Efficient Pre-training for Localized Instruction Generation of Procedural Videos

Anil Batra[♠], Davide Moltisanti[♦], Laura Sevilla-Lara[♠], Marcus Rohrbach[★], and Frank Keller[♠]

[♠]University of Edinburgh [♦]University of Bath [★]TU Darmstadt & hessian.AI
a.k.batra@ed.ac.uk

Abstract. Procedural videos, exemplified by recipe demonstrations, are instrumental in conveying step-by-step instructions. However, understanding such videos is challenging as it involves the precise localization of steps and the generation of textual instructions. Manually annotating steps and writing instructions is costly, which limits the size of current datasets and hinders effective learning. Leveraging large but noisy video-transcript datasets for pre-training can boost performance but demands significant computational resources. Furthermore, transcripts contain irrelevant content and differ in style from human-written instructions. To mitigate these issues, we propose a novel technique, Sieve & Swap, to automatically generate high-quality training data for the recipe domain: (i) Sieve: filters irrelevant transcripts and (ii) Swap: acquires high-quality text by replacing transcripts with human-written instruction from a text-only recipe dataset. The resulting dataset is three orders of magnitude smaller than current web-scale datasets but enables efficient training of large-scale models. Alongside Sieve & Swap, we propose Procedure Transformer (ProcX), a model for end-to-end step localization and instruction generation for procedural videos. When pre-trained on our curated dataset, this model achieves state-of-the-art performance on YouCook2 and Tasty while using a fraction of the training data. We have released code and dataset.[1]

1 Introduction

Instructional videos are abundantly available online and constitute a valuable resource for humans to acquire procedural knowledge, aiding in tasks like recipe preparation. However, the task of understanding the sequence of actions in instructional videos is challenging for current AI models. It involves two distinct tasks: (i) the temporal localization of steps in the video and (ii) the generation of a textual instruction for each step. There are a number of existing datasets for procedural learning, including COIN [28] and CrossTask [40], which are annotated with fixed, template-based labels (e.g., *pour-water*). These labels, however, fail to describe the varied details of instructions in the videos (e.g., attributes and fine-grained objects). Datasets annotated with human-written instructions exist in the recipe domain, in particular YouCook2 [38] and Tasty [24]. However, due to the

[1] https://github.com/anilbatra2185/sns_procx

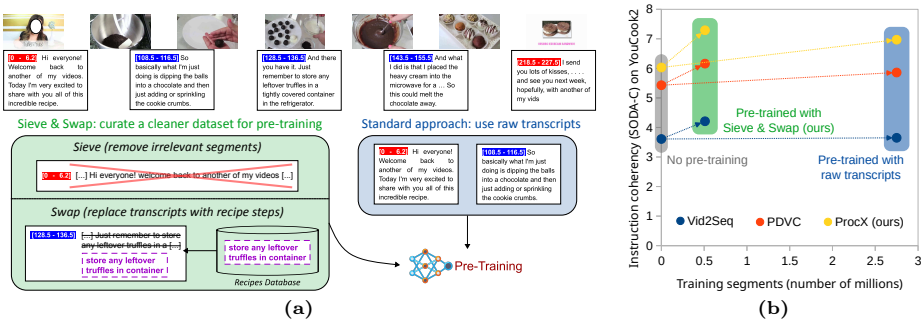

Fig. 1: (a) Sieve & Swap at a glance. We first remove irrelevant video ASR (Automatic Speech Recognition) segments, then substitute the raw transcripts with recipe steps retrieved from a recipe database. The resulting dataset is used for pre-training. Previous work uses raw and noisier transcripts, thus requiring larger amounts of data for effective pre-training. **(b)** With Sieve & Swap we generate a smaller but better pre-training dataset. Compared to using a larger number of raw segments (right), models achieve higher performance with a fraction of the data (middle). Note how the improvement from no pre-training (left) is steeper with Sieve & Swap. The plotted metric is the generated instruction coherency metric (SODA-C [9]), models were fine-tuned and tested on YouCook2 [38].

substantial annotation cost these datasets are limited in size, which poses a risk of over-fitting for large models. To achieve good performance with current neural architectures, large amounts of training data are required [7], and researchers have addressed this challenge by augmenting datasets or adopting strategies such as self-supervised training or pre-training with noisy web-scale datasets. Drawing inspiration from substantial advancements in caption generation achieved with datasets such as HowTo100M [20], Yang *et al*. [33] employ noisy video transcripts (automatically transcribed speech) as a form of weak supervision during pre-training to generate segment boundaries and instructions. Leveraging noisy data is beneficial, however this method trains on one million videos, requiring at least 12 TB of storage in low quality.

Another limitation of this strategy is sub-optimal multimodal learning due to irrelevant content and stylistic differences between transcripts and written language. For example, transcripts often include content like greetings and advertisements, which do not form part of the instructions that video segments are annotated with. Studies by Chafe *et al*. [6] and Einhorn *et al*. [8] highlight the distinct attributes of written and spoken language: written language is more concise and varied, while spoken language tends to be more protracted and repetitive. This creates a substantial domain gap. In this work, we propose Sieve & Swap, a technique that fuses an instructional video dataset and a recipe text dataset, resulting in a smaller multimodal dataset for effective procedure learning. Figure 1a summarizes our approach and contrasts it with prior methods. Specifically, we combine a subset of HowTo100M [20] of instructional cooking videos with the RecipeNLG [4] collection of text-only cooking recipes. We develop a novel method to retrieve relevant and high-quality reference descriptions for procedural learning, which involves extracting human-written instructional sentences from RecipeNLG. We employ sentence embeddings to filter irrelevant sentences from

video transcripts based on text similarity, and replace the relevant transcript sentences with instructional sentences from RecipeNLG. The resulting Sieve & Swap dataset is characterized by less noise, providing high-quality pre-training data for the task of human-style instruction generation. The Sieve & Swap dataset is smaller by three orders of magnitude (≈48K videos) than the pre-training datasets employed in previous studies (≈15M videos) [18, 33], but contains both segments with temporal boundaries and human-written instructions. Despite the smaller scale, models pre-trained on the Sieve & Swap dataset outperform the same models pre-trained on raw transcripts, as we show in Figure 1b.

Our second contribution is Procedure Transformer (*ProcX*), a model that combines pre-training and fine-tuning to localize steps in videos and generate textual instructions. ProcX is designed for effective pre-training on our Sieve & Swap dataset. It builds on the PDVC model [3, 30], but introduces a number of innovations. First, we note that set-based transformers (DETR [5]) with deformable attention [39] achieved significant improvements for temporal action detection [17, 25] and dense video captioning [30]. However, the naive scaling of prior models, such as PDVC, is non-trivial. For instance, deformable attention struggles to effectively utilize multi-scale temporal features, and LSTM-based text generation lacks parallelism and is difficult to train. To address these issues, ProcX combines key-aware deformable attention with a contrastive transformer for text generation, using an IoU-aware confidence score during inference. We validate each of these improvements in an ablation study and show that ProcX achieves state-of-the-art performance for procedure learning (see Figure 1b).

To summarize, our contributions are:

1. We propose a novel approach, Sieve & Swap, to generate a dataset for pre-training procedural learning models automatically. Our approach reduces the domain gap between pre-training text (transcribed speech) and target text (written instructions).
2. Using this approach, we create the Sieve & Swap dataset for pre-training models for procedural video understanding in the cooking domain. This dataset is three orders of magnitude smaller than previous datasets, allowing efficient pre-training of instruction localization and generation models.
3. We propose Procedure Transformer (*ProcX*), an improved instruction localization and description model, that we pre-train on the Sieve & Swap dataset to achieve state-of-the-art performance on YouCook2 and Tasty.

2 Related Work

Procedural learning is a challenging task in the field of video understanding, as it focuses on acquiring procedural knowledge from long untrimmed videos. Motivated by dense video captioning in open domain videos [12], Zhou *et al.* [38] were the first to introduce a dataset and task in the cooking domain to learn procedures from videos. Annotation styles for procedural learning include fixed,

template-based labels (e.g., COIN [28], CrossTask [40]) and human-written instructions (e.g., YouCook2 [38], Tasty [24]). While the former uses a limited set of labels (e.g., *pour-water*), the latter offers detailed descriptions but is expensive to annotate manually, resulting in smaller datasets (e.g., YouCook2 [38] with ≈1.3K videos). We focus on procedural learning by temporally localizing the events and generating detailed textual instructions. We operate in the cooking domain as it offers a diversity of both natural language and visual activities. Due to task complexity, prior work addresses the problem with two-stage training, where events are first localized and captions are later generated. Wang *et al*. [30] introduced single-stage model (PDVC) to jointly identify temporal boundaries and generate procedural instructions.

Recently, Yang *et al*. [33] introduced Vid2Seq, a new large model that leverages web-scale datasets (HowTo100M [20], YT-Temporal-1B [35]) to enhance performance through pre-training and finetuning. Unlike conventional methods, Yang *et al*. [33] employed speech transcripts as a form of weak supervision. The transcribed speech is given in input during inference as well, which greatly boosts Vid2Seq's performance. However, utilizing transcript sentences converts the problem into text-to-text translation, which makes the model less effective in utilizing the video modality. Furthermore, in many videos the transcribed speech may be unavailable and the model needs to rely on the visual modality alone. The Tasty dataset [24], derived from a web database, serves as an example of a dataset designed for this purpose. In contrast, we focus on models given only video input during training and inference time and reduce the size of the pre-training dataset (48K videos), requiring far less compute. Despite the limited computation budget we achieve state-of-the-art results thanks to our method that compiles a smaller and cleaner pre-training dataset and a more efficient transformer model.

External textual databases [1, 11] have been explored in prior work [16, 37] to curate pre-training data for procedural videos. However, this work focuses on video-segment representation learning, while we focus on generating fine-grained textual instructions along with temporal localization of key steps. Zhou *et al*. [37] employ visual content to align video segments with the nearest step from a textual database using multi-modal embeddings, without relying on ASR text. Conversely, Lin *et al*. [16] match each ASR text to the nearest step in a textual database to learn video-segment representation. Unlike prior work [16, 37], we sieve noisy ASR segments and replace conversational speech with human-written text without visual training. Our method uses five times less text than [16], which uses all ASR-transcribed text. Moreover, the text-replacement approach is crucial for human-style instruction generation.

3 The Task: Localized Instruction Generation

In this work, we are given an untrimmed procedural video showing a sequence of complex instructions that are necessary to complete a goal (e.g., a person

following a cooking recipe). Each instruction is labeled with the start and end times delimiting the instruction temporally, as well as with a fine-grained textual description. Our task is to output the start/end times of each instruction, together with its textual description, given only the untrimmed video as input. This task has been termed "Procedure Segmentation and Summarization" (PSS) in [3] and overlaps with Dense Video Captioning (DVC). In both cases the goal is to detect and describe activities in a video; however, key distinctions are that DVC doesn't involve instructions and that it allows overlapping segments. In our task, instructions are sequential parts of a single procedure (e.g., a recipe), and the focus is to provide a textual sequence summarizing the procedure. In fact, popular datasets for DVC such as ActivityNet Captions [12] collect non-procedural videos, i.e., captions describe what happens in the video, but they do not form the textual sequence of a procedure, which is the focus of our task.

One of the main challenges in localized instruction generation is finding an adequate training dataset since we cannot rely on the large ActivityNet Captions [12]. Other resources provide high-quality annotation of procedural videos (e.g., YouCook2 [38]), however, they are limited in scale and can only be used for fine-tuning or testing. The most common approach to tackle this problem is to pre-train on vast but noisy procedural datasets such a HowTo100M [20], which incurs large computation and storage costs. We instead propose to automatically curate a significantly smaller but cleaner dataset with our Sieve & Swap method, which we detail next. In Section 5 we also introduce a more efficient Transformer for this task.

4 Sieve & Swap

In this section we develop a framework to curate a dataset for efficient pre-training of instructional videos with an automatic pipeline.

4.1 Objective

Let a corpus $\mathcal{V}$ denote the set of untrimmed procedural videos with corresponding ASR (automatic speech recognition) transcripts. Our goal is to sieve out irrelevant transcripts and swap them with human-generated text sourced from an instructional knowledge base $\mathcal{R}$, while ensuring temporal alignment with the videos. In this case, $\mathcal{R}$ is a collection of text-only recipes, each comprising sequential steps. Formally, for each video $\mathbf{v}$ we obtain the transcribed speech with start t^s and end timestamps t^e recognized by an ASR system:

$$\mathbf{v} = (\mathcal{A}, T^v) \tag{1}$$

$$\mathcal{A} = \{(a_j, t^s_j, t^e_j)\}_{j \leq M} \tag{2}$$

where a is the automatically transcribed caption, t^s/t^e are the start/end times of the transcript segment, T^v is the video title and M is the number of ASR segments in the video. Our goal is to remove irrelevant transcript segments and

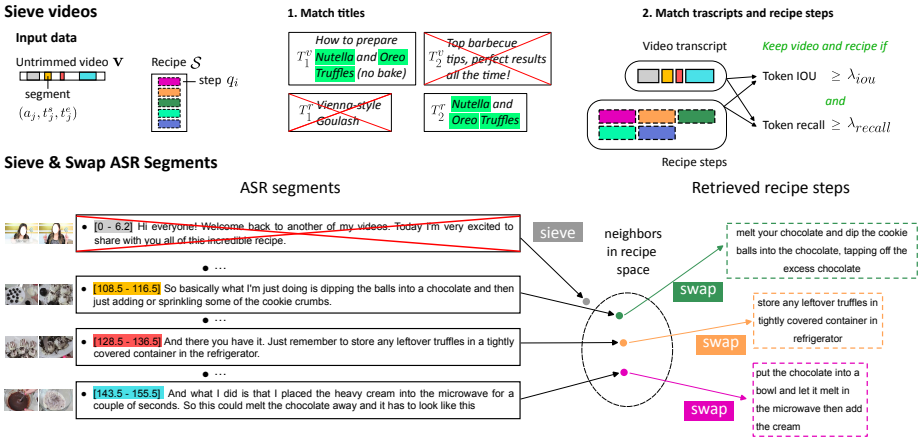

Fig. 2: Sieve & Swap: A technique to sieve and swap ASR transcripts with human written instructions. Top: we first sieve videos and recipes matching their titles and text content. Bottom: for each ASR transcript in each filtered video, we obtain a text embedding and look for the nearest neighbor among the text embeddings of each recipe step from the filtered recipes. If the retrieved neighbor is too far away (gray dots) we consider the transcript segment irrelevant and discard it. Otherwise (colored dots) we replace it with the recipe step corresponding to the retrieved neighbor.

substitute filtered transcripts with text from $\mathcal{R}$ to create an instruction sequence:

$$Sieve\,\&\,Swap(\mathbf{v}) = (\mathcal{A}^{Sieve\,\&\,Swap}, T^v) \tag{3}$$

$$\mathcal{A}^{Sieve\,\&\,Swap} = \{(q_i, t_i^s, t_i^e)\}_{i \leq N} \tag{4}$$

where q_i is the recipe step retrieved from the recipe dataset $\mathcal{R}$ describing the instruction in the corresponding video segment, and N is the number of filtered segments in the video. The number of instructional steps is less than the number of ASR transcript segments, i.e., $N \ll M$. While we replace the transcript text we keep the timestamps of the transcripts. The generated corpus serves as pre-training data for procedure learning, providing language supervision in the form of human-written instruction text rather than noisy ASR transcripts.

4.2 Dataset Creation

Source Datasets. We utilize all cooking videos from HowTo100M (≈338K) and all the cooking recipes with steps from RecipesNLG [4] (≈2M). HowTo100M is the largest dataset for pre-training video models, while RecipesNLG is collected to generate structured text sequences. We keep videos with a maximum duration of 10 minutes and require a minimum of five videos per cooking category (e.g., 'make a bean salad') obtained from video metadata. This set of videos is the corpus $\mathcal{V}$ with ≈241K videos paired with a title containing a high-level goal or a cooking recipe title. We first sieve irrelevant videos and recipes by pairing them based on their title and text content; we then apply Sieve & Swap on ASR segments to

discard irrelevant ASR text (e.g., greetings) and refine segments by substituting ASR transcripts with steps from the recipes database.

Sieving Videos. The source dataset contains a large amount of noisy videos and requires large computational resources. We filter the video dataset by pairing each video $\mathbf{v}$ with a set of textual recipes based on (i) word overlap between video and recipe titles and (ii) word overlap between ASR Transcript and recipe instructions. Let $\mathcal{R} = \{(\mathcal{S}_i, T_i^r)\}$ be the textual recipe dataset, where $\mathcal{S}_i = \{q_1, \ldots, q_H\}$ is the set of procedure steps in a recipe and T_i^r is the recipe title. We assume that the titles T^v and T^r have enough semantic information to map a video $\mathbf{v}$ to a set of recipes in $\mathcal{R}$. With this we refine both $\mathcal{V}$ and $\mathcal{R}$:

$$\mathcal{V}^1 = \{\mathbf{v} = (\mathcal{A}, T^v) \mid \exists\{T^v \cap T^r\} \neq \emptyset;\ \forall \mathbf{v} \in \mathcal{V},\ \forall T^r \in \mathcal{R}\} \tag{5}$$

To calculate the overlap between titles ($T^v \cap T^r$), we consider only content words (nouns, verbs, adverbs, adjectives) and ignore generic recipe words such as 'make', 'prepare', and 'bake'. Videos without a recipe mapping are removed, and recipes that are not paired to any video are likewise removed. We denote the new set of recipes with $\mathcal{R}^1$. The size of $\mathcal{V}^1$ at this stage is ≈110K videos.

We filter $\mathcal{V}^1$ and $\mathcal{R}^1$ further by comparing the content of the transcripts and recipe steps. First, we tokenize the transcripts and recipe steps, followed by lemmatization, and select only content words. Then, we compute the token-IoU and token-recall for each video-recipe pair. Finally, we keep the pairs with token-IoU $\geq \lambda_{iou}$ and token-recall $\geq \lambda_{recall}$. We set $\lambda_{recall} = 0.3$ and $\lambda_{iou} = 0.1$ to obtain the refined video set $\mathcal{V}^2$, which yields ≈52K videos. Later, we split the videos into a train and validation set. Videos with $\lambda_{iou} \geq 0.2$ form the validation set ($\approx 3K$ videos), whereas the remaining ≈48K videos with $0.1 \leq \lambda_{iou} < 0.2$ form the training set. Similarly, we note the refined recipe set with $\mathcal{R}^2$. We use these thresholds to balance storage/computation resources and the resulting noise, however these hyper-parameters can be tuned to increase the size of the dataset. Figure 2 (top) illustrates this part of the dataset creation.

Sieve & Swap ASR Segments. As mentioned earlier, ASR transcripts contain irrelevant content. Using a retrieval-based approach, we replace these transcripts with human-written instructional steps to improve the relevance of the text associated with the video segment. We retrieve the nearest recipe step $q \in \mathcal{S},\ \forall \mathcal{S} \in \mathcal{R}^2$ for each individual ASR transcript segment a (see Eq. 2) based on the cosine similarity between their text embeddings (extracted with MPNet [22, 26]). If the similarity is above a threshold, we swap a with q. Formally, let $\kappa(a)$ be a function that returns the nearest recipe step q for an ASR transcript a, where the similarity is calculated in the text embedding space. For a video $\mathbf{v} = (\mathcal{A}, T^v)$, recall that $\mathcal{A} = \{(a_j, t_j^s, t_j^e)\}$, we have (dropping T^v for clarity):

$$Sieve\,\&\,Swap(\mathbf{v}) = \{(q_j, t_j^s, t_j^e)\}\big|\ q_j = \kappa(a_j)\ \wedge\ sim\big(\phi\langle a_j\rangle, \phi\langle q_j\rangle\big) \geq \lambda_{sim} \tag{6}$$

where $\phi\langle . \rangle$ is the function to compute the embedding vector, sim is the cosine similarity and $\lambda_{sim} = 0.75$. We discard a video segment if its corresponding ASR transcript has a neighbor whose similarity falls below this threshold. The collection of videos with human-written instructional steps is finally defined as:

$$\mathcal{V}^{Sieve\,\&\,Swap} = \{Sieve\,\&\,Swap(\mathbf{v})|\ \forall\ \mathbf{v} \in \mathcal{V}^2\} \tag{7}$$

The size of $\mathcal{V}^{Sieve\,\&\,Swap}$ remains the same as $\mathcal{V}^2$ in terms of videos; however, the number of video segments drops from 2.75M to 0.51M. With this procedure (depicted at the bottom of Figure 2), non-instructional parts of transcripts are removed since our text database contains only recipe steps. In contrast to prior methods, our selection of transcripts with their temporal boundaries helps to bridge the gap between the pre-training and downstream tasks and filters the irrelevant transcripts. Additional analysis of the Sieve & Swap dataset and qualitative examples are provided in the supplementary material.

5 Procedure Transformer (ProcX)

We now present our ProcX model which extendeds PDVC [30]. We first discuss the core elements of PDVC and then introduce several architectural enhancements.

Preliminary: Set-based Localization and Captioning. The PDVC architecture has been inspired by the DETR model [5] and generates $\mathcal{N}$ proposals along with its captions using an LSTM to sequence events. Like DETR, it optimizes various losses using the Hungarian algorithm:

$$\mathcal{L} = \beta_{box}\mathcal{L}_{box} + \beta_{cap}\mathcal{L}_{cap} + \beta_{fl}\mathcal{L}_{fl} + \beta_{count}\mathcal{L}_{count} \tag{8}$$

where $\mathcal{L}_{cap}$ is the cross-entropy loss to predict the next token, $\mathcal{L}_{count}$ is the cross-entropy loss for segment count prediction, $\mathcal{L}_{fl}$ is the focal loss [15] between the positive and negative proposals, and $\mathcal{L}_{box}$ is the gIoU [23] loss between the predicted and ground-truth temporal boundaries. More information about the base model and its architecture can be found in the supplementary material.

Key-aware Deformable Attention. The PDVC employs vanilla deformable attention [39] to reduce the quadratic complexity of dense attention and achieve faster convergence. However, learning the attention weights based solely on the query by a linear projection (i.e., ignoring the query-key product) hampers the efficiency of cross-scale attention. As shown in Figure 3a, the entire scale collapses to focus on a single sampled point in the self-attention of the encoder module. Inspired by object detection techniques [13], we employ key-aware deformable attention for improved acquisition of multi-scale features, allowing diverse sampling across scales (see Figure 3b). More precisely, in vanilla deformable attention each head will sample M points from multi-scale temporal features S as value V based on sampling offsets. The sampling offsets Δp and their

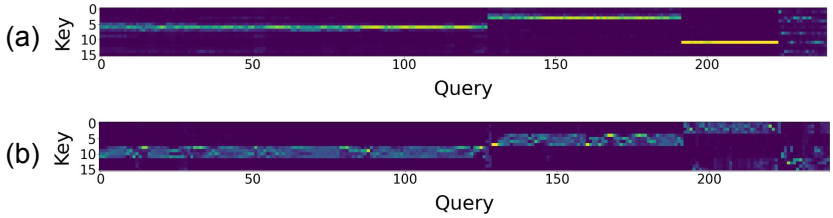

Fig. 3: Attention visualization of one head of the visual encoder. Attention is computed between multi-scale video features and sampled points across scales. In vanilla deformable attention **(a)**, the attention collapses to focus on a single sampled point and fails to attend to the entire video features. With our modified key-aware deformable attention **(b)**, the attention weights spread across the video and help to utilize the entire video context to learn better multi-scale video features.

corresponding attention weights are directly predicted from queries using two linear projections denoted as W_p and W_a. The sampled values V and projected attention weights are utilized to compute the deformable attention:

$$\Delta p = Q \cdot W_p \quad ; \quad V = \Phi(S, p + \Delta p) \cdot W_v \tag{9}$$

$$DeformAttn(Q, V) = \sigma(Q \cdot W_a) \cdot V \tag{10}$$

where Φ is the sampling function with bi-linear interpolation and σ is the softmax function. For the modified key-aware deformable attention, we propose to sample M points for both keys and values from the multi-scale features and utilize standard scaled-dot product-based attention, as formulated below:

$$K = \Phi(S, p + \Delta p) \cdot W_k \quad ; \quad V = \Phi(S, p + \Delta p) \cdot W_v \tag{11}$$

$$KeyDeformAttn(Q, V) = \sigma(Q \cdot K^{\intercal}) \cdot V \tag{12}$$

where W^k is a linear projection of sampled keys. The sampled key points participate in the key-aware deformable attention layer to weight the importance of each sampled value by comparing it with keys, as highlighted in Eq. 10, 12.

Contrastive Transformer. Due to the difficulty of optimizing the conditional likelihood objective for generating captions with vanilla transformer-based models, previous research employed denoising [18, 33] or contrastive losses [34]. We use the contrastive objective, which can be employed both in supervised settings and in pre-training, unlike the denoising objective, which is primarily utilized during pre-training. To learn better multi-modal video-to-text and text-to-video representations, we follow the decoupled architecture from the CoCa model [34] and employ the InfoNCE loss. However, it is difficult to establish a precise one-to-one correspondence between video and text during pre-training due to the misalignment between transcripts and video. We thus incorporate the MIL-InfoNCE loss [19] during pre-training and use standard InfoNCE during finetuning.

IoU-aware Confidence Score (gVFL). PDVC chooses the top-k proposals based on predicted confidence scores during inference. Due to the presence of

separate prediction heads, there is no relationship between the confidence score and the temporal boundary prediction in PDVC. This often results in higher confidence scores for proposals with poor temporal localization. We experimented with assuming oracle temporal localization during inference to confirm this observation. We employed the Hungarian Algorithm to establish a one-to-one mapping between predicted proposals and the ground-truth segments, yielding a substantial gain of ≈20% in segmentation quality. Similar issues in object detection models have been addressed using an IoU-guided confidence loss function [10, 36]. To address this issue in our setting, we replace the focal loss $\mathcal{L}_{fl}$ in Eq. 8 with a modified varifocal loss [36] to optimize the confidence scores. The standard varifocal loss employs the IoU between the predicted and ground-truth segments as a target for confidence score optimization. However, during the initial training stages, there might be no overlap between the proposal and ground truth segments, resulting in zero IoU. This can lead to sub-optimal confidence score optimization. To mitigate this, we use the generalized-IoU (gIoU) metric [23] instead of standard IoU. The gIoU calculates the distance between the predicted and ground-truth segments when there is no overlap (i.e., IoU = 0). More formally, our modified varifocal loss is:

$$\mathcal{L}_{g\text{-}vfl} = \begin{cases} -g(g\log(p) + (1-g)\log(1-p)) & g > 0 \\ -\alpha p^{\gamma}\log(1-g) & g = 0 \end{cases} \tag{13}$$

where p is the IoU-aware confidence score and g is the scaled gIoU between the generated ($\hat{b}$) and ground-truth (b) segment, i.e., $g = 0.5(1 + gIoU(b, \hat{b}))$.

6 Experimental Setup

6.1 Datasets

We evaluate two instructional video datasets: **YouCook2** [38] and **Tasty** [24]. These datasets consist of cooking videos with detailed annotations specifying instruction steps in English sentences along with temporal boundaries. YouCook2 videos are characterized by an unconstrained environment and a third-person viewpoint, while Tasty videos are captured using an overhead camera. Another crucial difference between YouCook2 and Tasty is that Tasty does not include corresponding transcripts. This exemplifies the more realistic and challenging setting of not having step narrations. The assumption of having narrations at test time (as made in Vid2Seq [33]) is not only limiting but also turns the problem into a text-to-text translation rather than a video-to-language problem. Detailed information about the datasets is available in supplementary material.

6.2 Evaluation Metrics

We follow prior work [3, 30, 33] and employ the official evaluation tool from [12] to compute evaluation metrics. The tool first determines the matched pair between the ground truth and predicted segments based on maximum temporal overlap

across IoU thresholds {0.3, 0.5, 0.7, 0.9}. Subsequently, it computes standard captioning metrics, i.e., METEOR [2], CIDER [29] and BLEU-4 [21] over the matched pairs. These metrics fail to account for the sequential coherence of steps, i.e., the extent to which the generated instructions cover the video's sequential progression. We thus additionally report SODA-C [9] for instruction sequences and SODA-D [3] for segment localization. SODA-D takes into consideration the unique nature of non-overlapping segments within instructional videos. We prioritize SODA-C as the primary metric for its more suitable approach, employing temporally ordered and one-to-one matching between predicted and ground-truth instructions. In contrast, prior metrics use one-to-many matching at different IoU thresholds. SODA-C further evaluates text quality through METEOR and F-measure scores, while also penalizing redundancy in instructions/text.

6.3 Implementation Details

We follow [3, 30] and pre-compute segment-level video features for both datasets at a fixed 16 fps, using S3D [19]. Features are resized to a fixed temporal size of 256 via nearest interpolation during pre-training and fine-tuning. For our ProcX model, we set the hyper-parameters for the visual encoder and the localization decoder following PDVC [30]. However, we enhance the capacity of the model by incorporating four layers in both the visual encoder and the localization decoder. To train the instruction generation model we learn a customized vocabulary comprising 10K tokens using Huggingface [31] and employing the WordPiece tokenization algorithm [27]. Further details in the supplementary material.

7 Experimental Evaluation

7.1 Ablation Studies

Sieve & Swap. We developed Sieve & Swap to automatically curate a smaller dataset for pre-training models with a limited budget. Here we validate our approach and report results obtained by training the model on raw transcripts. In Section 4.2 this is the dataset compiled after sieving videos with title and content overlap of words between recipes and ASR text, denoted with $\mathcal{D}_2^v$. It is important to note that videos here still contain raw ASR text segments. Table 1 compares results obtained pre-training on this dataset (Raw ASR) to the dataset obtained after filtering ASR text segments and refining them with Sieve & Swap. All models pre-trained with Sieve & Swap transcripts improve consistently across all metrics despite being pre-trained only on a fifth of the data. This demonstrates the effectiveness of our approach: we improve performance by curating a smaller and better pre-training dataset. We reduce noise by discarding irrelevant ASR transcripts (sieve stage), often filled with sentences unrelated to the instruction (e.g., greetings) and replacing instructional ASR transcripts with recipe steps (swap stage). We show additional ablations on using only-sieved ASR along with combining sieved ASR text and Sieve & Swap text in the supplementary material.

Table 1: Effectiveness of Sieve & Swap. Models were finetuned on YouCook2 and pre-trained with raw ASR transcripts or Sieve & Swap Captions. Segments are the number of video segments (and corresponding captions) used for pre-training. With Sieve & Swap the pre-training data points are reduced by a factor of five, but models still achieve better performance thanks to reduced noise and better captions. We highlight the primary metric SODA-C. Best results per section are in bold, while best results across all models are underlined.

Method	Visual Features	Segments (in Million)	Captions	YouCook2				
				SODA-C	METEOR	CIDER	B4	SODA-D
Vid2Seq [33]	CLIP	2.75	Raw ASR	3.65	3.89	14.57	0.52	29.64
Vid2Seq	CLIP	**0.51**	Sieve-&-Swap (Ours)	**4.21**	**4.19**	**15.86**	**0.61**	**29.71**
PDVC [30]	S3D	2.75	Raw ASR	5.81	6.27	32.44	1.26	32.71
PDVC	S3D	**0.51**	Sieve-&-Swap (Ours)	**6.16**	**6.85**	**36.27**	**1.61**	**32.80**
ProcX (Ours)	S3D	2.75	Raw ASR	6.97	6.34	38.23	1.71	37.08
ProcX (Ours)	S3D	**0.51**	Sieve-&-Swap (Ours)	**7.29**	**6.76**	**40.57**	**1.92**	**37.61**

Table 2: **(a)** Shows the effect of the adding Transformer and modified gIoU guided varifocal loss (gVFL) to PDVC [30]. **(b)** Shows the effect of the Key-Aware Deformable Attention in the video encoder and segmentation decoder.

(a)

Captioner	gVFL	YouCook2				
		SODA-C	METEOR	CIDER	B4	SODA-D
LSTM		5.43	5.40	27.42	0.85	31.48
Transformer	✗	4.94	**5.67**	28.88	1.19	28.64
Transformer	✓	**5.84**	5.34	**30.83**	**1.28**	**33.33**

(b)

Key Aware		YouCook2				
Encoder	Decoder	SODA-C	METEOR	CIDER	B4	SODA-D
✗	✗	5.84	5.34	30.83	1.28	33.33
✓	✗	5.95	**5.40**	**31.20**	**1.29**	33.26
✓	✓	**6.03**	5.35	30.84	1.27	**34.63**

ProcX Architecture Improvements. We systematically study the integration of the contrastive transformer and the gIoU-guided varifocal loss (g-VFL) to PDVC [30]. In Table 2a we show the performance of PDVC [30] with the LSTM captioner as well as the Transformer captioner with and without gVFL. Replacing the LSTM with a contrastive transformer improves caption generation but reduces SODA-D, leading to lower SODA-C scores. We attribute this performance drop to the utilization of a low confidence score for selecting top-k segments. To address this, we introduce a modified varifocal loss with generalized-IoU guidance to enhance the confidence score, improving our ProcX model's performance. Table 2b demonstrates the effectiveness of enhanced key-aware deformable attention in the visual encoder and segment localization decoder. By integrating the key-aware attention mechanism, we observe notable performance improvements in both self-attention during video encoding and cross-attention in the localization decoder, reflected in SODA-D. There is a marginal change in the other *caption* metrics as we introduce this attention mechanism in the *visual* encoding and decoding. The marginal gains from individual components are due to the small size of YouCook2 (0.8K), but overall we observe an improvement in SODA-C by 11.05% and 50.83% on YouCook2 and Tasty (3K) datasets (rows 3, 4 in Table 3).

7.2 Comparison to State of the Art

In the top section of Table 3 (rows 1–4), we compare results without pre-training on two datasets, i.e., models are directly fine-tuned on YouCook2 and Tasty.

Table 3: Comparison of our results on YouCook2 and Tasty with state-of-the-art models. All models receive visual input, but those in gray rows get both visual and ASR text input. Vid2Seq‡ and PDVC‡: results reported from [32]. Vid2Seq† and PDVC†: Results from our run. The evaluation metrics are an average of three different runs for our model.

	Method	Video Features	ASR Text	Pre-training Seg (M)	Pre-training Captions	YouCook2 (val) SODA-C	YouCook2 (val) METEOR	YouCook2 (val) CIDER	YouCook2 (val) B4	YouCook2 (val) SODA-D	Tasty (test) SODA-C	Tasty (test) METEOR	Tasty (test) CIDER	Tasty (test) B4	Tasty (test) SODA-D
Without PreTraining															
1.	Vid2Seq† [33]	CLIP	✗	-	-	3.61	3.83	14.24	0.42	27.16	2.58	3.03	9.48	0.38	35.88
2.	PDVC [30]	CLIP	✗	-	-	4.35	4.27	19.44	0.44	27.29	7.82	8.39	40.03	3.09	43.61
3.	PDVC [30]	S3D	✗	-	-	5.43	**5.40**	27.42	0.85	31.63	7.18	7.87	37.32	2.61	43.51
4.	ProcX (ours)	S3D	✗	-	-	**6.03**	5.35	**30.84**	**1.27**	**34.63**	**10.83**	**10.55**	**70.90**	**6.26**	**51.36**
With PreTraining															
5.	Vid2Seq‡ [33]	CLIP	✗	∼900.0	Raw ASR	5.70	-	25.30	-	-	-	-	-	-	-
6.	Vid2Seq†	CLIP	✗	0.5	Sieve & Swap (ours)	4.21	4.19	15.86	0.61	29.71	7.28	8.57	46.29	4.78	38.94
7.	PDVC‡ [30]	CLIP	✗	7.0	Video Chapters	5.90	**7.50**	34.70	-	-	-	-	-	-	-
8.	PDVC†	CLIP	✗	0.5	Sieve & Swap (ours)	5.48	5.83	30.48	1.33	29.78	8.01	8.72	43.68	3.51	44.71
9.	PDVC†	S3D	✗	0.5	Sieve & Swap (ours)	6.16	6.85	36.27	1.61	32.80	7.64	8.31	40.97	2.84	44.94
10.	ProcX (ours)	S3D	✗	0.5	Sieve & Swap (ours)	**7.29**	6.76	**40.57**	**1.92**	**37.61**	**11.66**	**11.24**	**78.13**	**7.09**	**52.16**
11.	Vid2Seq‡ [33]	CLIP	✓	136.0	Raw ASR	8.60	10.50	53.20	-	-	-	-	-	-	-
12.	Vid2Seq‡ [33]	CLIP	✓	∼143.0	Raw ASR + VC	10.30	12.30	67.20	-	-	-	-	-	-	-

Set-based transformers such as PDVC [3] and our ProcX demonstrate superior performance compared to the sequence-to-sequence model Vid2Seq [33]. ProcX consistently outperforms PDVC, underscoring the effectiveness of an end-to-end transformer-based approach in comparison to an LSTM-based model. ProcX improves SODA-C by 0.6 and 3 points on YouCook2 and Tasty, indicating higher temporal coherence in the generated instructions. ProcX also enhances localization performance (SODA-D) by 3 and 8 points on the two datasets.

In the middle section of Table 3 (rows 5–10), we present results obtained from pre-training models on various datasets and fine-tuning them on the respective downstream datasets. ProcX pre-trained on our Sieve & Swap dataset (row 10) achieves a new state-of-the-art, despite being pre-trained on only 0.5 million segments, as opposed to ≈900 million (row 5) or 7 million (row 7). This shows again both the effectiveness of our Sieve & Swap method to compile a better pre-training dataset and the efficiency of ProcX. Lastly, the bottom section of Table 3 (rows 11–12) includes results from Vid2Seq obtained using transcripts during both training and inference. We note a large gap with all video-only methods, however results are not directly comparable since using ASR text as input during inference gives a substantial advantage for this task. This is because ASR text has a significant content overlap with the generated captions and additionally provides a temporal segmentation of the video.

Zero Shot Procedure Learning. We also test models in a zero-shot setting, where they are only pre-trained and not fine-tuned on the downstream datasets. Here we pre-train models on our Sieve & Swap dataset as well as Video Chapters [32] (for Vid2Seq) and a subset of cooking videos from HowTo100M with raw ASR obtained after sieving (see Section 4.2). This experiment is reported in Table 4. ProcX consistently enhances caption metrics across both datasets, highlighting the efficacy of our new model. Importantly, all models pre-trained on Sieve & Swap dataset perform better than they do when pre-trained on the much larger Video Chapters [32] (7M segments) and the noisier raw ASR sub-

Table 4: Zero-shot performance on YouCook2 and Tasty. We pre-train models without fine-tuning them on the target datasets. Vid2Seq[‡] refers to the official results from VidChapters [32]. The results per section are in bold, while best result across all models are highlighted with underline.

Method	Video	Pre-training		YouCook2 (val)					Tasty (test)				
	Features	# Segments	Captions	SODA-C	METEOR	CIDER	B4	SODA-D	SODA-C	METEOR	CIDER	B4	SODA-D
Vid2Seq[‡] [33]	CLIP	7.0	Video Chapters [32]	0.70	0.50	1.10	-	-	-	-	-	-	-
Vid2Seq[†] [33]	CLIP	2.5	Raw ASR	0.25	0.35	0.64	0.01	4.63	0.91	1.28	1.78	0.01	25.12
Vid2Seq[†]	CLIP	0.5	Sieve & Swap (ours)	**1.33**	**1.73**	**3.83**	**0.04**	**17.52**	**3.32**	**4.52**	**12.52**	**0.56**	**37.55**
PDVC [30]	S3D	2.5	Raw ASR	0.72	0.78	2.39	0.00	10.51	2.27	2.78	6.82	0.13	**31.32**
PDVC	S3D	0.5	Sieve & Swap (ours)	**2.01**	**2.99**	**9.78**	**0.17**	**21.70**	**2.64**	**4.18**	**12.94**	**0.58**	25.52
ProcX (ours)	S3D	2.5	Raw ASR	1.10	1.14	2.86	0.02	17.83	2.65	3.41	7.93	0.35	**35.34**
ProcX (ours)	S3D	0.5	Sieve & Swap (ours)	**2.03**	**3.15**	**7.75**	**0.27**	**22.37**	**2.82**	**6.24**	**19.19**	**1.36**	24.97

set, demonstrating the effective generalization of the Sieve & Swap dataset for different models. We ascribe this to the fact that the Sieve & Swap dataset helps in reducing the domain gap between pre-training text (transcribed speech) and target text (human written instructions). We created the Sieve & Swap dataset with the goal of enhancing caption quality. However, we noticed that it also contributes to improved localization on YouCook2. The performance drop on the Tasty dataset for localization can be attributed to a mismatch in segment duration. Specifically, the average segment duration in Tasty is six seconds, closely matching with shorter transcript segments.

8 Limitations and Future Work

Sieve & Swap achieves state-of-the-art results with limited data and promises further gains by scaling the curated dataset and pre-training with larger compute. However, resource constraints currently hinder scaling, deferring it to future endeavors. Additionally, Sieve & Swap is generic and can be extended to other domains beyond cooking, e.g., leveraging WikiHow or utilizing domain transfer learning [14] from cooking instructional videos to general instructional videos.

9 Conclusion

Understanding procedural videos is a fundamental problem with a range of growing applications. It is challenging and requires solving two tasks: learning to localize steps and generating coherent descriptions for these steps. Models that solve this task need to have a high capacity and therefore require large amounts of training data. Existing pre-training approaches leverage video transcripts to create very large noisy datasets, which take a lot of time to train and have a domain gap between transcribed speech and written step description. To address these issues, we proposed Sieve & Swap, a technique to curate instructional video datasets leveraging language-only datasets of the same domain. The resulting dataset allows us to develop and train a novel large model, ProcX, using three orders of magnitude less data and surpassing existing state-of-the-art performance. Our curation approach is general and will open up new possibilities for training large video understanding models.

Acknowledgements. This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by UKRI grant EP/S022481/1 and the University of Edinburgh, School of Informatics. MR was funded in part by an Alexander von Humboldt Professorship in Multimodal Reliable AI sponsored by Germany's Federal Ministry for Education and Research.

References

1. wikihow. https://www.wikiHow.com/
2. Banerjee, S., Lavie, A.: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. pp. 65–72. Association for Computational Linguistics, Ann Arbor, Michigan (Jun 2005), https://aclanthology.org/W05-0909
3. Batra, A., Gowda, S.N., Keller, F., Sevilla-Lara, L.: A closer look at temporal ordering in the segmentation of instructional videos. In: British Machine Vision Conference (BMVC) (2022)
4. Bień, M., Gilski, M., Maciejewska, M., Taisner, W., Wisniewski, D., Lawrynowicz, A.: Recipenlg: A cooking recipes dataset for semi-structured text generation. In: Proceedings of the 13th International Conference on Natural Language Generation. pp. 22–28 (2020)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
6. Chafe, W., Tannen, D.: The relation between written and spoken language. Annual review of Anthropology **16**(1), 383–407 (1987)
7. Cheng, F., Wang, X., Lei, J., Crandall, D., Bansal, M., Bertasius, G.: Vindlu: A recipe for effective video-and-language pretraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10739–10750 (2023)
8. Einhorn, L.: Oral and written style: An examination of differences. Southern Journal of Communication **43**(3), 302–311 (1978)
9. Fujita, S., Hirao, T., Kamigaito, H., Okumura, M., Nagata, M.: SODA: story oriented dense video captioning evaluation framework. In: European Conference on Computer Vision. pp. 517–531. Springer (2020)
10. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 784–799 (2018)
11. Koupaee, M., Wang, W.Y.: Wikihow: A large scale text summarization dataset. arXiv preprint arXiv:1810.09305 (2018)
12. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: International Conference on Computer Vision (ICCV) (2017)
13. Li, F., Zeng, A., Liu, S., Zhang, H., Li, H., Zhang, L., Ni, L.M.: Lite detr: An interleaved multi-scale encoder for efficient detr. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18558–18567 (2023)
14. Li, J., Chen, G., Tang, Y., Bao, J., Zhang, K., Zhou, J., Lu, J.: Gain: On the generalization of instructional action understanding. In: The Eleventh International Conference on Learning Representations (2022)

15. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
16. Lin, X., Petroni, F., Bertasius, G., Rohrbach, M., Chang, S.F., Torresani, L.: Learning to recognize procedural activities with distant supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13853–13863 (2022)
17. Liu, X., Wang, Q., Hu, Y., Tang, X., Zhang, S., Bai, S., Bai, X.: End-to-end temporal action detection with transformer. IEEE Transactions on Image Processing (TIP) (2022)
18. Luo, H., Ji, L., Shi, B., Huang, H., Duan, N., Li, T., Li, J., Bharti, T., Zhou, M.: Univl: A unified video and language pre-training model for multimodal understanding and generation. arXiv preprint arXiv:2002.06353 (2020)
19. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9879–9889 (2020)
20. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2630–2640 (2019)
21. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
22. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), http://arxiv.org/abs/1908.10084
23. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 658–666 (2019)
24. Sener, F., Yao, A.: Zero-shot anticipation for instructional activities. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
25. Shi, D., Zhong, Y., Cao, Q., Zhang, J., Ma, L., Li, J., Tao, D.: React: Temporal action detection with relational queries. In: European conference on computer vision. pp. 105–121. Springer (2022)
26. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mpnet: Masked and permuted pre-training for language understanding. Advances in Neural Information Processing Systems **33**, 16857–16867 (2020)
27. Song, X., Salcianu, A., Song, Y., Dopson, D., Zhou, D.: Fast wordpiece tokenization. arXiv preprint arXiv:2012.15524 (2020)
28. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1207–1216 (2019)
29. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4566–4575 (2015)

30. Wang, T., Zhang, R., Lu, Z., Zheng, F., Cheng, R., Luo, P.: End-to-end dense video captioning with parallel decoding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6847–6857 (2021)
31. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
32. Yang, A., Nagrani, A., Laptev, I., Sivic, J., Schmid, C.: Vidchapters-7m: Video chapters at scale. In: NeurIPS (2023)
33. Yang, A., Nagrani, A., Seo, P.H., Miech, A., Pont-Tuset, J., Laptev, I., Sivic, J., Schmid, C.: Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10714–10726 (2023)
34. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: Contrastive captioners are image-text foundation models. arXiv preprint arXiv:2205.01917 (2022)
35. Zellers, R., Lu, J., Lu, X., Yu, Y., Zhao, Y., Salehi, M., Kusupati, A., Hessel, J., Farhadi, A., Choi, Y.: Merlot reserve: Neural script knowledge through vision and language and sound. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16375–16387 (2022)
36. Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N.: Varifocalnet: An iou-aware dense object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8514–8523 (2021)
37. Zhou, H., Martín-Martín, R., Kapadia, M., Savarese, S., Niebles, J.C.: Procedure-aware pretraining for instructional video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10727–10738 (2023)
38. Zhou, L., Xu, C., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. In: AAAI Conference on Artificial Intelligence (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17344
39. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2021)
40. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-task weakly supervised learning from instructional videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3537–3545 (2019)