Particularly, to ensure scalability for handling raw videos and sample efficiency in learning valid action sequence distributions, we *decompose* our policy π into two modules. The first module is *video segmentation*, which converts the untrimmed video history V_t into a sequence of video segments *i.e.* a segment history $S_k = (s_1, \ldots, s_k)$, where each segment corresponds to an action a_i that occurred in the video. The second module enables *forecasting i.e.* it transforms the output of the segmentation module and generates the plan (see Fig. 1 (bottom)). While a probabilistic formulation of this decomposition can be expressed as:

$$\pi = \sum_{S_k} \underbrace{P(a_{k+1}, a_{k+2}, \cdots \mid S_k, G)}_{\text{Forecasting}} \underbrace{P(S_k \mid V_t)}_{\text{Segmentation}}, \tag{6}$$

where the segment history S_k is a latent variable, the summation over all possible segment histories is intractable. We, however, use Eq. 6 as the guiding expression to formulate the input and output of both modules. Next, we include the technical details of both of these modules.

4.2. Segmentation Module

This module splits the untrimmed video history V_t into segment history S_k of multiple segments, each segment

[1] Future works may explore alternate policy optimizations based on reward shaping, inverse RL, or by employing additional online interactions through photorealistic simulation [38, 63, 71, 69, 82, 33, 34, 10, 80].

[2] A typical untrimmed video of 100p video of 5 minutes at 8fps will have 2^7 raw pixel values.

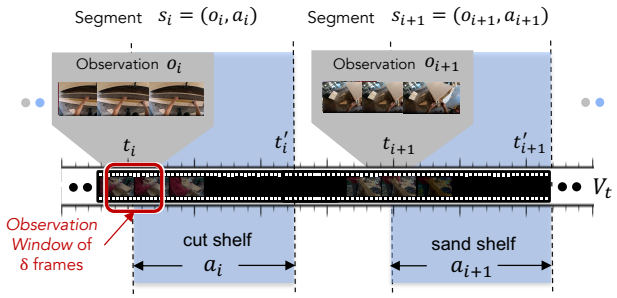

Figure 2: **VLaMP – Segmentation Module.** The untrimmed visual history V_t is converted into segments, each consisting of observation o_i and the action a_i. The observation o_i is the collection of video frames of δ seconds around the start time stamp t_i of the corresponding action a_i. Two such segments are shown here.

corresponding to an action. The segmentation is done using a video-action segmentation model in three steps: preprocessing, classification, and consolidation. In the preprocessing step, the raw video frames from V_t are bundled into fixed-length window clips c_i, each of length 1 second, to obtain $V_t = (c_1, \ldots, c_t)$. In the classification step, a video-action segmentation model is used to output the most probable action for each clip c_i, which can be denoted as $\tilde{A}_t = (\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_t)$. Finally, in the consolidation steps, we convert $\tilde{A}_t$ into a form that can be used by the forecasting module; this form consists of two sequences: **action history** A_k and **observation history** O_k. To this end, same actions in consecutive seconds in $\tilde{A}_t$ are consolidated to form the action history $A_k = (a_1, \ldots, a_k)$. As illustrated in Fig. 2, assuming t_i denotes the starting timestamp for a_i, we also extract video frames from $t_i - \delta/2$ to $t_i + \delta/2$ to obtain a *observation window* o_i corresponding to a_i, and consequently the full observation history $O_k = (o_1, \ldots, o_k)$. The resultant segment history is termed $S_k = ((o_1, a_1), \ldots, (o_k, a_k))$, summarized in Fig. 2.

4.3. Forecasting Module

The usefulness of π's decomposition expressed in Eq. (6) becomes apparent now. Modeling the segmentation module's output as segment history S_k, where each segment consisting of action and observations, allows writing the output of the forecasting module in an autoregressive manner:

$$P(a_{k+1}, a_{k+2}, \cdots \mid o_1, a_1, \ldots, o_k, a_k, G) = \prod_{i>0} \sum_{o_{k+i}} P(o_{k+i}, a_{k+i} \mid o_1, a_1, \ldots, o_k, a_k, G). \quad (7)$$

Illustrated in Fig. 3, this autoregressive expression allows the possibility of using any sequence-to-sequence neural network as the forecasting module in combination with pretrained text and video encoders for representing action and observation history A_k and O_k. This general-purpose framework allows the use of any neural sequence model – LSTM [28], GRU [13], Transformers [75], *etc*. We choose to instantiate the forecasting module for π using a pretrained transformer-based LM. Next we present the details of the encoders for the two modalities and the LM based sequence model.

Action encoder (f_{act}) Each action a_i in A_k is encoded by f_{act} and the output is denoted by α_i. Concretely, token embeddings are expressed as:

$$(\alpha_1, \ldots, \alpha_k) = (f_{\text{act}}(a_1), \ldots, f_{\text{act}}(a_k)), \text{where}$$
$$\alpha_i = (\alpha_i^1, \ldots, \alpha_i^{r_i}) \in \mathbb{R}^{r_i \times d} \quad (8)$$

As we illustrate in Fig. 3 (left), each action a_i is tokenized into r_i tokens using appropriate tokenizer for the LM, the tokens are indexed using the vocabulary of the LM, and are represented using an embedding lookup from the token embeddings of the LM to produce α_i. Here, r_i is the number of tokens and d is the dimensionality of token embeddings.

Observation encoder (f_{obs}). Visual cues play an integral role in knowing what actions lie ahead, towards achieving the goal prompt G. To this end, the visual observations O_k are encoded and play a critical role in the planner. Recall, the visual observation history O_k comprises of o_i corresponding to action a_i, each of δ frames. As illustrated in Fig. 3 (middle), we transform each o_i employing the widely-adopted S3D backbone [84] f_{S3D^*} ($*$ denotes backbone is frozen). We must project visual encodings to a shared latent space of action (language) embeddings described before (α_i). To this end, we map S3D features via a trainable transformer mapper f_{map}. Concretely,

$$(\beta_1, \ldots, \beta_k) = (f_{\text{obs}}(o_1), \ldots, f_{\text{obs}}(o_k)), \text{where}$$
$$\beta_i = (\beta_i^1, \ldots, \beta_i^\delta) \in \mathbb{R}^{\delta \times d} \text{ and } f_{\text{obs}} = f_{\text{S3D}^*} \circ f_{\text{map}} \quad (9)$$

Overall, as we show in Fig. 3 (right), the resultant encoded sequence of representation for S_k is thus

$$f_{\text{enc}}(S_k) = (\beta_1, \alpha_1, \ldots, \beta_k, \alpha_k) = H_k,$$

Sequence model (f_{seq}). Given the above encoding for the segment history S_k, the role of the sequence model is to predict a representation of the next token, that would in return enable VLaMP plan generation capabilities for VPA. Importantly, in the process of generating sequence of future actions autoregressively, we would also need to generate the representations of 'future observations'. Therefore, as we shown in Fig. 3, our sequence model, which consists of the transformer layers of a PTLM, also produces representations for vision (in addition to the necessary action tokens). Before proceeding further, we pause and introduce additional notation for the sequence model, which

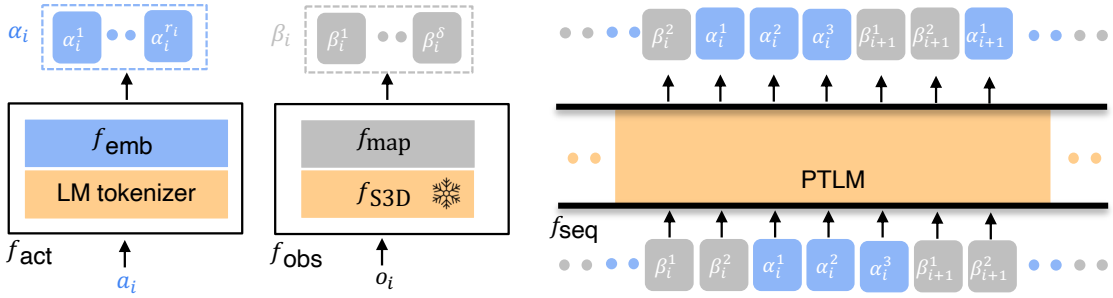

Figure 3: **VLaMP – Forecasting Module.** As shown in left and middle, actions and observations obtained from the segmentation module are encoded using appropriate modality encoders. The observation encoder leverages pretrained video encoder for observations, while also learning a mapper that aligns the representations from observations with actions. As shown on the right, VLaMP uses a joint sequence model on top of interleaved action (blue) and observation (gray) representations to forecast autoregressively the next representation.

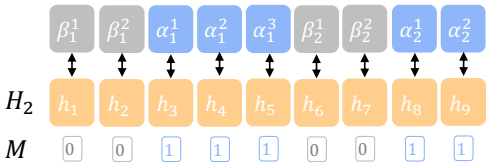

Figure 4: **Tokenized Sequence with Masks.** The encoded sequence of representations for $k = 2$ segments, denoted alternatively using modality agnostic notation of H_2 and mask M for next token prediction training.

will make the subsequent explanation for training and inference easier to follow. As shown in Figure 4, we alternatively denote the sequence of representations $(\beta_1, \alpha_1, \ldots, \beta_k, \alpha_k)$ by $H_k = (h_1, \ldots, h_n)$, with $n = k\delta + \sum_{i=1}^{k} r_i$. A binary mask $M = (m_1, \ldots, m_n)$, where m_i is 1 if the corresponding representation is for an action and 0 otherwise, can help obtain necessary action or visual observations. With this notation, given first j representations denoted as $h_{1:j}$, one step of the sequence model produces the representation for $j+1$, i.e., $f_{\text{seq}}(h_{1:j}) = \hat{h}_{j+1}$.

4.4. Training

The joint training of the segmentation and forecasting modules following Eq. (6) is intractable. [3] But, by exploiting the availability of unpaired training data, we approximate Eq. (6) by feeding in the output of the segmentation module to the forecasting module and training them separately, each on their respective labeled data. The video-action segmentation model is trained utilizing the VideoCLIP setup [85], where in the segmentation model performs classification to predict the action for each second of the video. The forecasting model is trained by adopting the next representation prediction objective. Unlike vanilla LM pretraining, however, we also need to train for predicting visual representations in addition to text (action). Therefore, we use two different losses L_{act} and L_{obs} for text and visual representations respectively. Specifically, L_{act} is the conventional cross-entropy loss over the LM's vocabulary V_{LM} for the action representations while L_{obs} is the mean-squared error between the predicted and the ground truth observation representations. The total loss is the sum of both the loss terms as shown in Eq. (11).

$$L = -\sum_{j=1}^{n} m_j \, L_{\text{act}}(\hat{h}_j) + (1 - m_j) \, L_{\text{obs}}(\hat{h}_j), \text{ where}$$

$$L_{\text{act}} = h_j \cdot \hat{h}_j - \log \sum_{p=1}^{|V_{\text{LM}}|} \exp\left(h_p \cdot \hat{h}_j\right) ; \quad (10)$$

$$L_{\text{obs}} = \frac{\|h_j - \hat{h}_j\|_2^2}{d} \quad (11)$$

In order to have a stable training, we use ground truth action history to construct S_k (and subsequently H_k) instead of the output of the segmentation module. Appendix B provides further details on loss and optimizers for training.

Inference. We next detail the inference procedure for VLaMP. Recall that we use $1:n$ to denote a sequence of n representations (i.e, $h_{1:n} = (h_1, \ldots, h_n)$). Additionally, we denote the concatenation operator over two representation sequences by "$\diamond$". With this notation at hand, we define the *score* of an action $a \in \mathcal{A}$ for following history $h_{1:n}$ as

$$\phi(h_{1:n} \diamond f_{\text{act}}(a)) = \sum_{j=1}^{r_a} a^j \cdot f_{\text{seq}}(h_{1:n} \diamond a^{1:j}), \quad (12)$$

where $\cdot$ is the vector dot product, and $f_{\text{act}}(a) = \alpha^{1:r_a} = (\alpha^1, \ldots, \alpha^{r_a})$ is the sequence of encoded representations

[3] Despite tuning attempts, we found joint optimization to be intractable and inefficient on resources.

for action a. In other words, this *score* is the sum of unnormalized log-probability under the sequence model using the standard softmax distribution. We use this scoring function with to perform beam search (detailed inference algorithm: Algo. 1 in App. B).

5. Experiments

We instantiate VLaMP's segmentation module utilizing VideoCLIP [85] that has been fine-tuned on COIN and CrossTask. Similarly, we instantiate VLaMP's sequence model f_{seq} (in the forecasting module) by GPT2 [60]. We utilize the open-sourced model weights of GPT2 (from HuggingFace [83]). For inference, we use Algo. 1 presented in Appendix B, with beam size $B = 10$ for CrossTask and a $B = 3$ for COIN.

5.1. Data and Baselines

Data. For a video V with goal G, both CrossTask [89] and COIN [73] provide annotations of the form $\{a_k, (t_k, t'_k)\}_{k=1}^{K}$, where a_k are the actions in the video, and t_k (resp. t'_k) are the start (resp. end) timestamps for a_k. [4] Given an annotated video consisting of K steps, we generate $K - l$ examples, each with input $x_k = (G, V_{t_k})$ and output $y_k = (a_{k+1}, \ldots, a_{k+l})$, for $k = 1, \ldots K - l$ (leaving at least l steps to predict in each example). [5] Therefore from M videos, we generated $N = \sum_{m=1}^{M}(K_m - l)$ examples, where K_m is the number of steps in the m-th video, forming a dataset $\mathcal{D} = \{x^{(j)}, y^{(j)}\}_{j=1}^{N}$ suitable for VPA (total number of samples in shown in Table 2).

Baselines. As a first step towards benchmarking, we utilize two heuristic baselines – a *random* baseline and *most probable action* baseline. Additionally, we also adopt a variety of strong goal-conditioned models. The procedure planning task is most relevant task from the literature to VPA, therefore, we adapt (details in App. A.1) the widely used DDN model introduced by Chang *et al*. [9], which is an established model in many procedural planning benchmarks in prior works [6, 87, 70]. Building bridges to the research community working on Ego4D's Long Term action Anticipation benchmark (LTA) [25], we also evaluate their best performing baseline. Finally, we include a prompt-based GPT-3 planner. Zooming out, these baselines span several paradigms: *prompt-based*, and *random* baselines are learning-free, the *most-probable* and *random* baselines are non-neural, while *DDN* and *LTA* baselines are learned and neural. Succinct descriptions of baselines are included next (details are deferred to Appendix A):

- *Random*: Predicts the plan by picking all l actions uniformly randomly from the set of all actions $\mathcal{A}$.
- *Random w/ goal*: A stronger baseline; for each goal G, we allow privilege access to a set of applicable actions to that goal $\mathcal{A}_G \subseteq \mathcal{A}$, and predicts the plan by randomly picking actions from the restricted set.
- *Most probable action*: Given the previous action a_j, picks the most probable next action a_j according conditional probability $Pr(a_i|a_j)$ obtained using frequency count from the training.
- *Most probable action w/ goal*: Akin to random w/ goal baseline, we also evaluate a goal-conditioned most probable action baseline, that uses a goal-specific set of actions $\mathcal{A}_G \subset \mathcal{A}$ during sampling. Since the most probable baselines, provide a probability distribution over the actions, we also employ beam search (for fairness, with the same beam size same as VLaMP), and pick the highest scoring plan.
- *Dual Dynamics Network (DDN)* [9]: Accounting for the difference in task definition, *i.e*. lack of visual goal, a direct application of DDN inference algorithm was not possible. So we keep DDN's network structure but use Algo. 1 for inference on VPA.
- *Ego4D Long Term Action Anticipation (LTA) [25]* : The best performing baseline for LTA uses a SlowFast visual encoder followed by a transformer aggregator. We adapt this baseline using the S3D encoder for a fair apples-to-apples comparison with VLaMP.
- *Prompt-based GPT-3 Planner*: Huang *et al*. [30] utilize a LLM as zero-shot planners, we also experiment with prompting a frozen pretrained large language model *i.e*. GPT-3 for VPA (additional details in App. A.2).

5.2. Quantitative Results

In the following, we include quantitative findings of benchmarking methods on two video data sources (Tab. 3) and head-on ablations (Tab. 4).

Improved performance across video datasets. As we show in Tab. 3, VLaMP significantly outperforms the baselines for both CrossTask and COIN. DDN that is customized for procedural tasks performs significantly better than heuristic baselines leading to a mAcc boost from $12.7 \rightarrow 24.1\%$ (row 2 and 5, $l = 4$, Tab. 3). With our novel decomposition and pretraining objective, VLaMP outperforms DDN with a further bump up from $24.1 \rightarrow 31.7\%$ (row 5 and 7).

Steady gains in short & long horizon predictions. As length of prediction l increases ($l = 1$ to $l = 4$), the performance naturally decreases, across all baselines and tasks. With this in mind and zooming in on COIN re-

[4] Despite the presence of prepositions, majority of actions in COIN and CrossTask can be described using a verb-noun pair. While similar verbs exist, they are never paired with the same nouns. Therefore, there are no repeated actions. Example: while verb 'lift' w/ noun 'barbell' and verb 'pickup' w/ 'button' exists in the action library, the complements ('lift button' & 'pick up barbell') do not.

[5] Since we evaluate for three and four next steps on our datasets, we use the maximum required length and set $l = 4$.

Method	$l=1$	$l=3$			$l=4$		
	{n/m}Acc	SR	mAcc	mIOU	SR	mAcc	mIOU
CrossTask Video Dataset [89]							
Random	0.9	0.0	0.9	1.5	0.0	0.9	1.9
Random w/ goal	13.2	0.3	13.4	23.6	0.0	12.7	27.8
Most probable	10.4	1.7	6.1	9.9	1.3	5.5	13.9
Most probable w/ goal	12.4	2.4	8.9	15.5	1.5	7.9	20.5
DDN [9]	33.4	6.8	25.8	35.2	3.6	24.1	37.0
LTA [25]	26.9	2.4	24.0	35.2	1.2	21.7	36.8
VLaMP (ours)	**50.6**	**10.3**	**35.3**	**44.0**	**4.4**	**31.7**	**43.4**
COIN Video Dataset [73]							
Random	0.1	0.0	0.1	0.2	0.0	0.1	0.2
Random w/ goal	24.5	1.7	21.4	42.7	0.3	20.1	47.7
Most probable	0.7	1.6	4.3	6.8	1.6	8.2	15.3
Most probable w/ goal	23.9	10.9	18.0	24.9	9.1	16.3	32.2
DDN [9]	29.3	10.1	22.3	32.2	7.0	21.0	37.3
GPT-3 prompt-based [30]	19.3	1.7	11.1	19.5	0.0	10.5	21.0
VLaMP (ours)	**45.2**	**18.3**	**39.2**	**56.6**	**9.0**	**35.2**	**54.2**

Table 3: **Performance on different datasets and horizons.** The mean of various metrics (Sec. 3.1) obtained using 5 runs with different random seeds (std. errors are provide in Appendix E). Note that the action and observation history are the output of the separately finetuned video-action segmentation model and hence are noisy compared to the ground truth history.

sults, we observe large gains of VLaMP over DDN (next best method). Particularly, a relative improvement of 54% (29.3 $\rightarrow$ 45.2) in mAcc for $l=1$ (row 7 and 8, Tab. 3) and 68% (21.0 $\rightarrow$ 35.2) for $l=4$ is demonstrated by VLaMP over DDN.

Goal-conditioning is crucial. A key difference between the task formulations of LTA and VPA is goal conditioning (see Sec. 2). Comparing rows for LTA model and VLaMP– there is almost 2x gap (27% *vs.* 51%, l=1, Tab. 3). This underscores the importance of goal-conditioning. To second this inference, in Tab. 4 (rows 1 and 2), we precisely ablate the effect of providing the goal as a *textual description* for the basic (last-observation-only) model. The only difference is goal prompt G, which increases mAcc performance from 44.5 $\rightarrow$ 53.1% for $l=1$ and 28.3 $\rightarrow$ 34.7% for $l=3$.

Goal-conditioned random and most probable baseline come close to DDN on easy metrics. For fair comparisons, we include heuristic baselines. We observe the performance of *random w/ goal* and *most probable w/ goal* comes close to DDN, when evaluated using lenient metrics like mAcc and mIOU (see COIN results in Tab. 3). Note, that these two baselines enjoy the privileged access to (a much smaller) 'relevant actions set' for a given goal. On an average, the relevant or feasible action set is smaller for COIN than CrossTask videos. This apriori access to feasible actions make *random w/ goal* and *most probable w/ goal* competitive (albeit, slightly unfair) baselines.

Prompt-based LLM planner is not competitive. We evaluate the prompt-based GPT-3 planner on COIN and find that the prompt based model performs significantly worse than VLaMP. This highlights that VPA is not easy to solve using just a PTLM and prompting.

	G	A_k	O_k	$l=3$			$l=1$
				SR	mAcc	mIOU	nAcc
1	✗	✗	o_k	6.8 ± 0.3	28.3 ± 1.9	34.8 ± 2.0	44.5 ± 3.8
2	✓	✗	o_k	8.9 ± 0.2	34.7 ± 0.7	41.6 ± 0.8	53.1 ± 2.0
3	✓	✓	✗	14.9 ± 0.3	37.8 ± 0.4	50.8 ± 0.6	48.0 ± 0.2
4	✓	✓	O_k	15.2 ± 0.3	43.5 ± 0.8	51.4 ± 0.9	64.8 ± 0.9
R	✓	✓	O_k	10.7 ± 0.2	36.5 ± 0.7	41.7 ± 0.6	61.4 ± 1.8

Table 4: **Role of different inputs and LM pre-training in VLaMP.** G, A_k, O_k denote the three inputs to VLaMP: goal, action history and observation history, respectively. Shorthand of o_k means only the most recent observation of the full history (O_k) is used. Mean $\pm$ std. error over 5 random seeds on CrossTask are reported. The row **R** corresponds to the forecasting module of VLaMP trained with random initialization (same model architecture) as opposed to initialization using the weights of a pre-trained LM.

5.3. Ablations and Error Analysis

We undertake head-on ablations to quantify effect of each component of VLaMP – action history, the visual history, and language pre-training of the LM. To remove confounding factors, in Tab. 4 we use the ground truth output for the segmentation module. Following this, we include detailed error analysis.

Action and observation history improve complementary planning metrics. As seen in all the rows with action history Tab. 4 (*i.e.* A_k), the provision of action history increases difficult metrics such as SR. In comparing rows 2 and 3, we see that SR increases relatively by 67% (8.9 $\rightarrow$ 14.9), simply by using action history even without access to any past observation. However, the lack of observation history affects nAcc, which drops relatively by 10% (53.1 $\rightarrow$ 48.0) when observation history is swapped by action history between rows 2 and 3. This implies that observation history is important to predict the person's state in the task and what they might do next.

Priors from the pre-trained LM improve performance. In Tab. 4 row **R**, we report the performance of VLaMP when its forecasting module is trained with random weight initialization, *i.e.*, the transformer of the same architecture trained from scratch instead of LM pre-training. We find the performance in row **R** is thus much lower than that of VLaMP with pre-trained LM shown in row 4. This underscores the crucial importance of LM pre-training.

Segmentation errors are detrimental. While the accuracy of video-action segmentation module (particularly, VideoCLIP model) is quite good – 80.2% for CrossTask (68.7%

for COIN), we believe this should be a major focus for future research. Why? The effect of this segmentation error in VLaMP's performance is quantified by comparing VLaMP row in Tab. 3 (CrossTask) with corresponding performance assuming perfect segmentation (Tab. 4, row 4). A relative 50% gap in success and big gaps in all other metrics.

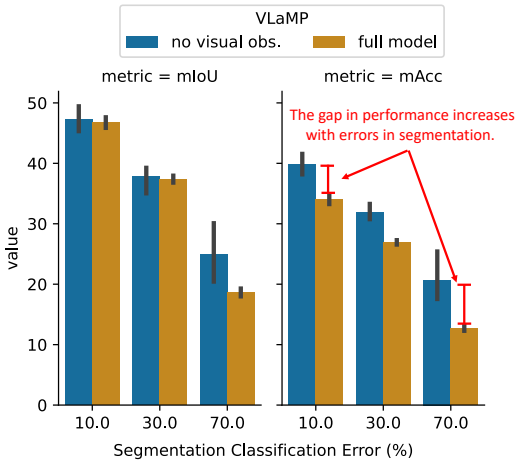

Figure 5: **Effect of segmentation errors.** Performance of VLaMP on CrossTask with classification errors (%) in the video-action segmentation. As the errors increase, the performance gap between the full model and the one without access to observations increases.

Visual observation history mitigates effect of video-action segmentation errors. Errors by segmentation module lead to mis-classification of actions, in turn leading to erroneous action history. To precisely study the effect of erroneous action history on VLaMP's performance, we perform controlled experiments wherein we add noise in the ground truth segmentation. This helps us tune desired segmentation error, irrespective of VideoCLIP's accuracy (in segmentation module). We achieve this by replacing a calculated % of ground-truth actions by random actions. Decreasing performance, with increasing segmentation classification error, is expected and observed in Fig. 5. Further, we compare two models variations, VLaMP(G, A_k, O_k) *i.e.* the full model (uses both action and observation history) and VLaMP(A, A_k) that does not have visual observation history. As we increase segmentation classification error, the gap in the performance of these model variations increases. We infer that visual observations add robustness in VPA against video-action segmentation errors.

Errors do not drop as more history is provided. In Fig. 6, we show mAcc w.r.t. the number of steps k in the history *i.e.* length of (O_k, A_k). As more information is available (due to more steps in the history), one may expect the accuracy to improve. However, this isn't the trend for many tasks. For *e.g.*, note the accuracy drops for the task "add oil to your car" w.r.t k. Digging deeper, we find two reasons for this pattern. *First*, we find that tasks with long history like "add oil to your car" tend to have repetitive steps towards the end of their action trajectories, making it difficult to predict the precise number and the pattern of such repetitions in VPA. *Second* is simply a data issue – long trajectories are also exponentially less frequent in the dataset. This forms the tail of the data distribution of action sequences in the dataset (see App. E for statistics). Consequently, prediction in this tail of activities, is a pertinent challenge for future research.

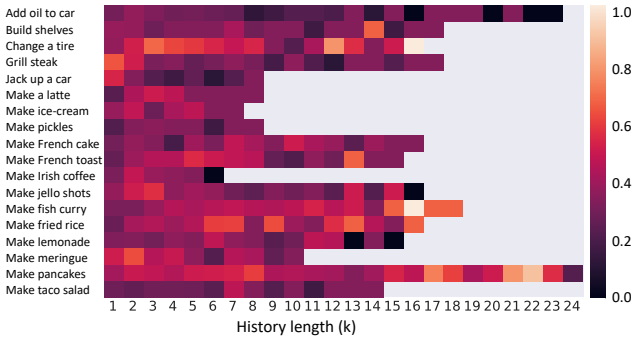

Figure 6: **Accuracy in the tail of long activities is challenging.** Longer history (O_k, A_k) do not necessary lead to better accuracy. Further longer histories occur with increasing rarity, leading to learning challenges. (Plot shows mAcc for $l = 3$ CrossTask *vs*. the #steps in history k)

6. Conclusion

Visual Planning for Assistance is a new and intuitive formulation to support planning from natural visual observations for assisting humans in day-to-day activities. We benchmark VPA using standardized suite of prior baselines and a new multi-modal sequence modeling formation of VLaMP. The novel decomposition of a VLaMP policy into video action segmentation and forecasting leads to several efficiency and modeling benefits, that benefit the community, even beyond VPA. Particularly, this allows to leverage pre-trained LM that leads to significant performance gains. Alternative decompositions, self-supervised pre-training objectives for PTLMs, modeling multiple actors, and tapping into other modalities (audio, camera metadata, *etc.*) [26, 55, 43, 72, 32, 88] are interesting ways forward for the community, on the exciting task of VPA.

Acknowledgements: The authors are indebted to Brian Chen for his efforts to open-source the code and LTA experiments, and to Gideon Stocek for supporting with the cluster management. We thank Vasu Sharma, Tushar Nagarajan, and Homanga Bharadhwaj for their writing feedback. We thank Weichao Mao for early discussions and Asli Celikyilmaz for her constant guidance.

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352, 2018. 2

[2] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022. 3

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

[4] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. 2023. 2

[5] Gedas Bertasius, Hyun Soo Park, X Yu Stella, and Jianbo Shi. First-person action-object detection with egonet. In *RSS*, 2017. 2

[6] Jing Bi, Jiebo Luo, and Chenliang Xu. Procedure planning in instructional videos via contextual modeling and model-based policy learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15611–15620, 2021. 2, 3, 7

[7] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*, 2022. 2

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 14

[9] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *European Conference on Computer Vision*, pages 334–350. Springer, 2020. 2, 3, 7, 8, 14, 16

[10] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020. 4

[11] Guo Chen, Sen Xing, Zhe Chen, Yi Wang, Kunchang Li, Yizhuo Li, Yi Liu, Jiahao Wang, Yin-Dong Zheng, Bingkun Huang, et al. Internvideo-ego4d: A pack of champion solutions to ego4d challenges. *arXiv preprint arXiv:2211.09529*, 2022. 2

[12] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *NeurIPS*, 2021. 2, 4

[13] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 5

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018. 2, 3

[15] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 2

[16] Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. Collaborating with language models for embodied reasoning. *arXiv preprint arXiv:2302.00763*, 2023. 2

[17] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the embodied ai workshop. *arXiv preprint arXiv:2210.06849*, 2022. 2

[18] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023. 2, 3

[19] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022. 2

[20] Yazan Abu Farha, Qiuhong Ke, Bernt Schiele, and Juergen Gall. Long-term anticipation of activities with cycle consistency. *arXiv preprint arXiv:2009.01142*, 2020. 2

[21] Antonino Furnari, Sebastiano Battiato, Kristen Grauman, and Giovanni Maria Farinella. Next-active-object prediction from egocentric videos. *Journal of Visual Communication and Image Representation*, 2017. 2

[22] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *ICCV*, 2019. 2

[23] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13505–13515, 2021. 2

[24] Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *CVPR*, 2022. 2

[25] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2, 3, 7, 8, 16, 17

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 9

[27] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *AAAI*, 2018. 4

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5, 14, 15

[29] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023. 3

[30] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *ICML*, 2022. 2, 4, 7, 8, 14

[31] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022. 2

[32] Unnat Jain, Iou-Jen Liu, Svetlana Lazebnik, Aniruddha Kembhavi, Luca Weihs, and Alexander G Schwing. Gridtopix: Training embodied agents with minimal supervision. In *ICCV*, 2021. 9

[33] Unnat Jain, Luca Weihs, Eric Kolve, Ali Farhadi, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander G. Schwing. A cordial sync: Going beyond marginal policies for multi-agent embodied tasks. In *ECCV*, 2020. 4

[34] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G. Schwing, and Aniruddha Kembhavi. Two body problem: Collaborative visual task completion. In *CVPR*, 2019. 4

[35] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *NeurIPS*, 2021. 2, 4

[36] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022. 2

[37] Bingyi Kang, Zequn Jie, and Jiashi Feng. Policy optimization with demonstrations. In *ICML*, 2018. 4

[38] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 4

[39] Anastasis Kratsios, Behnoosh Zamanlooy, Tianlin Liu, and Ivan Dokmanić. Universal Approximation Under Constraints is Possible with Transformers, Feb. 2022. arXiv:2110.03303 [cs, math]. 2

[40] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014. 2

[41] Shuang Li, Xavier Puig, Yilun Du, Clinton Wang, Ekin Akyurek, Antonio Torralba, Jacob Andreas, and Igor Mordatch. Pre-trained language models for interactive decision-making. *arXiv preprint arXiv:2202.01771*, 2022. 2

[42] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022. 2

[43] Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *ICML*, 2021. 9

[44] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022. 2

[45] Siyuan Brandon Loh, Debaditya Roy, and Basura Fernando. Long-term action forecasting using multi-headed attention-based variational recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2419–2427, 2022. 2

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

[47] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 3

[48] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 3

[49] Weichao Mao, Ruta Desai, Michael Louis Iuzzolino, and Nitin Kamra. Action dynamics task graphs for learning plannable representations of procedural tasks. *arXiv preprint arXiv:2302.05330*, 2023. 2

[50] Esteve Valls Mascaró, Hyemin Ahn, and Dongheui Lee. Intention-conditioned long-term human egocentric action anticipation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6048–6057, 2023. 2

[51] Esteve Valls Mascaro, Hyemin Ahn, and Dongheui Lee. Intention-conditioned long-term human egocentric action anticipation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6048–6057, January 2023. 17

[52] Nazanin Mehrasa, Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. A variational auto-encoder model for stochastic point processes. in 2019 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3160–3169. 2

[53] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human vidoes. *RSS*, 2023. 2

[54] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 3

[55] Himangi Mittal, Pedro Morgado, Unnat Jain, and Abhinav Gupta. Learning state-aware visual representations from audible interactions. In *NeurIPS*, 2022. 9

[56] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 3

[57] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020. 2

[58] Fabian Paischer, Thomas Adler, Vihang Patil, Angela Bitto-Nemling, Markus Holzleitner, Sebastian Lehner, Hamid Eghbal-Zadeh, and Sepp Hochreiter. History compression via language models in reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17156–17185. PMLR, 17–23 Jul 2022. 2

[59] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952, 2021. 3

[60] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, 2019. 7

[61] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 2

[62] Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. proscript: Partially ordered scripts generation via pre-trained language models. *arXiv preprint arXiv:2104.08251*, 2021. 4

[63] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. *ICCV*, 2019. 4

[64] Fadime Sener, Rishabh Saraf, and Angela Yao. Transferring knowledge from text to video: Zero-shot anticipation for procedural actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[65] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 154–171. Springer, 2020. 2

[66] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 862–871, 2019. 2

[67] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 2

[68] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022. 2

[69] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pages 477–490. PMLR, 2022. 4

[70] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters*, 7(2):4924–4930, 2022. 2, 7, 15

[71] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems*, 34:251–266, 2021. 4

[72] Andrew Szot, Unnat Jain, Dhruv Batra, Zsolt Kira, Ruta Desai, and Akshara Rai. Adaptive coordination for social embodied rearrangement. In *ICML*, 2023. 9

[73] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 1, 3, 4, 7, 8

[74] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021. 3

[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 4, 5, 15

[76] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017. 4

[77] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017. 2

[78] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 2

[79] Huiyu Wang, Mitesh Kumar Singh, and Lorenzo Torresani. Ego-only: Egocentric action detection without exocentric pretraining. *arXiv preprint arXiv:2301.01380*, 2023. 2

[80] Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. *NeurIPS*, 2020. 4

[81] Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander Schwing. Bridging the imitation gap by adaptive insubordination. In *NeurIPS*, 2021. 4

[82] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. Allenact: A framework for embodied ai research. *arXiv preprint arXiv:2008.12760*, 2020. 4

[83] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020. 7

[84] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 318–335, Cham, 2018. Springer International Publishing. 5, 14

[85] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021. 6, 7

[86] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are Transformers universal approximators of sequence-to-sequence functions? Feb. 2022. 2

[87] He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. P3iv: Probabilistic procedure planning from instructional videos with weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2938–2948, 2022. 2, 7, 15

[88] Chenhao Zheng, Ayush Shrivastava, and Andrew Owens. Exif as language: Learning cross-modal associations between images and camera metadata. In *CVPR*, 2023. 9

[89] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. 1, 3, 4, 7, 8