Stitch-a-Demo: Creating Video Demonstrations from Multistep Descriptions

Chi Hsuan Wu*, Kumar Ashutosh*, Kristen Grauman
UT Austin

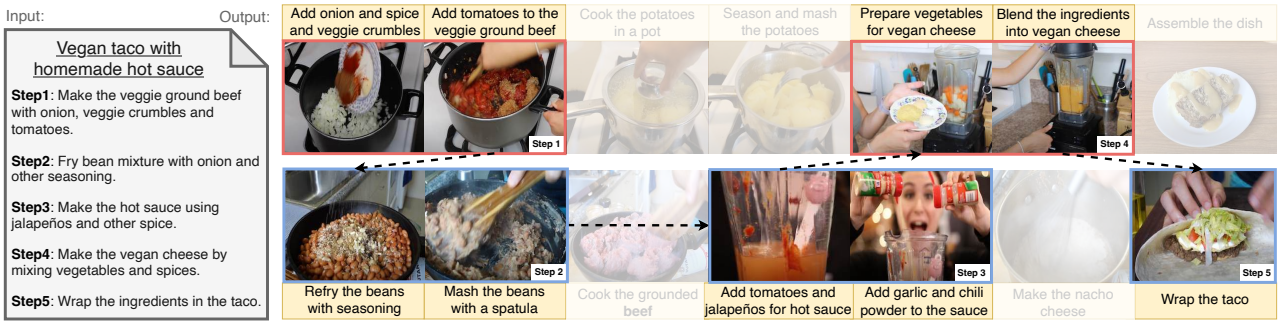

Figure 1. **Video demonstration from multistep descriptions**. Given multistep descriptions (left) aiming to achieve a procedural task, *e.g.* making *vegan taco*, our method obtains clips from thousands of instructional videos to visually demonstrate the procedure (right). The goal is for every clip to correctly describe a step, while maintaining visual consistency. Our proposed method goes beyond current retrieval and generation methods, which fail to faithfully ground multistep procedures in coherent visual sequences.

Abstract

When obtaining visual illustrations from text descriptions, today's methods take a description with a single text context—a caption, or an action description—and retrieve or generate the matching visual context. However, prior work does not permit visual illustration of multistep *descriptions,* e.g. *a cooking recipe or a gardening instruction manual, and simply handling each step description in isolation would result in an incoherent demonstration. We propose Stitch-a-Demo, a novel retrieval-based method to assemble a video demonstration from a multistep description. The resulting video contains clips, possibly from different sources, that accurately reflect all the step descriptions, while being visually coherent. We formulate a training pipeline that creates large-scale weakly supervised data containing diverse procedures and injects hard negatives that promote both correctness and coherence. Validated on in-the-wild instructional videos, Stitch-a-Demo achieves state-of-the-art performance, with gains up to 29% as well as dramatic wins in a human preference study.*

1. Introduction

Instructional or "how-to" videos are commonly used to learn new skills—such as cooking, gardening, repairing bikes, or yoga. These videos contain an explanation of the task, often in the form of multiple procedural *steps*, along with a rich visual demonstration. Video demonstrations have shown to be a great learning aid [57], significantly augmenting written step descriptions for human learners. Meanwhile, in robot learning, training with "passive" video of human skill executions is increasingly attractive for representation learning and efficient imitation [63, 71].

Despite the scale of instructional videos on the internet, they still only represent a sliver of all possible demonstration sequences, given the combinatorics of how different steps can potentially be combined. Any given video assumes a fixed sequence of steps, which might differ from the step description that a person wants to visualize, whether from a recipe book, an instructional manual, or their own imagination. For example, consider the step sequence to prepare a *Vegan Mexican Taco* by (a) making vegan ground beef, (b) mixing onions and seasoning, (c) making sauce, (d) making vegan cheese, and (e) wrapping it up. What if none of the videos on making a *Mexican Taco* covers these exact steps, in order? Some recipe might add mashed potatoes (Fig. 1, top row) or make nacho cheese (Fig. 1, bottom row). The problem is that no one video demonstration may show the *exact* steps of interest.

The task to provide a faithful video demonstration for a given sequence of textual step descriptions is technically challenging. Not only is the space of possible procedures very large, but also the wide range of expertise, availability of the tools and objects, and the multistep dependencies

*Equal contribution.
Project page: https://vision.cs.utexas.edu/projects/stitch-a-demo/

of procedural actions all add to the challenge. On the one hand, prior text-to-video methods [3, 6, 34, 38, 45, 62, 66, 67, 70] can retrieve clips that achieve good instantaneous video-text alignment capturing semantic similarity, but they stop short of retrieving consistent multistep depictions. On the other hand, video or image generation [22, 43, 55, 56] shines for creating imaginative outputs beyond the boundaries of any given dataset, but suffers from high computational requirements and hallucinations that reduce realism.

We propose Stitch-a-Demo, a novel method to obtain video demonstrations from multistep descriptions. We *retrieve* clips from multiple videos that best satisfy the input step description while ensuring temporal, visual, and environmental consistency. Our approach employs a novel *procedure evaluator* network, together with well-designed positives and negatives for training that represent correctness and visual consistency constraints. The training method leverages the prior from strong instantaneous video-text signals, and combines them with our designed constraints, to obtain correct video demonstrations from multistep descriptions. Moreover, we propose an adaptive search space reduction for real applications where the search space is large.

We consider three diverse procedural domains. We focus on cooking, which not only represents the single most popular procedural activity in online how-to's today, but also is particularly compelling due to its diversity of tools, ingredients, and physical techniques. Indeed, many influential large datasets and models center around cooking [1, 2, 6, 13, 29, 44, 46, 59, 61, 72]. To underscore the generality of our approach, we further translate the same model to other instructional domains—woodworking and gardening—that also exhibit multiple valid materials and techniques for completing a task. We introduce a large-scale weakly-supervised Stitch-a-Demo training dataset and a manually curated testing dataset. Rigorous quantitative experiments with in-the-wild videos and a human preference study show the effectiveness of our Stitch-a-Demo over state-of-the-art visual demonstration generation and retrieval methods [6, 55, 61, 62].

2. Related Work

Video and language representation learning. Videos are often paired with text captions or narrations that can help learn strong associations between text and video [3, 38, 44, 49, 70]. These video representations can then be used for a variety of downstream tasks, including captioning [27, 35, 47, 65, 77], text-to-video and video-to-text retrieval [12, 13, 17, 41, 44, 45, 70, 79], and action recognition [18, 21, 33, 36, 69]. Most of these tasks consider clips that are typically a few seconds long. Other tasks requiring longer video understanding are action anticipation [3, 20, 23] and procedure planning [8, 11, 64, 78, 80]. However, the query or description in these tasks is a single sentence. Unlike prior work, we learn to provide a video given multistep descriptions, effectively learning associations over long videos with procedural steps.

Learning from instructional videos. Beyond their use in video representation learning, instructional videos are useful for step understanding [5, 39, 58, 84], procedural planning [8, 11, 64, 78, 80], and step grounding [15, 25, 42]. Due to the detailed step descriptions accompanying visual demos in instructional videos, these tasks enable a deep understanding of standard procedures. Recent methods [4, 6, 81] go beyond a single video demonstration and attempt to reason across multiple demonstrations for task graph learning [5, 7, 16, 24, 39, 81, 82]. Despite learning from multiple videos, during inference those methods still handle one video at a time, *e.g.* to perform step forecasting. In contrast, we learn to stitch clips from multiple videos into a cohesive and correct visual demonstration, unlocking a deeper understanding of instructional videos.

Video or image from text descriptions. Obtaining a video or image from text has been studied mostly from two approaches—generation and retrieval. Media generation [9, 22, 26] is used to create *any* image or video from text descriptions—even unrealistic ones. Controlled media generation [9, 40, 60] is also used to edit images and videos to incorporate the desired change. With current generative model capabilities, video generation remains limited to short clips [10, 48, 75]. For instructional content (typically 5-30 minutes), prior works therefore generate a single illustrative image per step [32, 43, 55, 56], showing limited action information. Media generation is also prone to hallucinations, often producing unrealistic step illustrations unlike retrieval. We show that human judges prefer our method compared to the state-of-the-art ShowHowTo for image generation [55].

Video retrieval is the preferred approach when the right answer is known to exist in the candidate set. For instructional videos, retrieval has been used extensively in prior work [12, 13, 17, 41, 44, 45, 70, 76, 79]. Beyond the standard text-to-video retrieval setting, CoVR [62] retrieves a video demonstration based on a reference video/image and a modification text. Limited prior work explores video retrieval to illustrate cooking tasks [6] or recipes [61] (and inversely inferring a recipe from a photo [52]). VidDetours [6] identifies a detour between two cooking videos using a user's language query, e.g., "how do I make this without a blender?", a problem that is interesting but distinct from illustrating a sequence of step descriptions. Recipe2Video [61] creates a slideshow of each step and its image/video/audio demonstration, but is limited to retrieving clips based on rigid and inflexible metrics which impedes correctness, coherence, and object state consistency, as we show in results.

None of the existing retrieval methods is capable of re-

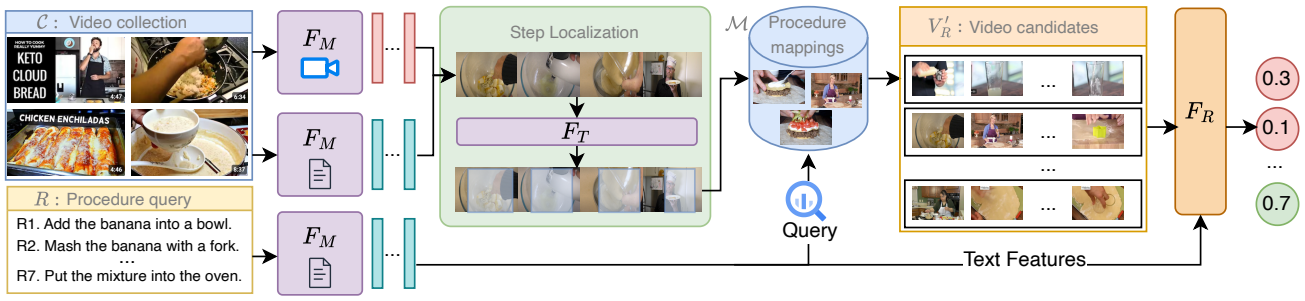

Figure 2. **Method overview.** The videos and the step descriptions in $\mathcal{C}$ are used to create a procedure mapping $\mathcal{M}$, using step localization F_T. The procedure query R and $\mathcal{M}$ give video candidates V'_R. The *procedure evaluator* F_R outputs the likelihood of each candidate.

trieving visually and logically coherent video demonstrations from sequential step descriptions, as we tackle in this work. Furthermore, unlike [6, 52, 61], we go beyond cooking to demonstrate our method on domains like gardening and woodworking.

3. Method

We first formally introduce the task in Sec. 3.1, followed by the model design in Sec. 3.2, the dataset construction idea in Sec. 3.3, and finally the implementation details in Sec. 3.4.

3.1. Task formulation

Given a multistep description for an instructional task, a.k.a. a procedure or a recipe, $R = (r_1, r_2, ..., r_n)$, where r_i is a natural language step description (Fig. 2 bottom left), and a collection of videos $\mathcal{C} = \{V^{(1)}, V^{(2)}, ..., V^{(N)}\}$ (Fig. 2 top left), we want to learn a function $\mathcal{F}$ that finds a video demonstration visually depicting the procedure R. The output video $V_R = (v_1, v_2, ..., v_n)$, is a sequence of video clips, where v_i is a segment from any video $V^{(j)}$, from time instances t_1 to t_2, *i.e.* $v_i = V^{(j)}[t_1 : t_2]$ and $V^{(j)} \in \mathcal{C}$, $t_1 < t_2$. Overall, $\mathcal{F}(R, \mathcal{C}) = V_R$. To recall, videos in $\mathcal{C}$ have diverse human-object interactions [51], object state changes [73], and step dependencies [5], making the task of learning $\mathcal{F}$ both interesting and challenging.

Owing to the scale and the diversity of instructional videos on the internet [44, 83, 84], we create V_R from multiple video demonstrations in $\mathcal{C}$ such that all the procedural steps are correctly shown, with maximum visual consistency. Compared to applying image generation to illustrate a step [32, 43, 55, 56], the proposed design has multiple advantages: it yields complete *video* demonstrations, known to be more useful for human learning [57]; it accounts for visual dependencies between the illustrated steps; and it is less prone to hallucinations and unrealistic outputs.

Clips in V_R need not originate from the same video: $v_i, v_j \in V_R, v_i \in V^{(x)}, v_j \in V^{(y)} \nRightarrow x = y$. We allow clips from different videos in the collection $\mathcal{C}$ to create V_R, since a single video demonstration may not be sufficient to represent an arbitrary procedure R (see Fig. 1). Furthermore, the optimal V_R should stitch together video clips that are not only *correct* (demonstrate the target steps) but also *coherent* (mutually consistent in terms of visual continuity and logic). Our model accounts for both, as we detail next.

3.2. Stitch-a-Demo model design

Next we discuss the model design to learn $\mathcal{F}$. The high-level idea is to first temporally localize step descriptions in all videos to form clip and description tuples, *i.e.* (v, r), followed by creating a candidate set for training. We design a *procedure evaluator* module that determines the likelihood that a sequence of (v_i, r_i), $i = 1, \ldots, n$, is a valid procedure. This model is trained to obtain V_R from $\{\mathcal{C}, R\}$. Fig. 2 shows the overview, and each part is described next.

Encoding videos and procedure text. We use a multimodal encoder F_M (*e.g.* CLIP, InternVideo2 [50, 66, 67]) to represent video and text. We obtain the procedure text feature $\mathbf{r}_i = F_M(r_i)$ for a step r_i, and a video clip feature $\mathbf{v}_i = F_M(v_i)$, where v_i is a video segment for a procedure step. We extract 1 feature from 8 frames per second. The video encodings are averaged over the duration of the video clip to obtain a step video clip feature, consistent with prior work [4, 6, 39]. We do not train F_M.

Localizing a clip in a video. Retrieving clips for V_R requires finding video clips associated with a procedural step r. We use a temporal localization function F_T (*e.g.* [15, 16, 30, 42]) to find the start and end time of a step in a video. Specifically, let $R^{(j)} = (r'_1, r'_2, ...r'_3)$ describe the procedure of $V^{(j)}$ in text. $[t_1, t_2] = F_T(V^{(j)}, r')$ which we use to obtain the clip $v' = V^{(j)}[t_1 : t_2]$. This process yields a pool of procedure steps and clips $\mathcal{P} = \{(r', v') \mid r' \in R^{(j)}, v' \in V^{(j)}, V^{(j)} \in \mathcal{C}, r' \sim v'\}$. From $\mathcal{P}$ and the query R, we construct a procedure mapping $\mathcal{M} = \{(r, v') \mid r \in R, (r', v') \in \mathcal{P}, r \sim r'\}$. We use DropDTW [15] as the pre-trained F_T, and keep it frozen; any improvement there will only improve our model's performance.

***Procedure evaluator* module.** The above modules help us obtain a map of a procedure's steps and corresponding clips in the video collection $\mathcal{C}$. Next, we propose a *procedure evaluator* F_R that finds the probability of the correct-

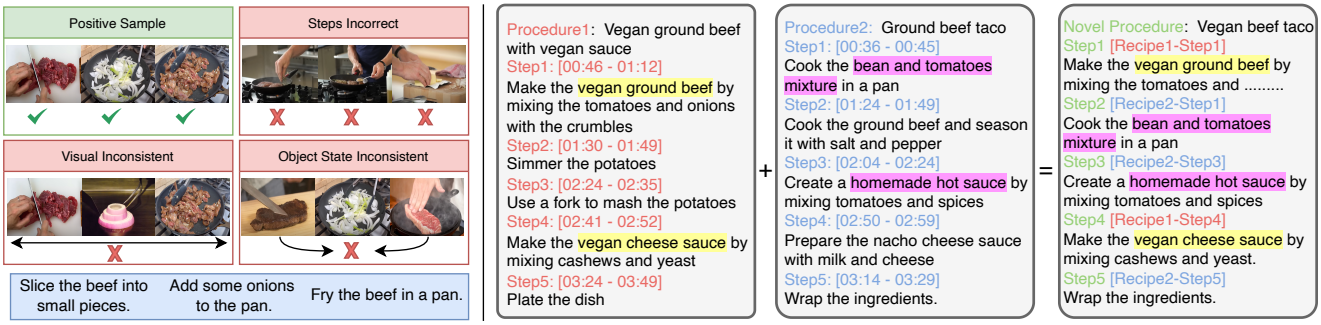

Figure 3. **Examples of hard negatives and procedure combination.** We design negative samples that violate step correctness, visual continuity, and object state continuity for contrastive learning (left). We show an example of combining step descriptions from n (here $n = 2$) video demonstrations into a novel procedure, using an LLM [19] (right). The novel procedure mixes steps from both descriptions.

ness of a candidate V_R, given the procedure steps. Specifically, the procedure correctness is given by

$$F_R\left((v'_1, v'_2, ..., v'_n) \;\middle|\; v'_i \sim r_i, (r_i, v'_i) \in \mathcal{M}\right) \in [0, 1],$$

where $v \sim r$ denotes the clip v shows the step r and $(v'_1, v'_2, ..., v'_n)$ is a full "stitched" candidate sequence for V_R. F_R has a transformer encoder that takes as input the concatenated features $F_M(r_i)$ and $F_M(v_i)$—one per procedural step. We take the output of the transformer encoder corresponding to the CLS token, followed by a linear layer to output a probability, $F_R()$.

A key novelty of our method is how we train the procedure evaluator. Instead of imposing heuristics to determine which are good stitched videos, our insight is to automatically generate a large-scale weakly supervised dataset with hard negatives that encourage correctness and coherence.

Sampling negatives. Building on ideas in contrastive learning [45, 50, 70], we create a hard negative set by modifying correct video demonstrations in a targeted way. Each modification violates a constraint for correctness or visual coherence. We describe each constraint first, followed by the modification we perform to obtain negative samples.

- **Step and goal correctness**: All clips in V_R must accurately represent the corresponding step description in R and also contribute meaningfully to the goal of R (and V_R). That is, $v_i \sim r_i \; \forall \, i$. For example, if r_i is *"add salt to the chicken broth"*, then v_i should demonstrate sprinkling salt into the soup, but not performing other actions on the broth or adding salt to something else, *e.g.* rice.
  Violating step and goal correctness: Given a video demonstration V_R, we create a negative sample V'_R by replacing any clip $v \in V_R, v \sim r$ with randomly selected v' such that $v' \nsim r$. Furthermore, to make the negatives harder, we ensure the source video of v' contributes to V_R, *i.e.* $v' \in V^{(j)} \implies \exists v'' \in V^{(j)}, v'' \in V_R$. These constraints lead to a strong negative from the same video sources, violating the step correctness. See Fig. 3 (left).
- **Visual continuity**: We aim to minimize the number of distinct video sources when constructing V_R. A new video must be selected only if the previously selected video is insufficient for a step r_i. Specifically, if $v_i \in V^{(j)}, v_i \sim r_i$, and there are two candidates for the next procedure step, *i.e.* $v \sim r_{i+1}, v' \sim r_{i+1}$, but $v \in V^{(j)}, v' \notin V^{(j)}$, then v should be chosen as the next demonstration. Note that many candidate subsequences will *not* originate from the same video; hence we do have positives that cross video boundaries.
  Violating visual continuity: If a video demonstration V_R contains three consecutive steps from the same source video, we form a negative by replacing the middle clip with a similar demonstration from a different source video, while still ensuring step correctness. Specifically, if $v_i \in V_R, v_i \sim r_i$, and $v_i \in V^{(j)} \; \forall i \in \{k, k+1, k+2\}$, we replace v_{k+1} with $v' \notin V^{(j)}, v' \sim r_{k+1}$. See Fig. 3.
- **Object state continuity**: An object must not be in a state that has undergone an irreversible transformation at a previous time. For example, there should not be a step with a *whole onion* after a step showing *onions* being chopped, or a step showing *unsanded wood* after a step showing the wood being sanded down.
  Violating object state continuity: We construct hard negatives in this category by changing $v_x, v_y \in V_R$ to $v, v' \in V^{(j)}$, such that v occurs before v' in $V^{(j)}$, and $v \sim r_x, v' \sim r_y$, but $x > y$. That is, the clips in the negative sample do not follow the temporal order in the original source video $V^{(j)}$. Even though some steps are interchangeable in procedures, enforcing this constraint is helpful given the perceptual damage in "undoing" a permanent transformation; see Sec. 4 and Supp. where we experimentally validate the usefulness of violating object state continuity for training negatives.

In summary, we consider various realistic constraints when combining demonstrations from multiple procedures. Our contrastive setup trains with these hard negatives. We show the effectiveness of each of these constraints in the results and discuss robustness to label noise below.

Training objective and inference. A correct procedure

demonstration has the ground truth F_R value of 1; 0 otherwise. During training, we use the standard Binary Cross Entropy (BCE) training objective. The choice is consistent with the output of the procedure evaluator F_R and the ground truth binary labels.

During inference, we have a set of video candidates $V'_R = (v'_1, v'_2, ..., v'_n)$ and the chosen video V_R is

$$\underset{(v'_1, v'_2, ..., v'_n)}{\arg\max} \; F_R(V'_R \mid v'_i \sim r_i).$$

That is, the candidate with the highest probability from the procedure evaluator.

Adaptive search space reduction. Finally, we discuss adaptive search space reduction for practical implementation. For a short clip or naive full-video retrieval, the candidate set scales as $O(N)$ for N videos. On the other hand, if video clips are allowed to be from distinct videos in an M-step procedure with an average of K clips in a video, the candidate set scales as $O((KN)^M)$. Thus, we propose an adaptive search space reduction technique that can capture the ground truth in a much smaller candidate set, as follows.

We first construct a collection of sets $\mathcal{S}$ where an element is defined as

$$S_i := \{(x, v) \mid x \in \mathbb{Z}, v \in V^{(i)}, v \sim r_x\}.$$

That is, each set contains procedural step indices that have a matching video clip in the i^{th} video. For example, if $V^{(1)}$ has clips v_1 and v_2 that match with steps r_1 and r_2 from the query R, then $S_1 = \{(1, v_1), (2, v_2)\}$. This problem is the same as a set cover problem [37]. The task is to find a subset of $\mathcal{S}$, *i.e.* a collection of S_i, such that they cover all steps in R with minimum number of set changes, *i.e.* video source changes. We use the greedy solution of this problem [37] to select top-K such set covers. We also use this method in constructing the *distractor set* for retrieval performance evaluation (Sec. 4). We show the effectiveness of this search space reduction in Sec. 4. See Supp. for details.

3.3. Stitch-a-Demo dataset

To train $\mathcal{F}$, we curate a large-scale automatic training data and evaluate the model with a suite of testing data. Each training instance consists of a (R, V_R) pair: a list of procedural steps and their associated video clips. Positive pairs stem from both real existing instructional videos as well as novel procedures we generate, as described next. Negatives are alterations of those positives formed as in Sec. 3.2.

Weakly-supervised training set. To augment the real positives, the high-level idea is to use all the narrations in the instructional video datasets, along with an existing language model, to create *realistic* weakly-supervised procedures by mixing steps from different demonstrations. This training data augmentation is essential to help the model learn to combine video clips from different demonstrations.

We have a collection $\mathcal{C}$ containing video demonstrations including HowTo100M [44], COIN [58], CrossTask [84], or HT-Step [2]. These datasets generally do not have annotations for procedural steps R; only [2] has a small scale cooking recipe description that we use for testing, see below. However, they have narrations that are converted to text using ASR, which we use to obtain procedural step descriptions. The ASR text is punctuated [25] and converted to sentences. Then, following [6], we use a language model (Llama-3.1 70B Instruct [14]) to convert the narration sentences to step-level, timestamped summaries. Next, we find similar video demonstrations using a sentence similarity score on the summaries (MPNet [54]), and choose pairs, triplets and quartets of procedures that have a pairwise similarity above a threshold $c = 0.8$.

As the last step, we provide those summary tuples to the language model and ask it to create a *valid* sequence of steps, effectively creating *novel* procedures. The question follows the format *"Create a new procedure by combining steps from the provided procedure summaries. Choose new steps from a different procedure only if the current procedure cannot be used alone. Procedures: ..."*. See Supp. for the full prompt, which promotes correctness, visual consistency, and accounts for object states. The summaries have the time duration of each step, thus the *novel* procedures use the corresponding start and end time for each video. We show the efficacy of using LLMs in Supp.

Overall, we obtain a weakly-supervised dataset $\mathcal{D}_w = \{(R, V) \mid R = (r_1, ..., r_n), V = (v_1, ..., v_n), \exists x \text{ s.t. } v_i = V_x[t_1, t_2], V_x \in \mathcal{C}\}$. See an example in Fig. 3 (right). As discussed in Sec. 3.1, clips within the same procedure can come from different source videos. We acknowledge that *training* samples in $\mathcal{D}_w$ might contain noise due to the inaccuracies of the LLM. However, the value of unlocking the larger dataset outweighs the risk of introducing noisy training signals, as we show below. Manually examining a random sample of labels, we find 75% to be high quality.

To reiterate, we use the idea of mixing procedures (recipes) to (a) create a larger training set, and (b) encourage the model to predict steps from distinct videos, if needed. The hard negatives, discussed in Sec. 3.2, are sampled from $\mathcal{D}_w$ for contrastive training. We validate this dataset $\mathcal{D}_w$ with ablations in Supp.

Stitch-a-Demo testing sets. We create a suite of testing sets to systematically evaluate all aspects of the model design and assumptions. Our testing suite has three types of samples: (1) augmented procedures formed using the process above (w for weakly supervised), (2) a manually annotated test set $\mathcal{D}_d$ (for detour) derived from prior work [6], and (3) standard unmixed demonstrations from original videos ($\mathcal{D}_v$, for full video). These three are complementary, offering tradeoffs in realism and strength of ground truth vs. scale and our control in generating samples. We

Method	Cooking								Woodworking				Gardening			
	SaD-MC		SaD-VD		HT-Step		COIN, CT		SaD-MC		COIN,CT		SaD-MC		COIN,CT	
	MR↓	R@50	MR↓	R@50	MR↓	R@50	MR↓	R@50	MR↓	R@50	MR↓	R@50	MR↓	R@50	MR↓	R@50
CoVR [62]	193	0.04	132	0.04	161	0.12	97	0.25	29	0.34	37	0.34	31	0.24	26	0.30
VidDetours [6]	124	0.21	80	0.31	125	0.22	37	0.61	30	0.28	31	0.24	34	0.24	40	0.24
Text-only	108	0.32	76	0.36	123	0.26	78	0.36	44	0.22	31	0.20	51	0.16	58	0.32
InternVideo [66]	36	0.55	8	0.71	68	0.42	19	0.67	38	0.34	43	0.28	45	0.12	45	0.28
Recipe2Video [61]	125	0.21	50	0.50	93	0.29	26	0.68	40	0.30	75	0.04	48	0.02	76	0.12
Ours	**3**	**0.84**	**3.5**	**0.91**	**40**	**0.56**	**6**	**0.88**	**24**	**0.36**	**30**	**0.38**	**26**	**0.36**	**25**	**0.36**

Table 1. **Results on video demonstration retrieval.** Comparison of the performance of our method against strong retrieval-based baselines and prior work using median rank (MR) and recall (R@50). The first two methods (CoVR [62] and VidDetours [6]) are relevant retrieval models, though not specifically designed for this task. Our method significantly outperforms all methods on all metrics, for all test datasets and three diverse procedural domains. SaD=Stitch-a-Demo. SaD-VD [6] and HT-Step [2] are available only for cooking. See text.

describe each in detail next.

Firstly, we have a held-out set from $\mathcal{D}_w$ used as a test set; we call it **Stitch-a-Demo-MC** set for Mixed Clips. This weakly supervised set is large-scale, but naturally incurs some noise (e.g., some step descriptions may not match their associated visual clip due to errors in the LLM summaries). While such noise is fine during training, it is a shortcoming for testing, and hence we complement with two more strongly ground-truthed test sets, defined next.

Secondly, we design a test set $\mathcal{D}_d$ called **Stitch-a-Demo-VD** that strings together the manually verified annotations from VidDetours [6] to compose a clean test set with minimal manual verification effort. See Supp. for details. Like $\mathcal{D}_w$, this test set allows evaluating $\mathcal{F}$'s ability to choose clips from different procedures, but unlike $\mathcal{D}_w$ it is manually verified and is available only for cooking videos [6].

Finally, we use **standard instructional video datasets** $\mathcal{D}_\mathbf{v}$ (based on CrossTask [84], COIN [58], and HT-Step [2]) for testing. We provide the step descriptions for a given video demonstration, and expect the model to *recover* all the video clips from the same video. In cases where the dataset is not specifically annotated for detailed steps (*e.g.* in [58, 84]), we use a language model (similar to $\mathcal{D}_w$ creation) to summarize the ASR text into procedure steps. All the output summary steps in these datasets are manually verified for correctness by us. Note that the ground truth clips in these datasets come from the *same* video in $\mathcal{C}$.

In short, sequences in both Stitch-a-Demo-MC and Stitch-a-Demo-VD contain video clips from multiple distinct videos in $\mathcal{C}$. Stitch-a-Demo-MC is auto-created and large-scale, whereas Stitch-a-Demo-VD is manually created and small scale. Meanwhile $\mathcal{D}_v$ is a large-scale source of real (ground truth) videos, but does not require the models to stitch across videos as needed in the ultimate use case.

3.4. Data and implementation details

Data sources and dataset statistics. Sourced from HowTo100M [44], the weakly supervised training set $\mathcal{D}_w$ and the testing set Stitch-a-Demo-MC consist of $446,623$ and $105,742$ samples across cooking, woodworking, and gardening, with $|\mathcal{C}| = 323,177$ and $2,857$, respectively. (See Supp. for sample counts per domain.) The test set SaD-VD derived from VidDetours [6] contains 100 recipes from 235 unique videos. There are only cooking annotations in [6]. Finally, the test data $\mathcal{D}_v$ from COIN [58] and CrossTask [84] 457 and 942 across the three domains, and HT-Step [2] contains $1,087$ cooking videos. HT-Step [2] contains step descriptions for cooking videos only from WikiHow [68], allowing evaluation with original human-written step descriptions. COIN [58] and CrossTask [84] evaluates performance with step descriptions auto-summarized from narrations. We group COIN and CrossTask into a test set since they are testing the same aspects (see Supp. for per dataset performance).

Method and training details. We use InternVideo2 [67] as the multi-modal encoder F_M. For step localization F_T, we use DropDTW [15] for its flexibility with extra or missing steps, and its reproducible codebase. The features used in [45] are trained on HowTo100M [44], thus, are used without re-training. For the procedure evaluator F_R, we use positional encoding and transformer encoder with 4 layers and 8 attention heads followed by an MLP layer with 1 hidden layer. Due to the diverse nature of the domains, we train a separate F_R for each of cooking, gardening, and woodworking. We optimize F_R for 10 epochs with Adam [31] on 8 Quadro RTX 6000. We set the learning rate as 3×10^{-4} and the batch size as 24. The F_M output, F_R input, and F_R hidden dimension are all 768.

4. Experiments

We first describe the baselines, testing setup, and metrics and then, the quantitative results. Next, we show the results of a human preference study, followed by ablations and results using our adaptive search space reduction technique. We also show qualitative results, including failure cases.

Baselines. We compare our method against strong base-

Figure 4. **Qualitative results.** Our method correctly visualizes the step descriptions (top), compared to prior work. The second to the fourth rows show representative outputs in cooking, woodworking, and gardening. Our method correctly shows video clips from two video sources. Each of the video sources alone cannot correctly demonstrate all the step descriptions. The last row contains some **failure cases**, showing the difficulty of the task. Here each keyframe represents a clip v; see project page for actual videos and additional failure analysis.

lines. We use the same text and video encoder across all baselines, when applicable. All the baselines are trained on the same data as our method for a fair comparison. Each baseline differs in how it selects the sequence of video clips v_i to associate with each step $r_i \in R$.

- **Text-only** computes the average similarity between the ASR transcript of each clip and each step instruction without using the visual cues.
- **InternVideo** [67] is a state-of-the-art vision-language model. This baseline computes the average similarity between the clip and the step instruction for each step. The baseline uses the same F_M and F_T as our model, but lacks our F_R transformer, i.e., our procedure evaluator.
- **Recipe2Video** [61] leverages manually-designed metrics that include temporal consistency, information coverage, and cross-modality retrieval to retrieve procedure clips. We reproduced the authors' prior results to ensure correctness of our implementation.
- **CoVR** [62] is a state-of-the-art method for composed video retrieval. Here it retrieves the clip based on the current step and the clip retrieved for the previous step.
- **VidDetours** [6] is a state-of-the-art method that we repurpose for our task. It retrieves the clip based on an original video segment and a user query. For each step, we set the step instruction as the user query and use the previous retrieved clip as the original video segment. We perform this operation sequentially for all procedure steps.

Further implementation details are available in Supp.

Test setup and metrics. For every test instance, we create a set of 499 incorrect *distractors* and measure the correct procedure retrieval performance out of 500 samples. The distractors contain hard negatives, including samples that violate a constraint, and candidates from the adaptive reduced search space (Sec. 3.2). Overall, these negative instances are carefully chosen to represent a wide range of possibilities. See Supp. for details of the negative set construction.

We use standard retrieval metrics *recall@50* and the median rank (MR). Recall is higher the better, whereas median rank is lower for better methods.

Results. Table 1 shows the results. Our method signifi-

Ours vs	Win rate (%) Step	Goal	Quality	Total
Recipe2Video [61]	0.77	0.74	0.74	0.77
InternVideo [66]	0.75	0.72	0.78	0.75
ShowHowTo [55]	0.94	0.94	0.85	0.98

Table 2. **Human preference results.** Our method is preferred by human judges compared to existing retrieval (first two rows) and generation (third row) methods. (Step/Goal: Step/Goal faithfulness, Quality: Visual quality, Total: Overall preference).

cantly outperforms all the strong baselines, on all datasets and metrics and across all three domains. Our gain is up to 29% better in recall and 33 ranks better in MR, compared to the second-best method, InternVideo [66], which lacks our key innovation, the learned procedure evaluator. We also compare with HiREST [76], a hierarchical retrieval baseline, and our model outperforms it by 0.42 and 0.32 in recall on SaD-VD and HT-Step, respectively.

Notably, our method is superior for both when the ground truth video contains demonstrations from multiple source videos, *i.e*. in Stitch-a-Demo-MC and Stitch-a-Demo-VD, as well as in cases where we recover the demonstration from step descriptions, *i.e*. in HT-Step [2] and COIN, CrossTask [58, 84]. We attribute this strong performance to our training design that incorporates strong negatives—thus allowing the model to generate output that is *correct* and *visually coherent*.

Fig. 4 shows some qualitative results, including failure cases. We see that our method correctly retrieves the clip showing a star shaped cutter, compared to Recipe2Video [61] and the generative ShowHowTo [55] (top row). We see our method correctly chooses video clips for given step descriptions, *e.g*. skipping adding beans, and showing tomato purée from another source video (second row). Our method also successfully retrieves clips for making a wall-mounted painted desk (third row) and preparing a plant mixure for transplanting (fourth row). Finally, we show some failure cases where the retrieval misses some small objects like a toothpick, showcasing the overall difficulty of satisfying the exact procedure steps.

Human preference study. We also conduct a human preference study to compare with the strongest retrieval and generation methods [55, 61, 66]. The study considers the cooking subset since typically, we found more people with cooking experience, over gardening and woodworking. This complements the results above established with automatic metrics. We use the same settings of InternVideo [66] and Recipe2Video [61] as above. For ShowHowTo [55], we follow its original setup and prompt it with the middle frame of the first video clip in the ground truth. All its remaining demonstration images are created with the first frame and the step descriptions R as the input.

We evaluate all the methods on four axes—step faithfulness, goal faithfulness, visual quality, and overall preference. Every sample is annotated by three subjects unrelated to this project. We compare two methods at a time (one of them always ours), for up to 60 samples per pair. See Supp.

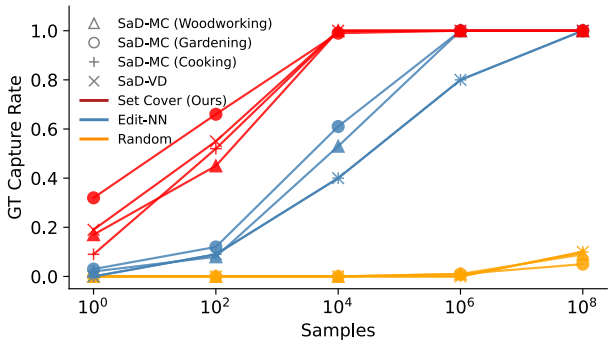

Figure 5. **Search space reduction.** Using the effective set cover algorithm, the ground truth (GT) is captured in the candidate set with high probability, even with small sample set sizes. See text.

Tab. 2 shows that our method is preferred over all competing approaches, outperforming retrieval-based methods and the generative ShowHowTo [55]. This study supports the practical value of our stitching framework: subjects with varying cooking experience (1–10 years) preferred our method 83:17 over original video demonstrations, showing that the naturalness of real videos does not offset their limitations in accurately portraying the target multistep task.

Adaptive search space reduction. Fig. 5 shows the percentage of ground truth retrieved across all test cases as a function of the number of retrieved cases per query. Our set cover algorithm captures the ground truth in the candidate set with high probability, even for low values of K, making it comparable to linear scaling. We see the same trend in Stitch-a-Demo-MC across three domains and -VD; the algorithm always captures ground truth instances containing only one video source, *i.e*. in $\mathcal{D}_v$. We perform better than other methods—randomly selecting clips, and edited-NN, which finds the nearest full video and replaces video clips from other neighbors. Overall, set cover helps in making our proposed method feasible for real applications.

5. Conclusion

We propose a novel method Stitch-a-Demo that stitches together video demonstrations that illustrate multistep textual descriptions of procedural tasks. Our method incorporates a novel procedure evaluator, a weakly-supervised large-scale train and clean test data, hard negatives that improve retrieval, and an efficient set cover approach. Our method outperforms strong baselines up to 29%, and human raters prefer our method over SOTA generation and retrieval methods. In the future, we will explore hybrid methods to integrate the retrieved clips with controlled generation, as well as ways to inject preferences into the illustrations such as demonstration speed or language style.

Acknowledgement

Research supported in part by the UT Austin IFML NSF AI Institute. We thank all the annotators for their efforts and the lab members in the UT Austin Computer Vision Group for helpful discussions.

References

[1] Mohamed A Abdelsalam, Samrudhdhi B Rangrej, Isma Hadji, Nikita Dvornik, Konstantinos G Derpanis, and Afsaneh Fazly. Gepsan: Generative procedure step anticipation in cooking videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2988–2997, 2023. 2

[2] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. In *NeurIPS*, 2023. 2, 5, 6, 8, 3

[3] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23066–23078, 2023. 2

[4] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. In *Advances in Neural Information Processing Systems*, pages 67833–67846. Curran Associates, Inc., 2023. 2, 3

[5] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[6] Kumar Ashutosh, Zihui Xue, Tushar Nagarajan, and Kristen Grauman. Detours for navigating instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18804–18815, 2024. 2, 3, 5, 6, 7, 4

[7] Siddhant Bansal, Chetan Arora, and CV Jawahar. United we stand, divided we fall: Unitygraph for unsupervised procedure learning from videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6509–6519, 2024. 2

[8] 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

[9] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 2

[10] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024. 2

[11] 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

[12] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021. 2

[13] 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: collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. 2

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 5, 1

[15] Mikita Dvornik, Isma Hadji, Konstantinos G Derpanis, Animesh Garg, and Allan Jepson. Drop-DTW: Aligning common signal between sequences while dropping outliers. *Advances in Neural Information Processing Systems*, 34: 13782–13793, 2021. 2, 3, 6

[16] Nikita Dvornik, Isma Hadji, Hai Pham, Dhaivat Bhatt, Brais Martinez, Afsaneh Fazly, and Allan D Jepson. Flow graph to video grounding for weakly-supervised multi-step localization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 319–335. Springer, 2022. 2, 3

[17] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021. 2

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2

[19] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 4

[20] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13515, 2021. 2

[21] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16102–16112, 2022. 2

[22] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Factorizing text-to-video generation by explicit image conditioning. In *European Conference on Computer Vision*, pages 205–224. Springer, 2024. 2

[23] 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

[24] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *CVPR*, 2024. 2

[25] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916, 2022. 2, 5

[26] Hexiang Hu, Kelvin CK Chan, Yu-Chuan Su, Wenhu Chen, Yandong Li, Kihyuk Sohn, Yang Zhao, Xue Ben, Boqing Gong, William Cohen, et al. Instruct-imagen: Image generation with multi-modal instruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4754–4763, 2024. 2

[27] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17980–17989, 2022. 2

[28] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

[29] Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman. Epic-sounds: A large-scale dataset of actions that sound. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2

[30] Xun Jiang, Xing Xu, Jingran Zhang, Fumin Shen, Zuo Cao, and Heng Tao Shen. Semi-supervised video paragraph grounding with contrastive encoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2466–2475, 2022. 3

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 6

[32] Bolin Lai, Xiaoliang Dai, Lawrence Chen, Guan Pang, James M Rehg, and Miao Liu. Lego: L earning ego centric action frame generation via visual instruction tuning. In *European Conference on Computer Vision*, pages 135–155. Springer, 2024. 2, 3

[33] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *arXiv preprint arXiv:2201.09450*, 2022. 2

[34] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020. 2

[35] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 2

[36] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 2

[37] Sherry Liang, Khalid Alanazi, and Kumail Al Hamoud. Set covering problem. *Cornell University Computational Optimization Open Textbook. Cornell University,[online document]*, 2020. 5

[38] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. In *NeurIPS*, 2022. 2

[39] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. 2, 3

[40] Yuanze Lin, Yi-Wen Chen, Yi-Hsuan Tsai, Lu Jiang, and Ming-Hsuan Yang. Text-driven image editing via learnable regions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7059–7068, 2024. 2

[41] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[42] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. Learning to ground instructional articles in videos through narrations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15201–15213, 2023. 2, 3

[43] Sachit Menon, Ishan Misra, and Rohit Girdhar. Generating illustrated instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6274–6284, 2024. 2, 3

[44] 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. 2, 3, 5, 6

[45] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 2, 4, 6

[46] Tushar Nagarajan and Lorenzo Torresani. Step differences in instructional video. In *CVPR*, 2024. 2

[47] Van-Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. Grit: Faster and better image captioning transformer using dual visual features. *arXiv preprint arXiv:2207.09666*, 2022. 2

[48] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 2

[49] Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5285–5297, 2023. 2

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 4

[51] Ivan Rodin, Antonino Furnari, Kyle Min, Subarna Tripathi, and Giovanni Maria Farinella. Action scene graphs for long-form understanding of egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18622–18632, 2024. 3

[52] Amaia Salvador, Michal Drozdzal, Xavier Giro i Nieto, and Adriana Romero. Inverse cooking: Recipe generation from food images. In *CVPR*, 2019. 2, 3

[53] 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*, 45(6):7836–7852, 2022. 2

[54] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020. 5

[55] Tomáš Souček, Prajwal Gatti, Michael Wray, Ivan Laptev, Dima Damen, and Josef Sivic. Showhowto: Generating scene-conditioned step-by-step visual instructions. *arXiv preprint arXiv:2412.01987*, 2024. 2, 3, 8

[56] Tomáš Souček, Dima Damen, Michael Wray, Ivan Laptev, and Josef Sivic. Genhowto: Learning to generate actions and state transformations from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[57] Dawn Surgenor, Lynsey Hollywood, Sinéad Furey, Fiona Lavelle, Laura McGowan, Michelle Spence, Monique Raats, Amanda McCloat, Elaine Mooney, Martin Caraher, et al. The impact of video technology on learning: A cooking skills experiment. *Appetite*, 114:306–312, 2017. 1, 3

[58] 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. 2, 5, 6, 8, 3, 4

[59] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Larina, Diane Larlus, Dima Damen, and Andrea Vedaldi. EPIC Fields: Marrying 3D Geometry and Video Understanding. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023. 2

[60] Shuyuan Tu, Qi Dai, Zhi-Qi Cheng, Han Hu, Xintong Han, Zuxuan Wu, and Yu-Gang Jiang. Motioneditor: Editing video motion via content-aware diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7882–7891, 2024. 2

[61] Prateksha Udhayanan, Suryateja Bv, Parth Laturia, Dev Chauhan, Darshan Khandelwal, Stefano Petrangeli, and Balaji Vasan Srinivasan. Recipe2video: Synthesizing personalized videos from recipe texts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2268–2277, 2023. 2, 3, 6, 7, 8, 4

[62] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. Covr: Learning composed video retrieval from web video captions. *arXiv preprint arXiv:2308.14746*, 2023. 2, 6, 7, 3, 4

[63] Beichen Wang, Juexiao Zhang, Shuwen Dong, Irving Fang, and Chen Feng. Vlm see, robot do: Human demo video to robot action plan via vision language model. *arXiv preprint arXiv:2410.08792*, 2024. 1

[64] Hanlin Wang, Yilu Wu, Sheng Guo, and Limin Wang. Pdpp: Projected diffusion for procedure planning in instructional videos. *arXiv preprint arXiv:2303.14676*, 2023. 2

[65] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 2

[66] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning, 2022. 2, 3, 6, 8, 4

[67] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024. 2, 3, 6, 7

[68] WikiHow. Wikihow. https://www.wikihow.com, 2025. 6

[69] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022. 2

[70] 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. 2, 4

[71] Xin Xu, Kun Qian, Bo Zhou, Shenghao Chen, and Yitong Li. Two-stream 2d/3d residual networks for learning robot

manipulations from human demonstration videos. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3353–3358, 2021. 1

[72] Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18493–18503, 2024. 2

[73] Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18493–18503, 2024. 3

[74] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. *arXiv preprint arXiv:1809.00812*, 2018. 2, 3

[75] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2

[76] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23056–23065, 2023. 2, 8

[77] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021. 2

[78] 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

[79] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. *arXiv preprint arXiv:2205.00823*, 2022. 2

[80] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. *arXiv preprint arXiv:2303.17839*, 2023. 2

[81] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *CVPR*, pages 10727–10738, 2023. 2

[82] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[83] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598, 2018. 3

[84] 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. 2, 3, 5, 6, 8, 4