STEPs: Self-Supervised Key Step Extraction and Localization from Unlabeled Procedural Videos

Anshul Shah[1]   Benjamin Lundell[2]   Harpreet Sawhney[2]   Rama Chellappa[1]

[1] Johns Hopkins University   [2] Microsoft Mixed Reality

{ashah95, rchella4}@jhu.edu {benjamin.lundell,harpreet.sawhney}@microsoft.com

Abstract

We address the problem of extracting key steps from unlabeled procedural videos, motivated by the potential of Augmented Reality (AR) headsets to revolutionize job training and performance. We decompose the problem into two steps: representation learning and key steps extraction. We propose a training objective, Bootstrapped Multi-Cue Contrastive (BMC2) loss to learn discriminative representations for various steps without any labels. Different from prior works, we develop techniques to train a light-weight temporal module which uses off-the-shelf features for self supervision. Our approach can seamlessly leverage information from multiple cues like optical flow, depth or gaze to learn discriminative features for key-steps, making it amenable for AR applications. We finally extract key steps via a tunable algorithm that clusters the representations and samples. We show significant improvements over prior works for the task of key step localization and phase classification. Qualitative results demonstrate that the extracted key steps are meaningful and succinctly represent various steps of the procedural tasks. Our code can be found at https://github.com/anshulbshah/STEPs.

1. Introduction

Rapid shifts in technology and business models have led to a mismatch between the skills needed by employers and the skills possessed by the labor force. It has been estimated that this mismatch will reduce manufacturing output by $2.4 trillion over ten years in the US alone [31, 33]. Increased attention has been placed on effective methods of "reskilling" workers [2]. Unfortunately, reskilling will not be easy: human expertise in performing a complex task takes years of training and mastery of domain-specific knowledge [10, 27].

Augmented Reality (AR) headsets can play an important role in collective reskilling efforts. AR headsets are known to improve the efficiency of front line workers during training
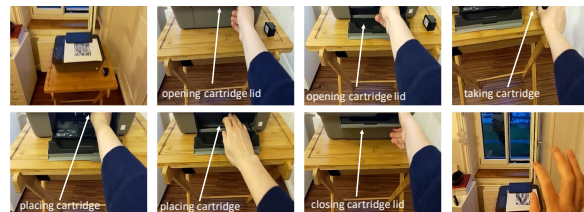

Figure 1. An illustrative example of automatically generated key steps using our approach STEPs for the task of changing the cartridge in a printer. Data was captured using a Microsoft HoloLens 2 and extracted using a publicly available repository. Our approach leads to plausible key steps for sub-tasks of 'opening cartridge lid', 'taking cartridge', 'placing cartridge' and 'closing cartridge lid' and 'end recording'. Note that the associated labels and arrows are for visualization purposes only and were not used for training.

and on the job, across industries as diverse as food service, manufacturing, medicine, and warehousing [1,14,74,76]. AR plays a strikingly similar role across these diverse use cases: to assist the user in completing a complex task, the headset renders a sequence of visual cues on real-world objects.

Our approach focuses on extracting key-steps of a complex task which is the most crucial component needed for automatic AR content creation. We employ a "learning-from-observation"-style framework [60], where an instructor is recorded while performing a complex task. The goal is to automatically parse the recording into *key steps* (KSs) that succinctly represent the complete task. This greatly streamlines the content creation process, as the trainer no longer has to manually edit the recording to find the key steps.

Consider the task of changing the cartridge in a printer. Using tools from the public repository [19], we captured data on a Microsoft HoloLens 2 of an "expert" undertaking this task. Using multiple cues – hand pose, head pose, eye gaze and first person video – we automatically generate the key-steps shown in Figure 1. Using multiple cues when observing experts perform procedural tasks is important when generating training materials for novices [12, 28, 32].

The key step extraction problem for complex procedural

tasks is challenging: (1) Recordings of tasks performed by experts are limited in number; (2) Supervision for key steps is hard due to the subjective nature of what constitutes a key step; (3) There are no large-scale datasets for real world procedural tasks. Unlike typical web-crawled videos used in video representation learning, procedures are often minutes long; and (4) Visual information alone might not adequately represent all the information in a scene. None of the prior works address all of the challenges mentioned above.

Our solution involves a two-stage approach to the KS extraction problem. First, we train a task-specific model to produce a context-rich feature vector for each frame of the recording without any labels. To overcome paucity of data, our Multi-cue key steps (STEPs) approach, (Figure 2 and Sec. 3), uses multiple temporal feature sequences corresponding to different cues as input. Different from prior works in self-supervised learning, we only train a per-modality temporal model which is applied on pre-extracted 'raw' features using a pre-trained feature extractor. To learn rich features, we propose a novel loss termed BMC2 (Bootstrapped Multi-Cue Contrastive Loss). The loss enforces temporally adapted features from each modality within the window to be close in a common latent space. Instead of using a fixed temporal window as used by related works, our loss first bootstraps a temporal window around each anchor using 'raw' features thus reducing the effect of false negatives. After training, the per-frame representations are clustered and KSs are sampled from the clusters. Use of off-the-shelf features enables fast temporal encoder training with extremely long temporal contexts; we can train a model on 17 Meccano bike assembly videos [8] for 300 epochs in under 6 minutes on a single GPU. Unlike recent works in self-supervised video representation learning [23, 37, 56], we do *not* rely on alignment between multiple recordings as a proxy task. *Inter-video alignment* as a learning loss requires multiple recordings of the same task, thus limiting its practical applicability to the problem of learning from few expert recordings.

Our approach is suitable for not only multi-modal data computed by first-person wearable devices such as Oculus Quest, Microsoft HoloLens and others, but also conventional procedural task videos. In this case, we use the visual stream to extract rich targeted modalities like optical flow for training.

We provide quantitative and qualitative results on seven datasets (both first-person and third person). Unfortunately, KS extraction is a subjective task and there are no publicly available large-scale annotated procedural learning datasets to evaluate KSE. Thus for quantatitive benchmarking, we compare our learned representations for the proxy task of Key Step localization (KSL). On that task, we outperform [7] by 3-18 points on F1 score. We also provide results on phase classification and outperform prior works by a wide margin.

In summary, following are our key contributions:

1. We propose an intra-video SSL-approach for procedure learning to train a light-weight temporal encoder with pre-extracted multi-cue features. We can train on long temporal sequences (1000s of frames) unlike prior works that train expensively large backbones.
2. We employ a novel loss, termed BMC2, that bootstraps information from pre-extracted multi-cue features to create a temporal window around an anchor, which is used to force representations from multiple modalities to be close in a common representation space. To the best of our knowledge, ours is the first work to employ multiple synchronized sensor and vision-derived cues often available in AR devices for key step extraction.
3. We obtain superior performance compared to state-of-the-art on several tasks across various datasets: KS localization, phase classification and Kendall's Tau with a low-complexity SSL module.

2. Related Work

STEPs is inspired by current video understanding research including KS localization, procedure learning, self-supervised learning and video summarization.

KS extraction & procedure learning of complex tasks. To obtain key steps, prior works use a state transition model [25], Mallows model [77], clustering and ordering of visual features [50] or weak alignment between visual and linguistic cues [83]. These approaches either rely on per-task training or additional language cues. A subset selection module is used as a teacher in [24] to obtain an unsupervised localization of KSs in a multi-task setting assuming access to task labels. Approaches that aim to learn features from complex task videos use pretext tasks to learn representations without supervision. They learn features using pairs of videos from a task by finding correspondences and imposing constraints like cycle-consistency [23], time-warping [37] and optimal-transport based alignment [56]. Recently, [7] proposed to use a loss function that builds over advances in video alignment and temporal consistency to learn strong models for procedure learning on both ego-centric and third person procedural datasets. Different from these prior works, we adapt off-the-shelf features and do not require pairs of videos in training. Multiple cues in training provide significant improvement over the state-of-the-art. We also propose an approach to improve the set of positives used to train models with temporal consistency. Our approach also works in a low-shot setting since it does not rely on alignment.

SSL for video understanding. Contrastive learning (CL) [34, 35, 66] shows impressive improvements on image-based self-supervision [13, 39, 52, 73, 80, 86]. Videos allow for additional constraints for self-supervision like discriminating temporally transformed version of the video [46], predicting speediness [9, 26, 95] or use of multiple camera views [16, 81]. Some works use alternate pretexts [47] while others

employ temporal coherence and ordering as signals [11, 42, 51, 63, 90, 93]. Most video-based SSL approaches focus on learning from short, trimmed videos depicting a single action. Real world task videos are rarely short or trimmed. To address this, [100] uses order verification to isolate actions from background, using global context and segment-based regularization [48], exploiting relations between clips [57] or devising an action boundary sensitive pretext task [94]. Unlike typical SSL models trained on a large collection of videos, we work with training on a few videos. Instead of training a feature extractor, we use off-the-shelf features and train a low complexity temporal model.

Multiple modalities for SSL: Multi-modal (e.g. audio and text) methods use cross-modal losses and augmentations [62, 64, 67, 68, 72, 89, 101] to learn features. Use of additional modalities/cues derived directly from the visual stream have been very successful for supervised learning [15, 21, 79]. Optical flow has been used by various works [36, 53, 91, 92] to learn and distill information from the motion stream while [71] explores cooperative CL for multi-modal SSL but for training a feature encoder on short trimmed videos. SSL relies on data augmentations which are not trivial when working with pre-extracted features. Working with multiple cues helps solve this problem by using contrastive learning similar to the ones used in approaches discussed above but without any explicit data augmentations. MM-SADA [65] proposed a modality alignment loss assuming access to trimmed action segments and applied it for domain adaptation. Our loss function is inspired by CoCLR [36] but differs in that our loss objectives employ temporal sequence of features instead of one feature per video. Further, we work with a CIDM-based objective and propose improvements to it. Different from these prior works, we do not aim to learn a feature extractor but a temporal model. We also note that natural language modality has been extensively used in procedure learning and related tasks [3, 6, 20, 22, 29, 55, 59, 69, 77, 83, 97]. Instead, our work explores the use of other modalities: on-device sensors like gaze and depth which are often available on AR devices or vision-derived modalities like motion or human pose. Language is only weakly aligned with text for procedural videos and might not be always available.

Unsupervised video summarization Video summarization without labels is another relevant thread for our work. Past works have used Reinforcement learning [96, 99], subset selection [82], clustering [18, 44, 84, 87] and GANs [4, 5, 41, 58, 98] or SSL [30] to extract summaries from videos. In contrast, KSs are task based and require understanding the video at an implicit semantic level. Our target use case does not have videos with abrupt changes, shot changes or extreme camera motion all of which can be cues for determining key steps. Instead we deal with videos which often have subtle changes as a task is performed. Our approach can be used for other high-level vision tasks such as phase

classification. That said, our learned features can be used with existing unsupervised video summarization approaches and alternative clustering based approaches.

3. Approach

A straightforward way of extracting key-steps would have been to use supervised learning to classify each frame of a video to KS or not a KS. Unfortunately, there do not exist any large scale procedural datasets with annotated key steps. One could use self-supervised learning to extract rich features from these datasets followed by finetuning on a small carefully labeled set. But since these videos are recorded by experts, even having a large unlabeled dataset is impractical. This is a problem since self-supervised learning has been shown to be data hungry. Thus we propose to use a two-stage solution of self-supervised representation learning followed by a cluster-and-sample pipeline for key step extraction. This entire pipeline does not use *any* labels to determine key steps. Next, we describe our approach assuming M modalities ($m_i, i \in [1, \cdots M]$).

3.1. Extracting raw features

The first step involves extracting raw features for use in the training pipeline. Our feature extractor could be any off-the-shelf backbone network like ResNet50 [40], RAFT encoder [85], human pose extractor or even on-device modalities like head pose, hand articulations, gaze and object pose [61]. The idea of adapting pre-extracted off-the-shelf features enables us to effectively leverage SSL on extremely small datasets.

During training, we divide the entire video into N chunks and randomly sample a frame from each chunk. Note that since our approach works with features instead of videos/images, we cannot employ any of the input-space augmentations. We apply temporal augmentations on features but do not generate multiple augmentations of the raw video.

The resultant features are denoted as $p_t^{m_i} \in \mathbb{R}^{D_i}$. We call these raw per-frame features (Fig. 2).

3.2. Encoding temporal context

The objective of our model is to take the raw per-frame features and adapt them so that these are useful for various downstream tasks. We use a temporal encoder (vanilla transformer) to capture the long-range, temporal dynamics of an instructional video. Specifically, the raw sequences $p_t^{m_i}$ are input to a per-modality temporal encoder to generate a sequence of adapted per-frame features $\tilde{q}_t^{m_i}$. Finally, a per-modality MLP extracts a projection of the adapted sequence and L2 normalizes it to obtain $q_t^{m_i} \in \mathbb{R}^D$.

3.3. Bootstrapped Multi-Cue Contrastive Loss

Ideally, we want to learn representations such that features corresponding to all frames of a step (like attaching the
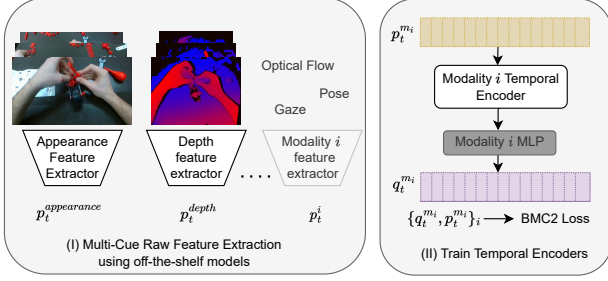
Figure 2. STEPs pipeline. (I) We first extract framewise multi-cue features $(p_t^{m_i})$ for an input video using off-the-shelf multi-cue feature extractors. (II) these features are then adapted to the task using a temporal feature encoder which returns framewise adapted features (q_t). We propose to use our BMC2 loss which enforces continuity in time and cross-cue contrastiveness to learn rich features. These features are then used with a variety of tasks including cluster + sampling for key-step extraction, key-step localization or phase classification.
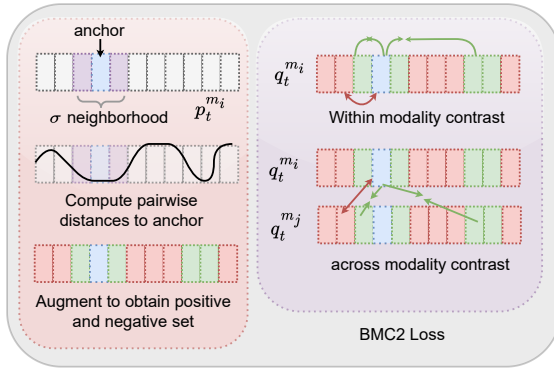

Figure 3. BMC2 loss. (Left) We start with a σ-neighborhood of the anchor and compute pairwise distances to other frame features of the video. We compute the mean distance within the neighborhood and use this as a threshold to find potential false negatives outside the window to obtain a new augmented positive set. (Right) We apply a multicue contrastive loss which enforces that positives are close in latent space across modalities.

leg of a table) lie close in the latent space. Since we do not have access to labels, we instead rely on a temporal consistency-based pretext task: features for frames close in time (positives) must be close in latent space while being far away from features of temporally far steps. Our contrastive learning objective is based on CIDM loss [37]. While this has been explored in the past [7, 37, 56], in this work we propose two improvements. We first define the vanilla loss and then describe our novel contributions to this loss.

The CIDM loss is used to enforce temporal consistency in the features. For an anchor a, the loss is defined as follows:

$$\mathcal{L}_{CL}(q_{t_a}, q_{t_j}, \mathcal{W}) = \mathcal{W}(t_a, t_j)\frac{d(q_{t_a}, q_{t_j})}{\gamma(t_a, t_j)}$$
$$+ (1 - \mathcal{W}(t_a, t_j))\,\gamma(t_a, t_j)\max\left(0, \zeta - d(q_{t_a}, q_{t_j})\right) \quad (1)$$

Here, $\gamma(t_a, t_j) = (t_a - t_j)^2 + 1$, $d(q_{t_a}, q_{t_j})$ is the Euclidean distance between q_{t_a}, q_{t_j}, ζ is the margin parameter, and $\mathcal{W}$ is the window such that, $\mathcal{W}(t_i, t_j) = 1$ if $|t_i - t_j| \leq \sigma$, and 0 otherwise. Here, σ is the window size used to define the set of positives.

While this loss has been successfully applied in prior works, it has certain issues. Consider the task of following a recipe. Chopping certain vegetables might repeat across the video. Using a fixed temporal window, could use two time-separated instances of the same step as negatives instead of positives thus giving an incorrect training signal. We propose an approach to fix this issue and expand the temporal window.

Temporal window generating function: The key idea is to find potential false negatives for a given anchor a by boot-strapping on the raw features. Intuitively, the raw features in different modalities should have enough signal to discard some false negatives and use these as positives instead thus improving the training signal. Given an anchor a, we have its raw features as $p_{t_a}^{m_i}$ and those of the other time steps as $p_{t_j}^{m_i} j \neq a$. We first calculate the embedding distance of each frame from the anchor. Next, we use the mean distance within the σ window as a threshold δ and recover time instances which have feature distances within that threshold. Thus, we have the following new window function:

$$\mathcal{W}'(t_a, t_j) = \begin{cases} 1, & \text{if } d(p_{t_a}^{m_i}, p_{t_j}^{m_i}) \leq \delta \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

We augment the original window to obtain $\tilde{\mathcal{W}} = \min(\mathcal{W}' + \mathcal{W}, 1)$.

Multi-cue consistency loss: In addition to enforcing temporal consistency between features from the same modality, we also enforce that features from different modalities within the window $\tilde{\mathcal{W}}$ are close in a common latent space. Our formulation easily allows us to do that by using q_{t_a}, q_{t_j} Eq. (1) from different modalities.

Overall BMC2 loss: We train our model with the weighted loss function :

$$\mathcal{L}_{BMC2} = \sum_{u,v=1}^{M} \lambda_{uv}\mathcal{L}_{CL}(q_{t_a}^{m_u}, q_{t_j}^{m_v}, \tilde{\mathcal{W}})),$$

where $\lambda_{uu} = 1$ for all u. Fig. 3 provides an illustration of the proposed loss function.

Note that, we *do not* back-propagate through the backbone feature extractors, $f^\star$, when training this pipeline. As the number of parameters in the temporal encoder and MLP modules is substantially smaller than the number of parameters in the backbone network (1.8M for our model vs 23M
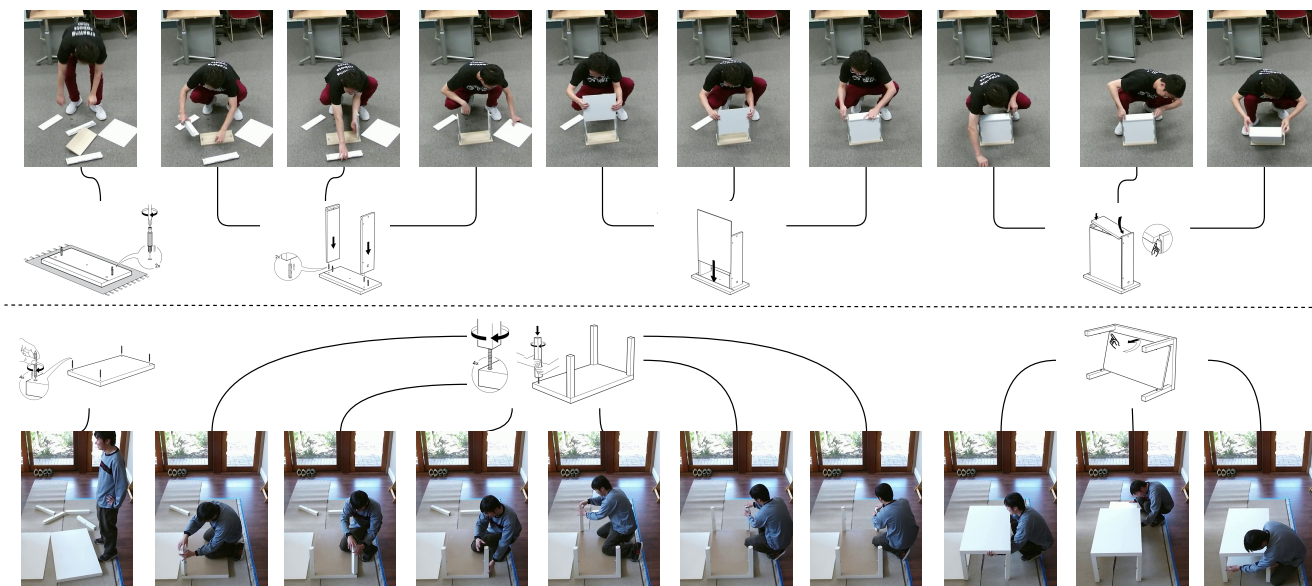
Figure 4. Key steps generated using STEPs. We show results on two videos from the assembly of 'Kallax Shelf Drawer' (top) and 'Lack Coffee Table' (bottom) respectively. We see that they generated key steps are plausible given the complex task. We also manually match the generated steps to the drawings in IKEA user manual and we see that that the steps correlate well with the manual. Note that some of our generated steps get matched to a single step in the manual since steps in the manual are often over simplifications.

for ResNet-50), precomputing the features substantially reduces the size of the gradients during training, allowing us to process very long feature sequences on modest hardware. In turn, this enables us to extract rich temporal information from the videos while simultaneously making the training process efficient. In particular, the features p_t are considered fixed throughout the training process and need only be computed once. They are completely determined by the choice of backbone network; we examine different choices of backbone networks in the supplementary material.

3.4. Clustering and Sampling

We follow a cluster-and-sample paradigm to sample KS once we have a trained model. Our approach borrows the clustering and ordering approach used in [50] but adapts their segmentation approach to sample *key steps* given a *single* video. Since the learned representations are quite effective, even simple K-Means clustering algorithm is sufficient to obtain partitions of the aggregated feature data for a video, each partition potentially corresponding to a sub-task or a key step. Our approach allows for a choice of a clustering algorithm based on user constraints, including algorithms that do not require a fixed number of clusters to be specified in advance [75]. To sample key steps, we first use a background frame rejection model [50] to reject steps which might correspond to background frames. Subsequently, each frame in the video is first assigned to one of the clusters. A step from each cluster is then chosen based on its distance to the cluster center and the key steps are temporally ordered.

Algorithm 1 Extracting Key steps

Input: Per-frame adapted features for the candidate video $\{\tilde{q}_t\}, t \in [1, \cdots, T]$, num clusters K
Output: Key-steps $KS(v) = \{a_k\}_{k=1}^{k=K}$,
  # Cluster the video
$\mathcal{C} = \texttt{cluster}(\{\tilde{q}_t\}, K)$ where $\mathcal{C} = \{C_i\}$
  # Obtain assignment and distance to closest cluster
$\{y_t, d_t\} = \texttt{predict\_cluster}(\{\tilde{q}_t\}, \mathcal{C})$
  # Sampling key-steps
for cluster $k = 1, \ldots, K$ **do**
    # Obtain time indices for frames assigned to $\mathcal{C}_k$
    $\mathcal{T}_k = \{t \,|y_t = k \;\forall t \in [1, \cdots, T]\}$
    # Remove elements farthest from the cluster center
    $\mathcal{T}'_k = \texttt{background\_reject}(\mathcal{T}_k, \alpha)$
    # Break into segments if adjacent indices are γ away
    $\mathcal{T}''_k = \texttt{split\_to\_segments}(\mathcal{T}'_k, \gamma)$
    # Sample key steps from each segment
    $a_k = \{t \sim \mathcal{T}''_{kj}\}$
end for
return $\{a_k\}_{k=1}^{k=K}$

While sampling from a cluster, we perform an additional partitioning step based on the timestamp to ensure that we sample key steps of the same potential sub-task but happening at different time instances. Algorithm 1 details our approach to cluster and sample key steps.

4. Experiments and Analyses

In this section, we evaluate the efficacy of our representation learning pipeline. KS extraction is a subjective task and

there do not exist any large scale datasets with labeled KS. Thus for quantatitive benchmarking, we compare our learned representations for the related task of Key Step localization (KSL). Furthermore, to demonstrate the usefulness of our SSL approach generally, we offer a quantitative comparison on other benchmarks used in the self-supervised video procedure learning for task based videos. Our qualitative visualizations of the succinct KS prove the efficacy of our approach. Finally, we provide analyses to understand the working of our approach.

Datasets: We evaluate our approach on four egocentric procedure learning datasets, Meccano [70] (toy bike assembly), EPIC-Tents [45] (camping tent assembly), CMU Kitchens [17] and EGTEA-Gaze+ [54] (cooking recipes). Unlike prior work which use just the visual stream from these datasets, we also explore the use of modalities like depth and gaze available with these egocentric datasets during the training stage. Because they are captured from an egocentric viewpoint and have other synchronized sensor recordings, these datasets are very relevant to our motivating use case of AR-based instruction. In addition to these, we also evaluate on three third-person procedural datasets, IkeaAssembly [8] (furniture assembly), CrossTask [100] and ProceL [25] which contain diverse procedures. We also evaluate on PennActions which while not a procedural dataset, is a common benchmark used in unsupervised video parsing works.

Implementation Details: Our temporal encoder consists of a two layer vanilla transformer model. We use two modalities $M = 2$ for all experiments in the main paper. Additional sensor modalities (like Gaze, Depth, etc) exist for some of these datasets but are not consistent across datasets. To tease out the effect of nature of sensor modalities from the learning approach, all comparisons with prior works use features extracted using ResNet-50 (m_1) and RAFT-OpticalFlow [85] (m_2) unless otherwise mentioned. We choose these two modalities since they can readily be extracted from any procedural task videos and depend on the appearance stream alone. During inference, we use ResNet-50 features alone (m_1) for a fair comparison with competing approaches.

4.1. Key Step Localization

Key Step Localization aims to localize each key step in a video. We follow the same evaluation protocol as prior work [7, 25, 50, 83, 88]. The learned representations are first clustered and all frames are assigned to the various clusters. Since we work in an unsupervised setting, these frames are then matched to ground truth segments using the Hungarian algorithm [49]. These mappings are then used to calculate the F-1 score and IoU scores used for comparison with prior works. F-1 score is the harmonic mean of precision and recall. Recall is computed as the ratio of number of frames having the correct key step prediction to the ground truth

number of key frames across all key steps. Precision is the ratio of the number of correctly predicted frames and number of frames predicted as key steps. Following [7], we report key-step averaged metric for the egocentric datasets and overall average for CrossTask and ProceL. We used the publicly available key step annotations provided by [7] for the egocentric datasets and use their evaluation code for a fair comparison.

We present our main results in Table 1 and Table 2 for Egocentric and third person procedural datasets respectively. We note that we consistently outperform the prior works with often large margins. As demonstrated in the Sec. 4.4, we can attribute the performance improvements to three key factors: ability to use long range dependencies due to use of pre-extracted features, use of multi-cue information during training, and refinement of the positive window for the contrastive learning. None of the prior works in procedure learning use long term temporal information in their features. We argue that these are essential to learning good features and is practically always available. Use of pre-extracted features makes this computationally tractable in our case. We would like to reiterate that for a fair comparison with prior works, we do *not* use m_2 during evaluation and use the same feature extractor (ResNet-50) as Bansal et al. [7].

4.2. Phase classification and Kendall's Tau

While Key-Step localization is the predominant way of evaluating approaches for procedure learning, our learned representations can be used for other tasks as well. Following prior work on video alignment [37, 56], we evaluate our learned representations for Phase Classification (PC). Phase classification calculates the per frame classification accuracy for fine-grained action recognition by training an SVM on the extracted per frame features of the training set. We show results on the IkeaAssembly dataset which is a dataset of people assembling Ikea furniture. The dataset depicts complex furniture assembly tasks performed by multiple people across views. The tasks are composed of multiple sub-tasks like flipping the table, attaching the leg etc. PennActions has videos of humans doing sports and exercises and are composed of multiple phases per action. We follow the same evaluation protocol as used in [23, 38, 56]. In addition, we also evaluate the Kendall's Tau (K.T) on the PennActions dataset[1] Table 3 shows results of our approach on these metrics. The improved performance shows that our approach learns generalizable features for complex videos. We see that our STEPs model which uses the same backbone as previous approaches (albeit without *any* finetuning) outperforms the recent approach of VAVA on most metrics for both datasets.

---

[1] Since K.T assumes a strict montonic order of actions, we report it only for the PennActions dataset.

Table 1. Procedure learning results on egocentric datasets. We see significant improvements using our approach on both F1 and IoU scores. We use the same backbone model as TC3I [7] for a fair comparison. Note that unlike [7], we do not finetune our backbones, or rely on video alignment. Our approach shows improvements using simple k-Means algorithm for clustering. All reported numbers for STEPs are averaged over three runs.

Approach	Clustering Algorithm	Video Alignment	Finetune Backbone	CMU-MMAC F1	CMU-MMAC IoU	EGTEA G. F1	EGTEA G. IoU	Meccano F1	Meccano IoU	EPIC-Tents F1	EPIC-Tents IoU
Random	-	-	-	15.7	5.9	15.3	4.6	13.4	5.3	14.1	6.5
Uniform	-	-	-	18.4	6.1	20.1	6.6	16.2	6.7	16.2	7.9
Bansal et al. [7]	kMeans	✓	✓	19.2	9.0	20.8	7.9	16.6	8.0	15.4	7.8
Bansal et al. [7]	PCM	✓	✓	22.7	**11.1**	21.7	9.5	18.1	7.8	17.2	8.3
STEPs	kMeans	✗	✗	**28.3**	**11.4**	**30.8**	**12.4**	**36.4**	**18.0**	**42.2**	**21.4**

Table 2. Procedure learning results on third person datasets. We compare with various state-of-the-art approaches and obtain consistent improvements. Unlike some prior works, we rely on appearance features alone at inference. We do not finetune our backbones, or rely on video alignment.

Approach	ProceL P	ProceL R	ProceL F1	CrossTask P	CrossTask R	CrossTask F1
Uniform	12.4	9.4	10.3	8.7	9.8	9.0
Alyarc et al. [3]	12.3	3.7	5.5	6.8	3.4	4.5
Kukleva et al. [50]	11.7	30.2	16.4	9.8	35.9	15.3
Elhamifar et al. [24]	9.5	**26.7**	14.0	10.1	**41.6**	16.3
Fried et al. [29]	-	-	-	-	28.8	-
Shen et al. [83]	16.5	31.8	21.1	15.2	35.5	21.0
Bansal et al. [7]	20.7	22.6	21.6	22.8	22.5	22.6
STEPs	**23.5**	**26.7**	**24.9**	**26.2**	25.8	**25.9**

4.3. KS Extraction

Our key goal in this paper is to be able to sample a few key steps corresponding to the complex task for use in AR content creation. The sections above demonstrate the effectiveness of our learned features for related tasks of Key-Step Localization and Phase Classification. Since we do not have any labeled datasets to evaluate KS extraction, we show qualitative results instead. In Figure 4, we visualize the key steps extracted on the Ikea Assembly [8] dataset. We visualize the extracted key steps for two randomly sampled videos for the tasks of assembling 'Kallax Shelf Drawer' and 'Lack Coffee Table' respectively. We also manually map them to diagrams from the official IKEA manuals [43]. We notice that the extracted key steps are semantically meaningful given the mapping. We did not include all steps from the manual since the curators of the dataset pre-assembled a few parts for ease of dataset collection. Note that what constitutes a key step can be very subjective and the number of steps needed to accomplish a task might depend on the application area. For example, a trainee trying to learn a new task might benefit from more steps than an an experienced person trying to transfer skills from a known task. For this visualization, we simply set the number of clusters $K = 10$.

We can also make use of clustering algorithms like [75] which automatically determine the number of clusters or even optimization-based algorithms to determine key steps given our learned representations. We include additional visualizations in the supplementary material. Our two-stage approach allows for this customization without having to retrain the model.

4.4. Ablations and Analyses

In this section, we present ablations and analyses of our approach to better understand its working. We work with the Meccano bike assembly dataset for the following experiments unless otherwise stated. We include ablations for other datasets in the supplementary material.

Use of On-Device sensor modalities: Since we work with pre-extracted features, our approach easily allows use of on-device sensor modalities commonly available in many AR/VR headsets. These sensors often give complementary and distilled information about the scene which can be useful for the task at hand. For example, consider the task of assembling a toy bike with many parts. The visual stream sees a lot of clutter in front of the table. The gaze sensor returns very low dimensional information about which part of the bike the person is looking at and might be helpful. The Meccano dataset was captured using an Intel RealSense SR300 headset and a Pupil Core Gaze tracker mounted on the participant. The dataset includes synchronized RGB, Depth and Gaze signals. The results using these sensors are presented in Table 4 (Use of on-device sensor modalities). We see that the use of on-device sensors improves the performance over model trained without any multi-cue training. Further, we see that Depth and gaze information is competitive to optical flow features. This is beneficial since for this dataset capture, these modalities comes for *free*. Our approach enables plug-and-play use of additional modalities.

Use of MM information at inference: For a fair comparison, following prior works we use only appearance cue during inference. In the following experiment, we make use of multi-modal cues during inference as well. In

Table 3. Proxy task results on IkeaASM and PennActions. Our feature encoder is able to adapt the off-the-shelf features for tasks like phase classification and Kendall's Tau (K.T). We show significant improvements over the state-of-the-art using the same backbone. Note that unlike prior approaches, we do not finetune our backbones. Use of motion information at inference leads to further improvements.

Approach	Video Alignment	Finetune Backbone	Ikea w/o Bgnd Phase Cls			Ikea with Bgnd Phase Cls			PennActions Phase Cls			K.T
			0.1	0.5	1.0	0.1	0.5	1.0	0.1	0.5	1.0	
Supervised	✗	✓	21.76	30.26	33.81	20.74	25.61	31.92	67.10	82.78	86.05	-
Random	-	-	17.89	17.89	17.89	17.03	17.41	17.61	44.18	46.19	46.81	-
ImageNet	-	✗	18.05	19.27	19.50	17.27	18.02	18.64	44.96	50.91	52.86	-
SAL [63]	✗	✓	21.68	21.72	22.14	22.94	23.43	25.46	74.87	78.26	79.96	0.63
TCN [78]	✗	✓	25.17	25.70	26.80	22.51	25.47	25.88	81.99	83.67	84.04	0.73
TCC [23]	✓	✓	24.74	25.22	26.46	22.70	25.04	25.63	81.26	83.35	84.45	0.74
LAV [37]	✓	✓	29.78	29.85	30.43	23.19	25.47	25.54	83.56	83.95	84.25	0.80
VAVA [56]	✓	✓	**31.66**	33.79	32.91	**29.12**	29.95	29.10	**83.89**	84.23	84.48	0.81
STEPs	✗	✗	28.59	**36.25**	**37.02**	26.5	**31.92**	31.52	83.34	**85.06**	**85.5**	**0.91**

Table 4. Benefits of using mutliple modalities. Our approach can easily be used with sensors often available on AR headsets. Use of multiple modalities at inference further improves performance.

Method	Training	Inference	F1	IoU
Random	-	-	13.4	5.3
Uniform	-	-	16.2	6.7
TC3I	RGB	RGB	18.1	7.8
Ours without Multi-Cue training				
STEPs	RGB	RGB	32.1	14.9
Use of RGB-derived modality				
STEPs	RGB+OF	RGB	**36.4**	**18.0**
Use of on-device sensor modalities				
STEPs	RGB+Gaze	RGB	35.4	17.0
STEPs	RGB+Depth	RGB	**36.6**	**18.1**
Use of different modalities at inference				
STEPs	RGB+OF	RGB	36.4	**18.0**
STEPs	RGB+OF	OF	30.6	15.6
STEPs	RGB+OF	RGB+OF	**37.6**	**18.1**

Table 4 (Use of different modalities at inference), we show results with Appearance+Motion cues at inference. We simply concatenate the features from both modalities and use them for the downstream task. We see that even simple late fusion leads to improvements showing complementary information in the two streams. Note that while these are referred to as separate cues, RAFT features (optical flow) are derived from the RGB stream.

How much training data do we need? Our work follows a learning-by-observation style framework. Our goal is to be able to learn from a few recordings of the complex task by experts and hence a practical system should be able to work with little training data. Collecting unlabeled data for general tasks might be easy through scraping YouTube, but extracting a lot of procedural data through AR systems is non-trivial. In Figure 5 we plot the F1 score as a function of training data on the Meccano dataset. We see that our approach gives good performance even with a single video. We hypothesize that this adaptability of the model to little
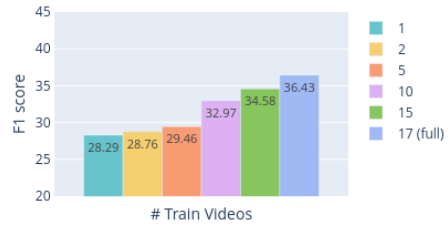

Figure 5. Amount of pretraining data. In this experiment we vary the amount of pre-training data that is used to train our encoder. Relying on off-the-shelf features lets us train our model to a good performance with even with a single video.

Table 5. Effect of different losses. In this experiment, we evaluate the effect of our different loss terms on the Meccano dataset. We see that our loss terms leads to performance improvements.

Approach	Multi-Cue training	BMC2	F-1	IoU
TC3I	-	-	18.1	7.8
STEPs	✗	✗	32.1	14.9
STEPs	✗	✓	33.1	15.8
STEPs	✓	✗	34.2	16.6
STEPs	✓	✓	**36.4**	**18.0**

training data is due to the use of pre-extracted features with a light-weight temporal module and use of long temporal sequences which gives the model lots of good signals during training.

Ablation analyses on various losses: We quantitatively evaluate the effect of each our loss terms. In Table 5 we show results on the Meccano dataset. We see that each of our proposed losses brings an improvement to the final performance.

Importance of long range temporal dependencies: Note that most previous approaches do not focus on the temporal aspect of the problem. We believe that temporal information is especially important for complex tasks since they often

Table 6. Effect of varying size of temporal context used for training. We see that the model achieves best performance at 1024 chunks, much higher than used by past approaches (< 100).

Window	128	256	512	1024	2048
F1	33.04	36.08	34.91	**36.43**	34.88
IoU	15.64	17.75	17.01	**18.01**	16.38

Table 7. Changing num clusters. We report the F1 scores for models trained on Meccano and EPIC-Tents dataset. We consistently outperform the state-of-the-art.

Approach	K=7	K=10	K=12	K=15
	Meccano			
Random	13.4	10.1	8.8	7.4
Uniform	16.2	13.2	11.8	11.2
Bansal et al. [7]	18.1	15.2	13.5	11.9
STEPs	**36.4**	**31.2**	**27.5**	**25.6**
	EPIC-Tents			
Random	14.1	10.6	9.1	8.3
Uniform	16.2	15.1	14.2	13.0
Bansal et al. [7]	17.2	11.1	12.1	9.5
STEPs	**36.7**	**32.8**	**29.4**	**29.7**

involve subtle and slow changes. Such changes may not be easily captured using a small temporal context used by some previous approaches. Prior works [37, 56] use very few sampled frames (~ 40) per video during training with little to no temporal context . This is sub-optimal since the videos from procedural datasets often contain 10000+ frames with sub-actions having long temporal spans (Figure 3 [8]). Fine-tuning a backbone while dealing with long temporal contexts is computationally expensive. Use of pre-extracted features enables us to use large temporal context which helps in capturing more information from the video. In Table 6, we empirically verify the effect of long temporal contexts.

Effect of number of clusters In Table 7, we evaluate our trained models for the Meccano and EPIC-Tents dataset by varying the number of clusters used during kMeans. We report the F1 score for this experiment. As noted by prior work, we obtain the best results at $K = 7$. Our approach consistently outperforms the baselines.

Alternate temporal encoders In the main paper, we use a transformer model to capture long range dependencies. In this section, we use alternate temporal encoders like LSTMs and TCNs. In Table 8, we compare these with a transformer based module. As noted by works in the past, transformers learn better features owing to the use of self-attention and thus are better suited for our problem.

Additional analyses and experiments: Please refer to the

Table 8. Comparison between different temporal modeling types on Meccano dataset. We see that using a transformer gives us the best performance for both metrics.

Temporal module	F1	IoU
TCN	32.5	14.8
LSTM	33.6	16.4
Transformer	**36.4**	**18.0**

supplementary material section for data and experiments including 1) comparison of features extracted from different backbones; 2) additional experiments and visualizations on key step extraction; 3) additional ablations and analyses; 4) dataset and implementation details; 5) Use of > 2 modalities during training.

5. Discussion and Conclusion

We believe in the potential to use AR headsets to transform on-the-job training and guidance. Key step extraction from egocentric task videos will be critical to enabling AR-based task guidance at scale. In this work, we investigate this problem and offer substantial evidence that accounting for long-range temporal dependencies, use of multi-cue information during training and runtime, and refinement of the positive window for the contrastive learning are crucial to key step extraction. These findings are consistent with psychological research that advocates for detailed, multi-modal observation of human experts when designing training programs for novices [28,32]. In future work, we hope to address several important related problems. Primary among these is collecting a detailed dataset of procedural task recordings from AR headsets. Also, incorporating recent advances in meta-learning and few-shot learning as ways to potentially improve key step extraction while reducing the amount of required training data is an important line of research. In short, the present work is only the beginning.

6. Acknowledgements

References

[1] Magid Abraham and Marco Annunziata. Augmented reality is already improving worker performance. *Harvard Business*

Review, 13:1–5, 2017. 1

[2] S Agrawal, Aaron De Smet, Pawel Poplawski, and Angelika Reich. Beyond hiring: How companies are reskilling to address talent gaps. *McKinsey Global Institute*, 2020. 1

[3] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 3, 7

[4] Evlampios Apostolidis, Eleni Adamantidou, et al. Ac-sumgan: Connecting actor-critic and generative adversarial networks for unsupervised video summarization. *IEEE T-CSVT*, 31(8):3278–3292, 2020. 3

[5] Evlampios Apostolidis, Eleni Adamantidou, et al. Unsupervised video summarization via attention-driven adversarial learning. In *Intl. Conf. on Multimedia Modeling*, pages 492–504. Springer, 2020. 3

[6] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. *arXiv preprint arXiv:2307.08763*, 2023. 3

[7] Siddhant Bansal, Chetan Arora, and CV Jawahar. My view is the best view: Procedure learning from egocentric videos. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 657–675. Springer, 2022. 2, 4, 6, 7, 9

[8] Yizhak Ben-Shabat, Xin Yu, et al. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *IEEE/CVF WACV*, pages 847–859, 2021. 2, 6, 7, 9

[9] Sagie Benaim, Ariel Ephrat, et al. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF CVPR*, pages 9922–9931, 2020. 2

[10] William L. Bryan and Noble Harter. Studies on the telegraphic language: The acquisition of a hierarchy of habits. *Psychological review*, 6(4):345, 1899. 1

[11] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*, pages 770–786, 2018. 3

[12] Neil Charness and Michael Tuffiash. The role of expertise research and human factors in capturing, explaining, and producing superior performance. *Human factors*, 50(3):427–432, 2008. 1

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[14] Jeffrey Clark and Philip G Crandall. Educational affordances of google glass as a new instructional platform for foodservice training. *Management & Education*, 13(1):28–32, 2019. 1

[15] Huseyin Coskun, M Zeeshan Zia, et al. Domain-specific priors and meta learning for few-shot first-person action recognition. *TPAMI*, 2021. 3

[16] Srijan Das and Michael S Ryoo. Viewclr: Learning self-supervised video representation for unseen viewpoints. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5573–5583, 2023. 2

[17] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. 2009. 6

[18] Yash Dhamecha, Swanand Gadekara, Sai Deshmukh, and Yashodhara Haribhakta. Video summarization using feature vector clustering. In *2nd Intl. Conf. on IoT, Social, Mobile, Analytics & Cloud in Computational Vision & Bio-Engineering (ISMAC-CVB 2020)*, 2020. 3

[19] Federica Bogo Dorin Ungureanu et al. HoloLens 2 Research Mode as a Tool for Computer Vision Research. *arXiv:2008.11239*, 2020. 1

[20] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 868–878, 2020. 3

[21] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022. 3

[22] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18952–18961, 2023. 3

[23] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *IEEE/CVF CVPR*, pages 1801–1810, 2019. 2, 6, 8

[24] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. In *ECCV*, pages 557–573. Springer, 2020. 2, 7

[25] Ehsan Elhamifar and Zwe Naing. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE/CVF ICCV*, pages 6341–6350, 2019. 2, 6

[26] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *IEEE/CVF CVPR*, pages 919–929, 2020. 2

[27] K. Anders Ericsson and Andreas C Lehmann. Expert and exceptional performance: Evidence of maximal adaptation to task constraints. *Annual review of psychology*, 47(1):273–305, 1996. 1

[28] Peter J Fadde and Leonard Zaichkowsky. Training perceptual-cognitive skills in sports using technology. *Journal of Sport Psychology in Action*, 9(4):239–248, 2018. 1, 9

[29] Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to segment actions from observation and narration. *arXiv preprint arXiv:2005.03684*, 2020. 3, 7

[30] Junyu Gao, Xiaoshan Yang, Yingying Zhang, and Changsheng Xu. Unsupervised video summarization via relation-aware assignment learning. *IEEE T-Multimedia*, 23:3203–3214, 2020. 3

[31] Craig Giffi, Paul Wellener, Ben Dollar, H Ashton Manolian, Luke Monck, and Chad Moutray. Deloitte and the manufacturing institute skills gap and future of work study. *Deloitte Insights*, 2018. 1

[32] Daniel Gopher, Maya Well, and Tal Bareket. Transfer of skill from a computer game trainer to flight. *Human factors*, 36(3):387–405, 1994. 1, 9

[33] C Guldimann and N Powell. Green collar jobs: The skills revolution canada needs to reach net zero. *RBC Special Reports*, 2022. 1

[34] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 2

[35] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEE/CVF CVPR*, 2006. 2

[36] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *Advances in NIPS*, 33:5679–5690, 2020. 3

[37] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *IEEE/CVF CVPR*, pages 5548–5558, 2021. 2, 4, 6, 8, 9

[38] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5548–5558, 2021. 6

[39] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF*, 2020. 2

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[41] Xufeng He, Yang Hua, et al. Unsupervised video summarization with attentive conditional generative adversarial networks. In *27th ACM Intl. Conf. on Multimedia*, pages 2296–2304, 2019. 3

[42] Kai Hu, Jie Shao, et al. Contrast and order representations for video self-supervised learning. In *IEEE/CVF ICCV*, pages 7939–7949, 2021. 3

[43] IKEA. Ikea global http://www.ikea.com/, 2022. 7

[44] Shruti Jadon and Mahmood Jasim. Unsupervised video summarization framework using keyframe extraction and video skimming. In *IEEE 5th Intl. Conf. on Computing Communication and Automation (ICCCA)*, pages 140–145. IEEE, 2020. 3

[45] Youngkyoon Jang, Brian Sullivan, Casimir Ludwig, Iain Gilchrist, Dima Damen, and Walterio Mayol-Cuevas. Epic-tent: An egocentric video dataset for camping tent assembly. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 6

[46] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *ECCV*, pages 425–442. Springer, 2020. 2

[47] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI conf. on AI*, volume 33, pages 8545–8552, 2019. 2

[48] Haofei Kuang, Yi Zhu, et al. Video contrastive learning with global context. In *IEEE/CVF ICCV*, pages 3195–3204, 2021. 3

[49] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[50] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019. 2, 5, 6, 7

[51] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *IEEE ICCV*, pages 667–676, 2017. 3

[52] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2020. 2

[53] Rui Li, Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Motion-focused contrastive learning of video representations. In *IEEE/CVF ICCV*, pages 2105–2114, 2021. 3

[54] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018. 6

[55] Yan-Bo Lin, Hung-Yu Tseng, Hsin-Ying Lee, Yen-Yu Lin, and Ming-Hsuan Yang. Exploring cross-video and cross-modality signals for weakly-supervised audio-visual video parsing. *Advances in Neural Information Processing Systems*, 34:11449–11461, 2021. 3

[56] Weizhe Liu, Bugra Tekin, Huseyin Coskun, Vibhav Vineet, Pascal Fua, and Marc Pollefeys. Learning to align sequential actions in the wild. *IEEE/CVF CVPR*, 2022. 2, 4, 6, 8, 9

[57] Dezhao Luo, Yu Zhou, Bo Fang, Yucan Zhou, Dayan Wu, and Weiping Wang. Exploring relations in untrimmed videos for self-supervised learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(1s):1–21, 2022. 3

[58] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017. 3

[59] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558*, 2015. 3

[60] Ryszard S Michalski and Robert E Stepp. Learning from observation: Conceptual clustering. In *Machine learning*, pages 331–363. Springer, 1983. 1

[61] Microsoft. Hololens2 https://www.microsoft.com/en-us/hololens/, 2023. 3

[62] 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

[63] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 3, 8

[64] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12486, 2021. 3

[65] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 122–132, 2020. 3

[66] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[67] Mandela Patrick, Yuki M Asano, Polina Kuznetsova, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. On compositions of transformations in contrastive self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9577–9587, 2021. 3

[68] Shraman Pramanick, Li Jing, Sayan Nag, Jiachen Zhu, Hardik Shah, Yann LeCun, and Rama Chellappa. Volta: Vision-language transformer with weakly-supervised local-feature alignment. *arXiv preprint arXiv:2210.04135*, 2022. 3

[69] 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. *arXiv preprint arXiv:2307.05463*, 2023. 3

[70] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Maria Farinella. The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1569–1578, 2021. 6

[71] Nishant Rai, Ehsan Adeli, Kuan-Hui Lee, Adrien Gaidon, and Juan Carlos Niebles. Cocon: Cooperative-contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3384–3393, 2021. 3

[72] Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, et al. Broaden your views for self-supervised video learning. In *IEEE/CVF ICCV*, 2021. 3

[73] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *International Conference on Learning Representations*, 2021. 2

[74] Jeremy S Ruthberg, Galen Tingle, Lisa Tan, Lauren Ulrey, Sue Simonson-Shick, Rebecca Enterline, Henry East-man, Jeffrey Mlakar, Robert Gotschall, Erin Henninger, et al. Mixed reality as a time-efficient alternative to cadaveric dissection. *Medical Teacher*, 42(8):896–901, 2020. 1

[75] M. Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2019. 5, 7

[76] Bjorn Schwerdtfeger, Rupert Reif, Willibald A Gunthner, Gudrun Klinker, Daniel Hamacher, Lutz Schega, Irina Bockelmann, Fabian Doil, and Johannes Tumler. Pick-by-vision: A first stress test. In *2009 8th IEEE international symposium on mixed and augmented reality*, pages 115–124. IEEE, 2009. 1

[77] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *IEEE CVPR*, pages 8368–8376, 2018. 2, 3

[78] Pierre Sermanet, Corey Lynch, et al. Time-contrastive networks: Self-supervised learning from video. In *IEEE ICRA*, pages 1134–1141. IEEE, 2018. 8

[79] Anshul Shah, Shlok Mishra, Ankan Bansal, Jun-Cheng Chen, Rama Chellappa, and Abhinav Shrivastava. Pose and joint-aware action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3850–3860, 2022. 3

[80] Anshul Shah, Suvrit Sra, Rama Chellappa, and Anoop Cherian. Max-margin contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8220–8230, 2022. 2

[81] Ketul Shah, Anshul Shah, Chun Pong Lau, Celso M de Melo, and Rama Chellappa. Multi-view action recognition using contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3381–3391, 2023. 2

[82] Yair Shemer, Daniel Rotman, and Nahum Shimkin. Ilssumm: Iterated local search for unsupervised video summarization. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1259–1266. IEEE, 2021. 3

[83] Yuhan Shen, Lu Wang, and Ehsan Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *IEEE/CVF CVPR*, 2021. 2, 3, 6, 7

[84] Nitesh Shroff, Pavan Turaga, and Rama Chellappa. Video précis: Highlighting diverse aspects of videos. *IEEE Transactions on Multimedia*, 12(8):853–868, 2010. 3

[85] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 3, 6

[86] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020. 2

[87] Pavan Turaga, Ashok Veeraraghavan, and Rama Chellappa. Unsupervised view and rate invariant clustering of video sequences. *Computer Vision and Image Understanding*, 113(3):353–371, 2009. 3

[88] Rosaura G VidalMata, Walter J Scheirer, et al. Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences. In *IEEE/CVF WACV*, 2021. 6

[89] Duo Wang and Salah Karout. Fine-grained multi-modal self-supervised learning. *arXiv preprint arXiv:2112.12182*, 2021. 3

[90] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *IEEE CVPR*, pages 8052–8060, 2018. 3

[91] Fanyi Xiao, Joseph Tighe, and Davide Modolo. Modist: Motion distillation for self-supervised video representation learning. *arXiv preprint arXiv:2106.09703*, 2021. 3

[92] Bo Xiong, Haoqi Fan, Kristen Grauman, and Christoph Feichtenhofer. Multiview pseudo-labeling for semi-supervised learning from video. In *IEEE/CVF ICCV*, pages 7209–7219, 2021. 3

[93] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 3

[94] Mengmeng Xu, Juan-Manuel Pérez-Rúa, et al. Boundary-sensitive pre-training for temporal localization in videos. In *IEEE/CVF ICCV*, 2021. 3

[95] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *IEEE/CVF CVPR*, 2020. 2

[96] Ui-Nyoung Yoon, Myung-Duk Hong, and Geun-Sik Jo. Interp-sum: Unsupervised video summarization with piecewise linear interpolation. *Sensors*, 21(13):4562, 2021. 3

[97] Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 825–828, 2014. 3

[98] Li Yuan, Francis Eng Hock Tay, Ping Li, and Jiashi Feng. Unsupervised video summarization with cycle-consistent adversarial LSTM networks. *IEEE T-MM*, 22(10), 2019. 3

[99] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *AAAI C-AI*, 2018. 3

[100] Dimitri Zhukov, Jean-Baptiste Alayrac, Ivan Laptev, and Josef Sivic. Learning actionness via long-range temporal order verification. In *ECCV*, pages 470–487. Springer, 2020. 3, 6

[101] Mohammadreza Zolfaghari, Yi Zhu, et al. Crossclr: Cross-modal contrastive learning for multi-modal video representations. In *IEEE/CVF ICCV*, pages 1450–1459, 2021. 3