Video Action Recognition with Attentive Semantic Units

Yifei Chen†  Dapeng Chen††  Ruijin Liu⋆  Hao Li⋆⋆  Wei Peng†

1 IIRC, Huawei  2 Xi’an Jiaotong University  3 Xiamen University
chenyifei14@huawei.com  chendapeng8@huawei.com

Abstract

Visual-Language Models (VLMs) have significantly advanced video action recognition. Supervised by the semantics of action labels, recent works adapt the visual branch of VLMs to learn video representations. Despite the effectiveness proved by these works, we believe that the potential of VLMs has yet to be fully harnessed. In light of this, we exploit the semantic units (SU) hiding behind the action labels and leverage their correlations with fine-grained items in frames for more accurate action recognition. SUs are entities extracted from the language descriptions of the entire action set, including body parts, objects, scenes, and motions. To further enhance the alignments between visual contents and the SUs, we introduce a multi-region attention module (MRA) to the visual branch of the VLM. The MRA allows the perception of region-aware visual features beyond the original global feature. Our method adaptively attends to and selects relevant SUs with visual features of frames. With a cross-modal decoder, the selected SUs serve to decode spatiotemporal video representations. In summary, the SUs as the medium can boost discriminative ability and transferability. Specifically, in fully-supervised learning, our method achieved 87.8% top-1 accuracy on Kinetics-400. In K=2 few-shot experiments, our method surpassed the previous state-of-the-art by +7.1% and +15.0% on HMDB-51 and UCF-101, respectively.

1. Introduction

Video action recognition is the fundamental task toward intelligent video understanding. Facilitated by deep learning, great advancements have been made in designing end-to-end network architecture, including two-stream networks [50, 54, 67], 3D convolutional neural networks (3D-CNN) [9, 16, 17, 23, 46, 52, 53, 59] and transformer-based networks [4, 14, 35, 44, 62]. The visual-languages models (VLM) [47, 25] have achieved great success in the last few years. Taking inspiration from this, recent works [55, 26, 41] started exploiting video representation learning under semantic supervision. In particular, these works propose a paradigm that trains video representations to align with action-name text embeddings, which provide richer semantic and contextual information than conventional one-hot labels. Besides showing competitive performance in closed-set action recognition scenarios, adopting VLM demonstrates great learning effectiveness toward recognizing unseen or unfamiliar categories compared to methods that do not utilize text embeddings.

Despite the achieved improvements, we believe that relying solely on the semantic information of action labels is far from enough to fully harness the advantages of the VLMs. This is because action labels are abstract concepts, and directly mapping a video clip to them may confront a noticeable gap. We propose alleviating the problem by relating the action labels with some common entity, ranging from static scenes to moving body parts. For example, “playing golf” can be associated with people, golf equipment, and grass, which could provide fine-grained and specific guidance for visual alignments. We refer to these action-related basic entities as semantic units (SU). In practice, SUs are extracted from discriminative language descriptions of the whole action set, including body parts, objects, scenes, and motions. Our basic idea is to use visual features to select related SUs and then use SUs to guide the decoding of spatiotemporal video features. Compared with the previous approaches, utilizing SUs can bring two advantages: (1) The text corresponding to semantic units can better explain which factors determine an action. (2) The re-usability of semantic units can greatly alleviate learning difficulty in the zero-shot/few-shot scenario.

Another concern arises when we explore fine-grained visual-language correlations through SUs: the visual branch of current VLMs produces frame-level representations, which may hinder the sensitive correspondences to SUs. To
leverage the local visual appearance, we further introduce multi-region attention (MRA) modules upon the pre-trained visual encoders. MRA spatially perceives sub-regions of each image and enhances the visual features with region-aware info, therefore enabling visual features to select SUs globally and locally. The final video representation is produced by a cross-modal video decoder with the selected SUs as the queries and the enhanced visual features from frames as the keys and values. The cross-modal decoder is a stack of cross-attention and 1D temporal convolution hybrid modules. Cross-attention makes the representation focus on the most critical visual appearance by attentive semantic units, while the temporal convolution exploits the motion cue for video representation. In summary, our contributions are threefold: (1) We utilize semantic units to guide the representation learning of actions. Because the semantic units are fine-grained and reusable, the learned representation can improve both the discriminative ability and the transferability. (2) We introduce multi-region attention (MRA) to the visual branch of VLM, which originally provided global representations. MRA perceives region-aware features for each image, enabling sensitive correspondence to multiple fine-grained semantic cues behind actions. (3) We propose a cross-modal decoder that generates the final video representation. This module utilizes attentive semantic units to highlight critical visual appearances while exploiting motion cues.

2. Related Work

Network Architectures in Action Recognition With the significant advances in deep learning, the current video recognition approaches aim to learn effective spatial-temporal representation by well-designed network architectures. The two-stream networks apply 2D CNNs in video recognition [67, 54, 50], where they combine the color image and optical flow to obtain inter-frame cues. The availability of larger video classification datasets, such as Kinetics [27], facilitates the training of 3D CNNs [16, 53, 46, 52, 59, 17, 16]. 3D CNN factorizes the convolution across spatial and temporal dimensions and can capture temporal variation among the local neighborhood. The recent transformer-based approaches advance the state-of-the-art previously set by 3D CNNs, including ViViT [3], Timesformer [4], Video Swim Transformer [35], and Multiview Transformer [62]. Because of the self-attention modules, the transformer-based approaches can capture the long-term spatial-temporal dependencies, leading to better performance. Our method also employs Transformers as backbones: a VIT-like network structure to extract the frame-and-region-level representations and a decoder with semantic-unit attention.

Vision-Language Model in Action Recognition Vision-language models (VLM) are making rapid progress and impressive performance on multiple downstream tasks, including detection [22], segmentation [57], caption [39], summarization [40], etc. There is also a trend to extend VLMs to video action recognition. VideoCLIP [60] replaces image-text with video-text pairs to formulate a video-language pretraining for the downstream action recognition task. However, such an “training from scratch” strategy is data-hungry and computationally expensive. Another pipeline is to finetune upon the available vision-language pre-trained model such as CLIP [47] and ALIGN [25]. ActionCLIP [55] firstly proposes a “pre-train, prompt and finetune” framework. Prompting-CLIP [26] introduces a cross-frame attention mechanism and utilizes the video content embedding to generate the text prompts, which is used to enhance the text embedding. To improve training efficiency, EVL [34] and ST-Adapter [43] fix the CLIP backbone and add a lightweight network module.

Figure 1. **Left:** Compared with conventional video learners that encode frames directly, our method utilizes attentive semantic units (SU) to guide the representation learning of videos. The learned representation contains rich semantics and thus can be effectively adapted to the target representation given by the text representation of the action name. **Right:** With the utilization of semantic units (SU), our method can achieve impressive results on zero/few-shot learning. With the model trained on Kinetics-400, we achieve 48.1% and 70.8% top-1 accuracy in zero-shot and K=16 few-shot experiments, which is 3.5% and 6.6% higher than the second best methods.
to exploit the spatiotemporal information. Unlike all the above methods that leverage semantics by direct mappings, i.e., from video embedding to text features of action label names, our approach further utilizes the capability of pre-trained VLMS by introducing fine-grained semantic units to guide the learning of video representations.

**Human-Object Interaction in Action Recognition** The interactions between the object and the human are essential for action recognition. It was first studied for still images to resolve the ambiguities where two actions have almost identical poses. Towards action recognition tasks, the HOI methods develop a fine-grained manner based on bounding boxes of objects, humans, and scenes [12, 15, 21, 32]. For example, Fang et al. [15] explicitly consider the body parts and their pairwise correlations, then fuse the body representations with the scene and objects. The idea of HOI is then applied to the video data by considering the motion cues. Something-else [37] mainly works on the dynamic interactions between hands and objects. ActionGenome [24] represents an action by a spatiotemporal scene graph, describing the person-object relationships. AKU [36] infer actions from body part movements with the assistance of visual-driven semantic knowledge mining. Our inspiration for exploring action-related entities comes from the above works. However, in contrast to their reliance on detectors for localization and recognition, our approach efficiently leverages text features of semantic units to select the body parts/instances in frames.

### 3. Methodology

The overall framework of our method is illustrated in Fig. 2. Our method aims to learn more transferable and expressive video representation for action recognition by involving “semantic units” (SU) for guidance. We generate and embed semantic units (Section 3.1), then incorporate the semantic units into the network architecture. The overall architecture has three main modules for video representation (Section 3.2). The training details are introduced in Section 3.3.

#### 3.1. Semantic Units

Inspired by cognitive science [29, 64], we believe an effective video representation should describe the visual appearance and temporal variation of the body, object, and scenes. These factors can be represented by languages and have correspondences in visual appearance. We call them “semantic units”.

Semantic units are in the form of discriminative descriptions. We select nouns or phrases from action names. Then we utilize WordNet [38], Wikipedia, and a few manual modifications to generate discriminative descriptions to represent the context of the selected texts. e.g., the word polo is enhanced as polo, a game played on horseback with a long-handled mallet to be distinguished from the meaning polo shirt. Before generation, we remove the stop word in action names via NLTK [5]. On the other hand, we add body-related semantic entities that are sufficiently discriminative by themselves, including head, arms, hands, hips, legs, and feet. They occupy the majority of human activities and are shared by most actions. An illustration of producing semantic units is provided in supplementary materials.

All the semantic units are collected in the set $S = \{s_k\}_{k=1}^K$. We manually categorized them into “body parts”, “objects”, and “scenes”. For a minority of valid words that solely depend on body motions, such as sidekick and squat, we categorize them as motion units. Therefore,

$$S = \{S_{\text{body}}, S_{\text{object}}, S_{\text{scene}}, S_{\text{motion}}\}. \quad (1)$$

We employ the text encoder of CLIP to encode every semantic unit and its elaborative description to feature vectors. The feature vectors of all the semantic units are concatenated to be $S \in \mathbb{R}^{K \times d}$, where $K$ is the number of semantic units, and $d$ is the dimension of the text feature.
3.2. Network Architecture

Given a video segment with $T$ frames, we extract the visual representations $X$ for each frame by a region-aware image encoder, and utilize a semantic query generator to produce the queries $Q$. The visual representations and queries of all the frames, $X$ and $Q$, are fed to the video decoder, which outputs the video representation $z$.

**Region-Aware Image Encoder** The region-aware image encoder, as shown in Fig. 3a, takes the video frame as input and provides the frame-level and region-level representations. The encoder consists of two parts: a ViT initialized by CLIP is used to generate frame-level representations; Two consecutive multi-region attention (MRA) modules are used to generate the region-level representations.

For a frame $I \in \mathbb{R}^{H \times W \times 3}$, we divide the image into $N$ non-overlapping patches following ViT. The patches are then embedded into vectors $\{e_{cls}\}_i^{N+1}$. A learnable embedding $e_{cls}$ is prepended to the sequence of the embedded patches, called [class] token. The embeddings are concatenated to form a matrix, which is further added with the position embeddings $E_{pos}$ to obtain $E \in \mathbb{R}^{(N+1) \times d}$.

$$E = [e_{cls}^T, e_1^T, e_2^T, \ldots, e_N^T]^T + E_{pos}. \quad (2)$$

We feed $E$ to ViT, producing the visual representations $Y$:

$$Y = [y_{cls}, y_1, y_2, \ldots, y_N]^T, \quad (3)$$

where $y_{cls} \in \mathbb{R}^d$ is the frame-level representation according to ViT, and $(y_1, y_2, \ldots, y_N)$, obtained by averaging the patch embeddings of the last two layers of the ViT, are the representations for the corresponding patches. Inspired by strategy from the re-ID community [10, 13], the $N$ patch representations are split into groups based on their affiliation to the four vertically arranged equal-size regions, forming $\{Y_i\}_{i=1}^4$. The procedure is illustrated in Fig. 4. Utilized as the keys and values, patch representations are further fed into an $L$-layer cross-attention (MRA) module, where each group $Y_i$ of them serves the generation of the corresponding region-level representation:

$$r_i^l = MRA^l(y_i^{l-1}, Y_i), \quad l = 1, \ldots, L. \quad (4)$$

In particular, $l$ denotes the index of MRA and $L = 2$, $r_i^1$ are learnable embedding for region $i$ with positional embeddings added, and $r_i^2$ is region-level representation we finally obtain. MRA consists of Multi-Head Attention (MHA) and Multi-Layer Perceptron (MLP), thus can be represented by:

$$MRA(a, b) = c + MLP(c),$$
$$c = a + MHA(a). \quad (5)$$

It is noteworthy that $\{r_i^1\}_{i=1}^4$ are paralleledly computed by MRA via an attention mask. The final output of the region-aware image encoder is:

$$X = \left[ (y_{cls})^T, (r_1^1)^T, (r_2^1)^T, (r_3^1)^T, (r_4^1)^T \right]^T. \quad (6)$$

$X \in \mathbb{R}^{5 \times d}$ combines the frame-level representation and the region-level representations.

**Semantic Query Generator** We generate query feature $Q$ for a frame with $S$ and $X$ as input. As illustrated in Fig. 3b, we feed the representations $X$ to Semantic Attention (SA), a parameter-free block allows each visual representation in $X$ to select the semantic units.

$$w_{ij} = \frac{\langle x_i, s_j \rangle}{\| x_i \| \cdot \| s_j \|}, \quad a_{i,j} = \frac{\exp(w_{ij}/\tau)}{\sum_{j=1}^K \exp(w_{ij}/\tau)}, \quad (7)$$

where $x_i \in \mathbb{R}^d$ and $s_j \in \mathbb{R}^d$ are $i$th row and $j$th row of $X$ and $S$. $w_{i,j}$ is cosine similarity between $x_i$ and $s_j$. With the
of different regions along the feature channel, obtaining
after Eq. 6 and Eq. 8, obtaining
along the feature channel and the temporal dimension,
ated by applying MLP and MEAN over $b$.
4. The initialized query $b$ and denote the visual representations and queries for the
specifically, we perform the cross attention for each region
the region-aware image encoder. The video decoder pro-
SUs’ text embeddings, while the keys and values are from
the representation of the queries to attend to the visual fea-
ture across the frames. It is noteworthy that the input queries
to the video decoder are derived from the weighted sum of
-classes semantic units
under temperature $S$ between the
affinities matrix $A \in \mathbb{R}^{5 \times K}$ between the
representations $X$ and $S$. The query features $Q \in \mathbb{R}^{5 \times d}$ for
frame $I$ is then calculated in a weight-sum manner:
$$Q = AS. \quad (8)$$

Video Decoder Given a video $[I_1, I_2, ..., I_T]$, we generate
the feature maps and queries for each frame according to
Eq. 6 and Eq. 8, obtaining $X \in \mathbb{R}^{T \times 5 \times d}$ and $Q \in \mathbb{R}^{T \times 5 \times d}$:
$$\hat{X} = [X_1, ..., X_i, ..., X_T], \quad \hat{Q} = [Q_1, ..., Q_i, ..., Q_T]. \quad (9)$$

The video decoder, as shown in Fig. 3c, consists of a sequence of a 1D-convolution layer and a cross-attention (CA) module, whose processing mechanism is similar to
Eq. 5. The 1D-convolution layer processes the queries
along the temporal dimension, while the CA module utilizes
the representation of the queries to attend to the visual fea-
ture across the frames. It is noteworthy that the input queries
to the video decoder are derived from the weighted sum of
SU's text embeddings, while the keys and values are from
the region-aware image encoder. The video decoder pro-
motes communication between the two modalities. More
specifically, we perform the cross attention for each region
and denote the visual representations and queries for the $r$th
region as $\hat{X}_i \in \mathbb{R}^{T \times d}$ and $\hat{Q}_i \in \mathbb{R}^{T \times d}$. The procedure of
the video decoder is as follows:
$$\hat{H}_i^m = \hat{Q}_i^{m-1} + 1D\text{conv}(\hat{Q}_i^{m-1})$$
$$\hat{Q}_i^m = CA^m(\hat{H}_i^m, \hat{X}_i), \quad m = 1, ..., M. \quad (10)$$

where $m$ indicates the layer index of the decoder and $M = 4$.
The initialized query $\hat{Q}_i^0 = Q_i$, and the output query
after $M$-times attention is $\hat{Q}_i^M$. We concatenate the queries
of different regions along the feature channel, obtaining
$\hat{Q}_i^M \in \mathbb{R}^{T \times 5 \times d}$. The video representation $z$ is a vector calculated by applying MLP and MEAN over $\hat{Q}_i^M$ sequentially
along the feature channel and the temporal dimension,
$$z = \text{MEAN}(\text{MLP}(\hat{Q}_i^M)). \quad (11)$$

3.3. Training Details

Loss function We supervise the video representation with the text features of the action labels. Specifically, we feed
action names with a prefixed template “a video of a per-
son” into the frozen text encoder initialized by CLIP. The
resulting text feature of the $i$th action is $c_i$. The goal of the
proposed network is to maximize the similarity between $z_n$ and $c_i$ if $z$ belongs to the $i$th action. The loss function can be implemented by the cross-entropy loss:
$$\mathcal{L} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{I} y_{i,n} \log \left( \frac{\exp(c_i^\top z_n)}{\sum_{j=1}^{I} \exp(c_j^\top z_n)} \right), \quad (12)$$

where $z_n$ the representation of the $n$th video. The training set has $N$ videos belonging to the $I$ actions. If the $n$th video belongs to the $i$th action, $y_{i,n} = 1$, otherwise $y_{i,n} = 0$.

Network Training The ViT backbone in the region-aware
image encoder is initialized by CLIP, while the other param-
eters in the region-aware image encoder and video decoder
are randomly initialized. For network parameter training,
we adopt an AdamW optimizer with an initial learning rate
of $8 \times 10^{-6}$ and $8 \times 10^{-5}$ for the ViT backbone and the re-
mainder parts, respectively. We train the network with 30
epochs and a weight decay of 0.001. The learning rate is
warmed up for 5 epochs and decayed w.r.t. a cosine sched-
ule. The input video follows the main sparse sampling
method [54] and augmentation strategy [41] with a frame
resolution $224 \times 224$. The hyperparameter temperature of
softmax in the semantic query generator is set to 0.01. All
the experiments are conducted with 8 32GB-V100-GPUs.
More details are provided in supplementary materials.

4. Experiments

Datasets We benchmark our method on four popular
video action recognition datasets. (1) Kinetics-400 [27]
is a collection of around 240k training and 20k validation
videos for 400 classes. Each clip covers around 10s. (2)
Kinetics-600 [8] is an extension from Kinetics-400, which
consists of around 410k training and 29k validation videos
for 400 classes. (3) UCF-101 [51] contains 13,320 video
clips with 101 classes. There are three splits of test data. (4)
HMDB-51 [28] consists of 7,000 videos with 51 classes and
has three test data splits. We conduct fully-supervised ex-
periments on Kinetics-400 and Kinetics-600. With the pre-
trained model on Kinetics-400, we conduct few-shot and
zero-shot experiments on UCF-101 and HMDB-51.

4.1. Fully Supervised Comparison

Settings We conduct the fully-supervised experiments on
Kinetics-400 and Kinetics-600. Each video is sampled with
8 or 16 frames. The multi-view inference is with 3 spatial
crops and 4 temporal clips. Three variants of the proposed
Table 1. Comparison with state-of-the-art on Kinetics-400. FLOPs and throughput are reported per view with tools provided by X-CLIP [41]. For multi-modal approaches, parameters in the text branch are not counted. * indicates pretraining with a video-text collection.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pretrain</th>
<th>Frames</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Views</th>
<th>#Param. (M)</th>
<th>FLOPs (G)</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Methods with ImageNet or web-scale image pretraining</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MViTv1-B, 64×3 [14]</td>
<td>-</td>
<td>64</td>
<td>81.2</td>
<td>95.1</td>
<td>3×3</td>
<td>36.6</td>
<td>455</td>
<td>7</td>
</tr>
<tr>
<td>Uniformer-B [30]</td>
<td>IN-1k</td>
<td>32</td>
<td>83.0</td>
<td>95.4</td>
<td>4×3</td>
<td>50.0</td>
<td>259</td>
<td>-</td>
</tr>
<tr>
<td>TimeSformer-L [4]</td>
<td>IN-21k</td>
<td>96</td>
<td>80.7</td>
<td>94.7</td>
<td>1×3</td>
<td>121.4</td>
<td>2380</td>
<td>3</td>
</tr>
<tr>
<td>Mformer-HR [44]</td>
<td>IN-21k</td>
<td>16</td>
<td>81.1</td>
<td>95.2</td>
<td>10×3</td>
<td>381.9</td>
<td>959</td>
<td>-</td>
</tr>
<tr>
<td>Swin-L [35]</td>
<td>IN-21k</td>
<td>32</td>
<td>83.1</td>
<td>95.9</td>
<td>4×3</td>
<td>200.0</td>
<td>604</td>
<td>6</td>
</tr>
<tr>
<td>Swin-L(384↑) [35]</td>
<td>IN-21k</td>
<td>32</td>
<td>84.9</td>
<td>96.7</td>
<td>10×5</td>
<td>200.0</td>
<td>2107</td>
<td>-</td>
</tr>
<tr>
<td>MViTv2-L(312↑) [31]</td>
<td>IN-21k</td>
<td>64</td>
<td>81.2</td>
<td>95.1</td>
<td>3×3</td>
<td>218.0</td>
<td>455</td>
<td>7</td>
</tr>
<tr>
<td>ViViT-H/16 [3]</td>
<td>JFT-300M</td>
<td>32</td>
<td>84.9</td>
<td>95.8</td>
<td>4×3</td>
<td>647.5</td>
<td>8316</td>
<td>-</td>
</tr>
<tr>
<td>MTV-H/16 [62]</td>
<td>JFT-300M</td>
<td>32</td>
<td>85.8</td>
<td>96.6</td>
<td>4×3</td>
<td>-</td>
<td>3705</td>
<td>-</td>
</tr>
</tbody>
</table>

| **Methods with web-scale language-image pretraining** |                |                |        |        |       |             |           |            |
| MTV-H/16 [62]                | WTS-17B*       | 32             | 89.1   | 98.2   | 4×3   | -           | 3705      | -          |
| PromptingCLIP-B/16 [26]      |                | 16             | 76.9   | 93.5   | 5×5   | 95.5        | -         | -          |
| ActionCLIP-B/16 [55]         |                | 16             | 82.6   | 96.2   | 10×3  | 105.2       | 282       | -          |
| ActionCLIP-B/16 [55]         |                | 32             | 83.8   | 97.1   | 10×3  | 105.2       | 563       | -          |
| EVL-B/16 [34]                |                | 32             | 84.2   | 96.6   | 3×1   | 114.9       | 1777      | -          |
| EVL-L/14 [34]                |                | 32             | 87.3   | 97.6   | 3×1   | 357.9       | 2696      | -          |
| EVL-L/14(336↑) [34]          |                | 32             | 87.7   | 97.8   | 3×1   | 357.9       | 6065      | -          |
| X-CLIP-B/16 [41]             |                | 8              | 83.8   | 96.7   | 4×3   | 131.7       | 145       | 33         |
| X-CLIP-B/16 [41]             |                | 16             | 84.7   | 96.8   | 4×3   | 131.7       | 287       | 17         |
| X-CLIP-L/14 [41]             |                | 8              | 87.1   | 97.6   | 4×3   | 451.2       | 658       | 8          |
| X-CLIP-L/14(336↑) [41]       |                | 16             | 87.7   | 97.4   | 4×3   | 451.2       | 3086      | 2          |
| ASU-B/16 (ours)              | CLIP-400M      | 8              | 84.1   | 96.3   | 4×3   | 140.2       | 146       | 30         |
| ASU-A/16 (ours)              | CLIP-400M      | 16             | 84.8   | 96.7   | 4×3   | 140.2       | 288       | 14         |
| ASU-L/14 (ours)              | CLIP-400M      | 8              | 87.8   | 97.8   | 4×3   | 425.3       | 660       | 8          |
| ASU-L/14(336↑) (ours)        | CLIP-400M      | 16             | 88.3   | 98.0   | 4×3   | 425.3       | 3084      | 2          |

Table 2. Comparison with state-of-the-art on Kinetics-600.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pretrain</th>
<th>Frames</th>
<th>Top-1</th>
<th>Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTV-H/24, 32×3 [14]</td>
<td>-</td>
<td>32</td>
<td>83.8</td>
<td>5×1</td>
</tr>
<tr>
<td>Swin-L(384↑) [55]</td>
<td>IN-21k</td>
<td>32</td>
<td>86.1</td>
<td>10×5</td>
</tr>
<tr>
<td>ViViT-H/16x2 320 [3]</td>
<td>JFT-300M</td>
<td>32</td>
<td>83.0</td>
<td>4×3</td>
</tr>
<tr>
<td>ViViT-H/16x2 [3]</td>
<td>JFT-300M</td>
<td>32</td>
<td>85.8</td>
<td>4×3</td>
</tr>
<tr>
<td>TokenLearner-L/10 [49]</td>
<td>JFT-300M</td>
<td>32</td>
<td>86.3</td>
<td>4×3</td>
</tr>
<tr>
<td>Florence(384↑) [63]</td>
<td>FLD-900M</td>
<td>32</td>
<td>87.8</td>
<td>4×3</td>
</tr>
<tr>
<td>CoVer [65]</td>
<td>JFT-3B</td>
<td>96</td>
<td>87.9</td>
<td>1×3</td>
</tr>
</tbody>
</table>

| MTX-H [62]                   | WTS-17B        | 32             | 89.6   | 4×3   |
| X-CLIP-B/16 [41]             |                | 8              | 85.3   | 4×3   |
| X-CLIP-B/16 [41]             |                | 16             | 85.8   | 4×3   |
| X-CLIP-L/14 [41]             |                | 8              | 88.3   | 4×3   |
| ASU-B/16 (ours)              |                | 8              | 85.7   | 4×3   |
| ASU-L/14 (ours)              |                | 16             | 86.4   | 4×3   |
| ASU-L/14 (ours)              |                | 8              | 88.6   | 4×3   |

network, i.e., ASU-B/16, ASU-L/14, and ASU-L/14-336, adopt ViT-B/16, ViT-L/14, and ViT-L/14-336 for the region-aware image encoder, respectively.

**Results** In Tab. 1, we compare with the state-of-the-art methods on Kinetics-400. Under the standard 224×224 input resolution setting, our method achieves the second place 87.8%, which is 0.5% higher top-1 accuracy than the third-ranked method EVL [34]. It is noteworthy that MTV-H [62] performs the best by adopting much more pre-training data (70M video-text pairs with about 17B images [41]) and a larger pre-trained model(Vit-H). Our method outperforms MTV-H pre-trained on JFT-300M by 2.0% and achieves the best results among the CLIP-based methods. Under 336×336 input resolution setting, our method achieves the best Top-1 accuracy of 88.3%, which improves upon the previous highest result (X-CLIP [41] and EVL [34]) by 0.6%. Tab. 2 presents the results on Kinetics-600. Our ASU outperforms X-CLIP by 0.4% and 0.6% by using 8 and 16 frames per video, respectively. Moreover, our ASU achieves 88.6% with 4× fewer input frames and 42.5× fewer pre-training data compared to the current state-of-the-art method MTV-H. Our method achieves superior results among the methods that adopt similar-level pre-trained models. We attribute the effectiveness to the proposed visual-semantic attention mechanism, where the attentive semantic units can help exploit critical visual information for learning discriminative video representation.
4.2. Few-shot Comparisons

Settings The few-shot experiments are evaluated on HMDB-51 and UCF-101. The training set is constructed by randomly sampling 2, 4, 8, and 16 videos from each class and setting the frame number in each video to 8 or 32. Following [41], we adopt the first split of the test set as an evaluation set and report the result of a single-view inference.

Results In Tab. 3, we present the results of $K$-shot learning. Compared with methods employing single-modality pre-trained models, our method outperforms them by a large margin. e.g., when $K=2$, it improves Swin-B by 39.2% on HMDB-51 and 38.1% on UCF-101, demonstrating the effectiveness of the semantic knowledge in the few-shot settings. Moreover, our method still achieves state-of-the-art performance compared with the methods that employed the cross-modal pre-trained model. On HMDB-51, our method with eight frames per video achieves 57.7% top-1 accuracy under $K=2$-shot learning, which is higher than the previous best ActionCLIP with 2.7%. On UCF-101, our method achieves 91.4% top-1 accuracy under $K=2$-shot learning, which outperforms X-Florence by 7.4%. Our method has achieved consistent superiority from $K=2$ to $K=16$, indicating the effectiveness of semantic units.
of the validation set of UCF-101.

Effects of Proposed Components We first investigate the effectiveness of the proposed components. The experiments are conducted under fully-supervised learning of Kinetics-400. (1) We treat the ActionCLIP-B/16-Transf, which introduces a 6-layer self-attention transformer followed by a mean pooling layer stacking on top of the CLIP image encoder as the baseline. Results are presented in Tab. 6, and the first row denotes the baseline. (2) Compared with the baseline, replacing the transformer of baseline with our video decoder and introducing semantic units to generate queries for cross attention bring a 0.9% gain in term of the top-1 accuracy. The results demonstrate the effectiveness of enhancing visual representations with semantic embedding. (3) To validate the effectiveness of region-level representations, we disable the semantic-guided enhancement branch by feeding randomly initialized query embedding into the video decoder and retraining the model. The region-level representation can improve the top-1 accuracy of baseline by 0.7%. (4) Combining the region-level representation and semantic unit further improve the value by 0.6%. The results indicate using proper region-level representation as enhancement can help alignments about instances and texts. (5) Besides, equipping temporal relations in video decoder can further boost the top-1 accuracy by 0.2%. Overall, our proposed components can raise the top-1 accuracy of the baseline from 81.1% to 82.6%.

Investigation of Semantic Units To validate the effect of semantic units, we first construct a baseline model, which replaces the queries to the video decoder with visual features from the region-aware image encoder. The results are reported in the first row in Tab. 7. Then we investigate the effectiveness of each sub-collection in semantic units as defined in Eq. 1. From Tab. 7, we can observe the models equipped with semantic units improves the baseline, especially for the 2-shot (+5.1%) and 0-shot (+10.1%) experiments. The results indicate that visual representations without semantic guidance are not robust when facing a severe lack of data. We also conduct experiments to evaluate the four types of semantic units. In particular, the set of “objects” is the most helpful to the results, indicating the objects are discriminative factors in the action recognition. The results in rows 2,3,4,5 show that each part of semantic units can bring performance gain, and their combination (in row 6) can achieve the best results. The phenomenon indicates that different types of semantic units complement each other, and languages benefit action recognition performance, especially in the transferability of the model.

Effects of Different Region Configurations We conduct experiments with different spatial configurations to exploit spatial information for better action recognition. In particular, we split the whole image into regions according to two aspects: the number of split regions and the approach of spatial slicing. Results are presented in Tab. 8. There are two observations. (1) Increasing the number of partitions will not continuously improve performance. When the size of each region is too large, the visual representations cannot align with the semantic units of small targets. On the contrary, when the size of each region is too small, the receptive field of each region cannot cover the visual appearance of most semantic units. (2) Vertical divisions cannot bring gain to the performance. We propose an assumption for the phenomenon. In an action video, the variations of the body parts usually happen in horizontal regions so that the horizontal regions can keep the majority of part structures across the frames, which can help better describe an action. In comparison, the vertical divisions will break such structures, leading to a performance decrease.

5. Discussion

Conclusion In this paper, we propose a new video representation learning framework for action recognition based on pre-trained visual-language models. To leverage fine-grained visual-language correlation, we introduce “semantic units” that provide factorized and reusable textual knowledge hiding behind actions. A multi-region attention module that perceives region-aware information is proposed to better capture fine-grained alignments. Under the
guidance of selected semantic units, a cross-modal decoder is designed for decoding spatiotemporal video representations. Our approach produces discriminative video representations, outperforming the state-of-the-art approaches in fully supervised comparisons. The semantic units’ reusability further alleviates the learning difficulty of adapting our model to new scenarios, achieving superior results in the zero-shot/few-shot settings.

**Limitations and Future Work** There are two potential improvements for ASU to explore in the future. Firstly, the generation of semantic units still requires a few manual operations. With the increasing maturity of large-scale language models such as the GPT series [7, 42], the current procedure can be fully automated with appropriate prompts and CoT [58] designs. Secondly, ASU and other recent VLM-based approaches consume significant computational resources on pre-trained backbones during the inference phase. From the perspective of practical application, incorporating more computational-efficient techniques is a valuable research field.

**References**


[25] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation

[26] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In ECCV (35), pages 105–124, 2022. 1, 2, 6


[31] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Manmatha, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In CVPR, pages 4794–4804, 2022. 6


[34] Zhihui Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen CLIP models are efficient video learners. In ECCV (35), pages 388–404, 2022. 2, 6


[41] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In ECCV (4), pages 1–18, 2022. 1, 2, 5, 6, 7


[54] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaou Tang, and Luc Van Gool. Temporal segment
networks: Towards good practices for deep action recognition. In ECCV (8), pages 20–36, 2016. 1, 2, 5


[65] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M. Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition. CoRR, abs/2112.07175, 2021. 6

