LoCATe-GAT: Modeling Multi-Scale Local Context and Action Relationships for Zero-Shot Action Recognition

Sandipan Sarma ¹, Divyam Singal ², and Arijit Sur ²

¹Indian Institute of Technology Guwahati
²Affiliation not available

December 7, 2023

Abstract

The increasing number of actions in the real world makes it difficult for traditional deep-learning models to recognize unseen actions. Recently, pretrained contrastive image-based visual-language (I-VL) models have been adapted for efficient zero-shot scene understanding, with transformers for temporal modeling. However, the significance of modeling the local spatial context of objects and action environments remains unexplored. In this work, we propose a framework called LoCATe-GAT, comprising a novel Local Context-Aggregating Temporal transformer (LoCATe) and a Graph Attention Network (GAT) that take image and text encodings from a pretrained I-VL model as inputs. Motivated by the observation that object-centric and environmental contexts drive both distinguishability and functional similarity between actions, LoCATe captures multiscale local context using dilated convolutional layers during temporal modeling. Furthermore, the proposed GAT models semantic relationships between classes and achieves a strong synergy with the video embeddings produced by LoCATe. Extensive experiments on two widely-used benchmarks UCF101 and HMDB51 show we achieve state-of-the-art results. Specifically, we obtain absolute gains of 2.8% and 2.3% on these datasets in conventional and 8.6% on UCF101 in generalized zero-shot action recognition settings. Additionally, we gain 18.6% and 5.8% on UCF101 and HMDB51 as per the recent TruZe evaluation protocol.
LoCAte-GAT: Modeling Multi-Scale Local Context and Action Relationships for Zero-Shot Action Recognition

Sandipan Sarma, Divyam Singal, Arijit Sur

Abstract—The increasing number of actions in the real world makes it difficult for traditional deep-learning models to recognize unseen actions. Recently, pretrained contrastive image-based visual-language (I-VL) models have been adapted for efficient “zero-shot” scene understanding, with transformers for temporal modeling. However, the significance of modeling the local spatial context of objects and action environments remains unexplored. In this work, we propose a framework called LoCAte-GAT, comprising a novel Local Context-Aggregating Temporal transformer (LoCAte) and a Graph Attention Network (GAT) that take image and text encodings from a pretrained I-VL model as inputs. Motivated by the observation that object-centric and environmental contexts drive both distinguishability and functional similarity between actions, LoCAte captures multi-scale local context using dilated convolutional layers during temporal modeling. Furthermore, the proposed GAT models semantic relationships between classes and achieves a strong synergy with the video embeddings produced by LoCAte. Extensive experiments on two widely-used benchmarks – UCF101 and HMDB51 – show we achieve state-of-the-art results. Specifically, we obtain absolute gains of 2.8% and 2.3% on these datasets in conventional and 8.6% on UCF101 in generalized zero-shot action recognition settings. Additionally, we gain 18.6% and 5.8% on UCF101 and HMDB51 as per the recent “TruZe” evaluation protocol.

Index Terms—Zero-shot learning, action recognition, transformer, graph attention network.

I. INTRODUCTION

THE ability of humans to analyze motion patterns and perceive an action subsequently follows their understanding of object recognition. In the last decade or so, computer vision as a research area has improved in leaps and bounds following the deep learning revolution, with CNNs surpassing human performance in object recognition. Naturally, video understanding has been gaining immense attention recently, where human action recognition (HAR) in videos is among the front-runners [1]. Realizing that machines mimicking human intelligence for HAR can have potentially explosive real-world applications, several studies have been conducted in areas like video surveillance, autonomous driving, sports analysis, and others. However, these methods are restricted by the curse of deep learning – lack of large-scale annotated training data.

Sandipan Sarma and Arijit Sur are with the Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati, Assam 781039, India (e-mail: sandipan.sarma@iitg.ac.in; arijit@iitg.ac.in). Divyam Singal contributed to this work during his time at the Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati, Assam 781039, India (email: divyamsingal01@gmail.com).

Fig. 1. Illustration of how object-centric and environmental context plays a dual role in zero-shot HAR – (a) similar spatiotemporal action differentiation based on object dependency (in red labels); (b) differentiation by action environment; (c) human behavior to functionally-similar objects (in blue labels); (d) human behavior in similar environments.

Zero-shot learning (ZSL) [2] approaches have recently emerged to alleviate this problem. Contrary to supervised learning, where all categories to be recognized are predefined, a zero-shot paradigm learns from the visual data of only a few seen classes during training. At test time, it achieves knowledge transferability from the seen to unseen domains/classes. Beyond methods based on generative models [3], [4], knowledge graphs [5], [6], and transformers [7], a new line of work based on large-scale pretraining of contrastive image-based visual-language (I-VL) models [8], [9] seems very promising in addressing zero-shot HAR. The impressive performance of these models across a wide set of vision tasks stems from a strong representation learning using millions of image-text pairs publicly available on the web, allowing even zero-shot capabilities [8]. Consequently, a few recent works have tried efficiently adapting these I-VL models to downstream video tasks. In the context of HAR, learning video-specific/task-specific prompt vectors to obtain better discriminative text representations for action classes and temporal modeling using transformers [10], [11] have been explored recently. However, transformers usually have a limited ability to consider local spatial context, which could be a major hindrance in contextual information flow during temporal modeling. When distinguishing two actions, the object of interaction can play an important role even if the spatiotemporal motion trajectories of the body parts are relatively similar (e.g., throwing a hammer vs. discus). Even for differentiating actions independent of
any object of interest, an additional semantic extracted from video frames can be useful (e.g., understanding “the presence of water” can differentiate breaststroke and horse riding). On the other hand, humans also tend to interact similarly with functionally-similar objects (punching bags and speed bags) and environments (water bodies). Therefore, our intuition is that local contextual cues from objects and environments play a dual role — fueling both distinguishability and functional similarity. This can be detrimental to zero-shot HAR for efficient knowledge transfer from seen to unseen actions.

In a multimodal approach like zero-shot, how well we capture class relationships plays a vital role in how strong the semantic bridge between seen and unseen classes will be. Knowledge graphs (KGs) have proven helpful as a structured medium for modeling prior class knowledge, and graph neural networks (GNNs) have been exploited in the recent past that leverage these KGs for transferring seen knowledge to unseen [5], [12], [15] like Word2Vec [16] and GloVe [17] for initializing the node representations while working in the zero-shot setting. However, action names can be defined using multiple words as well, such as apply eye makeup, in which case an average word embedding for all words in the class name is used as node representation. Ghosh et al. [13] show that this approach does not always capture the correct relationships between classes and demonstrate improvements using Sent2Vec embeddings [18] instead. On the contrary, we feed a handcrafted prompt to the text encoder of CLIP for every action class, and the output representations initialize the class nodes in our KG. Then, a Graph Attention Network (GAT [19]) is trained to model the relationships of a node with its neighbors via an attention mechanism. Contrary to the widely-used GCNs [20] for knowledge transfer [6], GATs consider different statistical strengths for different neighbors of a node, producing better embeddings and achieving substantial improvements over GCN.

The main contributions of this work are as follows:

- We propose a novel temporal transformer called LoCATe that looks for frame-wise local spatial context and aggregates them to account for temporal dependencies to perform zero-shot HAR in videos. We feed visual embeddings obtained from CLIP to LoCATe, which has a hierarchical structure to capture multi-scale context.
- To the best of our knowledge, this is the first work that uses GAT for modeling action relationships in a zero-shot setting. Our GAT demonstrates two aspects in HAR: 1) the importance of node initialization using powerful textual representations from CLIP, and 2) the strong synergy achieved between CLIP-based temporal modeling and attention modeling for action classes using GAT which provides an improved alignment between visual–semantic spaces.

- Extensive experiments show that our framework has the ability to tackle the polysemy of action verbs. We evaluate our framework on two challenging zero-shot HAR benchmarks – UCF101 and HMDB51 – and outperform the state-of-the-art on both of them in the conventional setting. In the generalized zero-shot setting, we obtain an absolute gain of 8.6% over the existing methods on UCF101 and comparable results on HMDB51.

II. RELATED WORK

Zero-shot action recognition (ZSAR). The initial works in ZSAR extended the idea of zero-shot image classification [2],

Fig. 2. Polysemy of action verbs (shown in red). The human pose may differ significantly depending upon a primary object of interest (top row), the environment/field of activity (middle row), or a mixture of both (bottom row).
Fig. 3. **Proposed framework.** Given video frames as input, a CLIP image encoder produces the frame encodings. Our novel temporal transformer takes these, along with temporal positional encodings, as input and passes them down the encoder structure. The intermediate output goes through an LCA block, where local context is aggregated from multiple scales. Temporal averaging produces video embeddings. Additionally, semantic relationships between actions are learned using a graph attention network (GAT). Visual-semantic alignment of video and semantic embeddings using a cosine similarity enables ZSAR.

Although I-VL models [8], [9] have shown impressive zero-shot transferability of prior knowledge, acquiring a large amount of labeled video-text pairs for training them is a significant obstacle in HAR. Recently, ActionCLIP [36] became the first to introduce a pretrain, prompt and finetune paradigm in HAR. Ju et al. [10] additionally attach a temporal transformer and formulate several video understanding tasks under the same umbrella. XCLIP [11] builds a cross-frame transformer for frame-level message passing and then integrates them to get the video features. However, they ignore the local spatial context in the frames while establishing temporal dependencies.

**Graph Neural Networks.** Since their inception, knowledge graphs (KGs) have been extensively used to mimic the human ability to use structured prior knowledge and build relationships between concepts [37]. In HAR, GCNs [20] have been the go-to networks for spatiotemporal modeling of actions [15], where a convolution operation aggregates information from node neighbors, calculating static weights based on node degrees. [5], [6], [13], [15], [38] use GCNs for modeling semantic relationships between actions. In this work, we use a Graph Attention Network [19] (GAT) and aim to improve upon GCN-based action representations by incorporating multi-head attention over the node features, emulating the practical scenario that an action class has a dynamic relationship with its neighbors. As per [13], initial node representation drives the learning in the later stages. Hence, we choose to initialize the nodes with the robust text encodings produced by CLIP.

### III. Approach

**A. Problem formulation**

In ZSAR, we are given a training set $D_{train} = \{(v_i, l_i)\mid v_i \in \mathcal{V}_{seen}, l_i \in \mathcal{S}\}$, where $v_i$ is a video clip of a seen class $l_i$. A separate set of novel (unseen) data $D_{novel} = \{(v_j, l_j)\mid v_j \in \mathcal{V}_{unseen}, l_j \in \mathcal{U}\}$ is given such that the sets of seen and unseen classes are disjoint, i.e., $\mathcal{S} \cap \mathcal{U} = \phi$. We work...
in the more realistic inductive ZSAR setting – where video samples of unseen classes are unavailable during training – instead of the transductive setting [12], [39]. Then, the task in conventional ZSAR (CZSAR) is to learn a classifier:

\[ f_{\text{zsl}} : V_{\text{unseen}} \rightarrow \mathcal{U} \]  

(1)

For generalized ZSAR (GZSAR), a small subset of \( V_{\text{seen}} \) (\( V_{\text{unseen}}^{\text{sub}} \)) is used as the set of seen samples at test time. The objective changes to learning a classifier:

\[ f_{\text{gzsl}} : V_{\text{seen}}^{\text{sub}} \cup V_{\text{unseen}} \rightarrow \mathcal{S} \cup \mathcal{U} \]  

(2)

B. System overview

Inspired by the recent success of I-VL models adapted to the ZSAR task [10], [11], [36], we propose to use CLIP [8] to get video and text encodings. Specifically, during training, given a video clip \( v_i \in \mathbb{R}^{T \times H \times W \times C} \) of \( T \) sampled frames, each of spatial resolution \( H \times W \) with \( C \) channels, we feed it into the image encoder of CLIP \( \Phi^{\text{vis}}(\cdot) \), obtaining frame-wise visual encodings:

\[ \Omega(v_i) = [\omega_1, \omega_2, \ldots \omega_T] = \Phi^{\text{vis}}(v_i), \]  

(3)

where \( v_i \in V_{\text{seen}} \) and \( \omega_i \in \mathbb{R}^d \). Furthermore, to generate text encodings for a class, the text encoder of CLIP \( \Phi^{\text{text}}(\cdot) \) needs a prompt template [36] as input (further explained in Sec. III-D1). These constitute the set of semantic embeddings:

\[ \Psi = \{ \psi(l_k) = \Phi^{\text{text}}(\text{prompt}(l_k)) | l_k \in \mathcal{S} \cup \mathcal{U} \} \]  

(4)

The proposed framework has two main components. The first is a visual branch for context-aware temporal modeling of the frame-wise video encodings \( \Omega(v_i) \) using LoCATe, which captures multi-scale local spatial context and aggregates them from the temporal dimension. The second is a semantic branch for modeling action relationships using GAT [19] that models a node’s relationships with its neighbors via a multi-head attention mechanism and learns seen-unseen associations. In the following sections, we provide an in-depth discussion of the two components and how they contribute to visual-semantic alignment.

C. Visual branch: Context-aware temporal modeling

Attention mechanisms have proven to be useful in image classification for extracting discriminative information [40] and form the core of transformer-based models. Recently, some transformer-based models [8], [41] pretrained in a multimodal fashion with visual-language data (I-VL models) have shown promising zero-shot generalizability in areas like image classification. However, videos pose a bigger challenge. Firstly, unlike images, large-scale labeled video-text pairs for pretraining I-VL models are hard to collect and can incur enormous computation and memory costs. And secondly, the semantic meaning of a video clip is engraved within multiple individual frames and the spatiotemporal correlations between them. Recently, a few works [10], [11] have successfully adapted these I-VL models to action recognition by focusing on temporal modeling via transformers. Such transformer encoders usually consist of multi-head self-attention (MHSA), layer normalization (LN), and MLPs. But the inputs to them are meticulously obtained frame-level visual encodings from a previous stage of the framework, and hence these transformers just act as an aggregator of frame-level encodings. Moreover, their capabilities are limited when it comes to capturing local spatial context, which could be detrimental to ZSAR. Object-centric and environmental contexts can influence both distinguishability and functional similarity between actions (Fig. 1), eventually assisting in knowledge transfer from seen to unseen actions.

In our visual branch, after extracting frame-level visual encodings for a video \( v_i \), we pass them through our Local Context-Aggregating Temporal Transformer (LoCATe) that performs self-attention operations. Moreover, instead of using an MLP like traditional transformers, we propose a Local Context Aggregator (LCA) block to leverage multi-scale local context from the frame-level encodings. For LoCATe, we first prepare our input embeddings as:

\[ x_i = \Omega(v_i) + \rho_{\text{temp}} \]  

(5)

where \( \rho_{\text{temp}} \) denotes temporal positional encodings, which are learnable vectors. These embeddings pass through LayerNorms and MHSA:

\[ \hat{x}_i = x_i + \text{MHSA}(\text{LN}(x_i)) \]  

\[ \hat{x}_i = \text{LN}(\hat{x}_i) \]  

(6)

Having accounted for global spatiotemporal attention via MHSA within the tokens \( \hat{x}_i \), we reshape them to 2D feature maps and pass them through an LCA block that has three separate convolutional branches (Fig. 4). Each branch consists of a 1 \( \times \) 1 convolutional (CONV) layer, followed by a 3 \( \times \) 3 dilated CONV layer. Different dilation factors in different branches allow us to tune the receptive field sizes, capturing local context from multiple scales. The output feature maps are refined via a Convolutional Block Attention Module (CBAM [14]), which consists of spatial and channel attention blocks. From a frame’s perspective, channel attention discovers the contribution of a feature map during learning, while spatial attention looks for the essential signals to learn from that feature map, hence refining the output features as a whole.
We then use another $1 \times 1$ CONV to obtain a desired number of feature maps and pass them through a global average pooling (GAP) layer for summarizing the resulting spatial information. We ensure that the output from each branch, when concatenated, yields $d$-dimensional embeddings such that they can be passed on to the next encoder layer. Averaging when concatenated, yields information. We ensure that the output from each branch, different actions.

D. Semantic branch: Modeling action relationships

Text encodings that represent action class semantics inherently establish a relationship between classes. However, we construct a knowledge graph (KG) and use a Graph Attention Network (GAT) to emphasize the associative strength between different actions.

1) Constructing the KG: Instead of using off-the-shelf semantic networks like ConceptNet to build KGs [5] or relying on skeleton-based KGs [13], we take a more scalable approach. Inspired by Ghosh et al. [13], we build a KG with one node for each action class $l_k$ of a dataset, where $l_k \in \mathcal{L}$. Additionally, in the KG of every dataset, we add nodes corresponding to classes in Kinetics-400 [42] since previous works [13], [22] have shown advantages of augmenting classes from other large-scale datasets in the KG. To initialize the node features, previous graph-based approaches have mostly used word embeddings. For example, [5] uses Word2Vec [16], [15] uses GloVe [17], and [13] uses Sent2Vec [18] embeddings. On the contrary, we use one of the standard handcrafted prompt templates [8] for an action class $l_k$ – “A video of a person [l_k]” – obtaining semantic embeddings $\psi(l_k)$ from CLIP and normalizing them. Initializing the action nodes with these embeddings provides a two-fold advantage. Firstly, there is no need to average word vectors for every word to get embeddings for multi-word action class names like “Apply eye makeup”. Ghosh et al. [13] have previously illustrated that the averaging approach [5], [12], [43] fails to capture the correct relationships between actions. And secondly, since graph-based approaches rely heavily on the input graph and its node representations, using $\psi(l_k)$ instead of other embeddings like Sent2Vec [6], [13] proves to be complementary to the video embeddings from LoCATe, as we show in the ablations.

Edge formations are done following KG1 [13], where node $i$ is connected to $j$ if $j$ is among the top $N_G$ closest neighbors of $i$, as per the cosine similarity of their node features:

$$\Theta(\psi(l_i), \psi(l_j)) = \frac{\psi(l_i) \cdot \psi(l_j)}{||\psi(l_i)||_2 \cdot ||\psi(l_j)||_2} \tag{7}$$

We also make the resulting adjacency matrix $A$ symmetric and add self-loops for every node.

2) Learning semantic associations: Instead of summing the neighborhood features like GCN [20] or simply averaging over them, GAT [19] employs multi-head attention to aggregate the dynamic relationships of a node with its neighborhood. To compute the node embedding $h_i^{(l+1)}$ of layer $(l + 1)$ from the embeddings of layer $l$, GAT computes unnormalized pair-wise attention coefficients $c_{ij}^{(l)}$ using the transformed embeddings $z_i^{(l)}$ of adjacent nodes $i$ and $j$:

$$z_i^{(l)} = W^{(l)} \cdot h_i^{(l)}$$

$$c_{ij}^{(l)} = \text{LeakyReLU}(\delta^{(l)}(z_i^{(l)} \parallel z_j^{(l)})) \tag{8}$$

Here, $W^{(l)}$ is a learnable weight matrix, $\parallel$ is a concatenation operation, and $\delta^{(l)}$ is a learnable weight vector. For a neighborhood $\mathcal{N}_i$ of node $i$, softmax is applied to normalize the attention coefficients:

$$\alpha_{ij}^{(l)} = \frac{\exp(c_{ij}^{(l)})}{\sum_{k \in \mathcal{N}_i} \exp(c_{ik}^{(l)})} \tag{9}$$

Finally, the outputs are aggregated from $M$ different attention heads:

$$h_i^{(l+1)} = \left\| \sigma \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l,m)} \cdot z_j^{(l,m)} \right) \right\|_1 \tag{10}$$

where $\alpha_{ij}^{(l,m)}$ and $z_j^{(l,m)}$ are the attention coefficients and transformed embeddings from the $m^{th}$ head and $\sigma$ is a non-linear activation function. The output node features $h_i^{out}$ are mapped to $d$-dimensions.
E. Model training and inference

In our two-stage training process, we first train the transformer LoCaTe with seen data. Once trained, we map the \( v_i \in \mathbb{R}^d \) to \( |S| \) outputs and use it as a classifier with weight matrix \( W_{LoCAte} \in \mathbb{R}^{d \times |S|} \). Next, we train the GAT and take the output node features corresponding to seen classes \( h_{j}^{out} \in \mathbb{R}^{d \times |S|} \). Visual-semantic alignment is learned by optimizing a mean squared error loss:

\[
    L_{MSE} = ||W_{LoCAte} - h_{j}^{out}||_2^2
\]

At test time, a batch of \( B \) video samples are passed through LoCaTe to obtain their video embeddings \( v_i^{out} \in \mathbb{R}^{B \times d} \). Node features corresponding to the unseen classes \( h_{j}^{out} \in \mathbb{R}^{d \times |U|} \) \((j \in U)\) are extracted from the trained GAT, and a cosine similarity is utilized to find the similarity between a video embedding \( v_i^{out} \) and the semantic embeddings:

\[
    \Theta(v_i^{out}, h_j^{out}) = \frac{v_i^{out} \cdot h_j^{out}}{||v_i^{out}||_2 \cdot ||h_j^{out}||_2}
\]

The most similar class is assigned as the test label for a given video \( v_i \). In the case of generalized ZSAR, node features corresponding to a node \( j \in S \cup U \) are considered for computing similarity scores.

IV. EXPERIMENTS

In this section, we conduct experiments on two benchmark datasets in two different settings – conventional ZSAR (CZSAR) and generalized ZSAR (GZSAR). Furthermore, several ablation studies illustrate our framework’s efficacy.

A. Datasets and evaluation protocols

There are two datasets predominantly used by previous works for experiments in ZSAR – UCF101 [49] and HMDB51 [50]. They consist of 13,320 and 6,766 videos from 101 and 51 classes, respectively. Unlike zero-shot image classification [2], disjoint seen-unseen splits for these datasets in ZSAR have not been standardized yet. The most widely used ones are several 50/50 random splits of the dataset classes into seen and unseen since it becomes possible to obtain large enough sets for both learning and evaluation. We follow [30] and use the 50/50 splits for a fair comparison with a wide array of existing works. Hence, we obtain data from 51 and 26 classes for training and 50 and 25 classes for CZSAR testing with UCF101 and HMDB51 datasets, respectively.

While evaluating our method in the CZSAR setting, we report the top-1 accuracy for the unseen classes. In the GZSAR setting, we compute average top-1 accuracy for seen classes \( S_{acc} \) and unseen classes \( U_{acc} \) and report the harmonic mean \( \frac{1}{2} [S_{acc} + U_{acc}] \) of these two as the primary evaluation metric.

Recently, a new evaluation protocol named TruZe [51] has emerged that redefines the sets of seen and unseen classes for UCF101 and HMDB51. It ensures a fair comparison among models by avoiding any unseen class that overlaps with the Kinetics-400 dataset commonly used for pre-training, thus maintaining the ZSL premise. TruZe provides a new set of 70/29 seen and 31/22 unseen classes for the UCF101/HMDB51 datasets.

B. Implementation details

All experiments are performed on a single NVIDIA A100 80GB GPU based on a PyTorch framework.

1) Visual branch: Our frame sampling strategy follows [7], [26], wherein we prepare \( T = 16 \) frames for a video clip. These frames are processed to have a spatial dimension of \( 224 \times 224 \). Following the observation of ActionCLIP [36] that ViT-B/16 provides the best parameter-accuracy balance, we adopt the same backbone for the image and text encoders of our pretrained CLIP. Throughout the framework, both of these encoders are kept frozen, and the output image and text encodings have a dimension \( d = 512 \). The transformer LoCaTe

<table>
<thead>
<tr>
<th>Method</th>
<th>UCF101</th>
<th>HMDB51</th>
</tr>
</thead>
<tbody>
<tr>
<td>SJF [44]</td>
<td>8.9 ± 2.2</td>
<td>10.5 ± 2.4</td>
</tr>
<tr>
<td>ASR [29]</td>
<td>27.5 ± 1.9</td>
<td>10.7 ± 1.5</td>
</tr>
<tr>
<td>GA [33]</td>
<td>17.5 ± 2.2</td>
<td>20.1 ± 2.1</td>
</tr>
<tr>
<td>CI-GNN† [37]</td>
<td>29.7 ± 4.2</td>
<td>18.9 ± 3.5</td>
</tr>
<tr>
<td>OZA [43]</td>
<td>30.3</td>
<td>N/A</td>
</tr>
<tr>
<td>TS-GCN [5]</td>
<td>33.4 ± 3.4</td>
<td>21.9 ± 3.7</td>
</tr>
<tr>
<td>PS-GNN† [38]</td>
<td>35.1 ± 4.6</td>
<td>24.2 ± 3.3</td>
</tr>
<tr>
<td>OD [35]</td>
<td>37.3 ± 2.1</td>
<td>36.1 ± 2.2</td>
</tr>
<tr>
<td>ADPE† [23]</td>
<td>22.5</td>
<td>21.7</td>
</tr>
<tr>
<td>VC-GCN† [12]</td>
<td>35.6 ± 2.1</td>
<td>32.5 ± 2.5</td>
</tr>
<tr>
<td>GZV [4]</td>
<td>37.5 ± 2.3</td>
<td>N/A</td>
</tr>
<tr>
<td>TSE† [39]</td>
<td>51.5 ± 4.1</td>
<td>30.8 ± 7.1</td>
</tr>
<tr>
<td>TSA† [39]</td>
<td>51.8 ± 4.4</td>
<td>37.1 ± 10.8</td>
</tr>
</tbody>
</table>

LoCaTe-GAT (Ours) | 60.4 ± 1.3 | 33.7 ± 1.6 |
is initialized with a Gaussian distribution $\mathcal{N}(0, 0.02)$, where the positional encoding $\rho_{\text{temp}} \in \mathbb{R}^{T \times d}$. The MHSA block has 8 attention heads, and no dropout layer has been used inside the block. An LCA block within the transformer encoder contains three branches $b_1, b_2$, and $b_3$ for capturing multi-scale context. Each branch $b_i$ has a $3 \times 3$ dilated convolution layer with dilation factor $d_i$ (values 1, 2, 4 for $i = 1, 2, 3$ respectively). LoCAte is trained using an Adam optimizer with a learning rate of $2 \times 10^{-7}$ and a batch size of 22 videos. Following previous work [7], [26], we use one clip during training and 25 clips during testing per video. Our best results are achieved using one transformer encoder per block. For UCF101 and HMDB51, our transformer is trained for 16 and 28 epochs, respectively.

2) **Semantic branch**: During KG construction, the number of nodes equals $|S| + |U| + |K_{400}|$, where $K_{400}$ is the set of classes from Kinetics-400 [42]. Following previous works [6], [10], we also reaped the benefits of using these additional nodes for capturing richer class relationships but do not utilize them while learning visual-semantic alignment. After being initialized by $d$-dimensional text encodings from CLIP, edges are formed between these nodes with $N_{G_i} = 5$. We apply a three-layer GAT model [19], with the first two layers having 4 and the last layer having 6 attention heads. We train it with an Adam SGD optimizer having a learning rate of $10^{-3}$ for 100000 epochs. To prevent overfitting, we use dropout layers with a rate of 0.6.

C. Zero-shot results

1) **Conventional ZSAR**: We compare our method with an extensive array of previous works. Our focus is only on the inductive setting due to its compliance with the zero-shot criteria of not having unseen class samples during training. Table I shows that our proposed method, LoCAte-GAT, consistently improves over the state-of-the-art (SOTA) across both benchmarks. We achieve absolute gains of 2.8% and 2.3% on UCF101 and HMDB51 over the SOTA, X-FLO [11].

2) **Generalized ZSAR**: Being the realistic but more difficult setting in zero-shot literature, only a few methods present their model’s efficacy in the GZSAR setting. A few methods [12], [23], [37], [39] attempt to enhance model generalizability using a transductive approach (training with a few unseen class samples as well). In Table II, we show that our (inductive) method beats these transductive methods too. On UCF101, we beat the closest method [39] considerably by 8.6%. On HMDB51, our results are comparable to the state-of-the-art, only behind OD [35] and the transductive method TSA [39] by 3.4%. One reason behind this lesser performance could be that many actions in HMDB51, like sit, walk, and laugh, are insensitive to object-centric or environmental context, the fundamental motivation behind our novelty.

3) **Evaluation on TruZe protocol**: The recent work TruZe [51] claims that their newly proposed seen-unseen splits are fairer but more restrictive. They justified this by comparing the top-1 accuracy for a wide range of ZSAR models using random 50/50 splits and TruZe splits and found that all models perform relatively poorly with TruZe splits. Table III presents the CZSAR performance of several models as reported by [31], [51]. Despite harder splits, LoCAte single-handedly beats the SOTA [31] by a whopping absolute gain of 16.7% on UCF101. Adding the GAT further increases accuracy by 1.9%. On HMDB51, LoCAte-GAT achieves an absolute gain of 5.8% over the current best method E2E [26].

D. Ablation studies

For a uniform evaluation, we perform our studies on the UCF101 dataset due to its diversity in terms of action classes. Moreover, there are large variations in camera motion, object appearance, pose, object scales, and viewpoint that are necessary to establish the robustness of our model components.

1) **Effect of visual and semantic branches**: Temporal modeling is especially essential for video understanding, as shown by previous works on ZSAR [10], [11]. However, by capturing object-centric and environmental contexts, we beat the SOTA method. We first evaluate our framework having only the visual branch (i.e., without GAT). In this case, similarity scores are computed at test time between the video encodings given by CLIP alone could not provide. We also achieve good inference speed (0.54 videos per second). Our model is lightweight with 5.85 million parameters – significantly lower than others like ActionCLIP [36] (141.7 million).

2) **Improvements by capturing multi-scale context**: To analyze the benefits of capturing local context from multiple scales, we consider reducing the number of branches in the LCA block (Fig. 4) to 2 and 1 (from the default 3). We first remove one branch with dilation factor $d = 1$ and then remove two branches with $d = 1$ and $d = 2$ to observe the effect of neglecting contextual cues from smaller scales. The decreasing accuracy with decreasing branches in Table V

<table>
<thead>
<tr>
<th>Method</th>
<th>UCF101</th>
<th>HMDB51</th>
</tr>
</thead>
<tbody>
<tr>
<td>LATEM [52]</td>
<td>15.5</td>
<td>9.1</td>
</tr>
<tr>
<td>SYNC [53]</td>
<td>15.3</td>
<td>11.6</td>
</tr>
<tr>
<td>BI-DIR [45]</td>
<td>15.7</td>
<td>10.5</td>
</tr>
<tr>
<td>OD [35]</td>
<td>22.9</td>
<td>21.7</td>
</tr>
<tr>
<td>E2E [26]</td>
<td>45.5</td>
<td>31.5</td>
</tr>
<tr>
<td>ER [30]</td>
<td>51.1</td>
<td>N/A</td>
</tr>
<tr>
<td>STSCR [31]</td>
<td>57.2</td>
<td>N/A</td>
</tr>
<tr>
<td>LoCAte (Ours)</td>
<td>73.9</td>
<td>37.1</td>
</tr>
<tr>
<td>LoCAte-GAT (Ours)</td>
<td>75.8</td>
<td>37.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component</th>
<th>Top-1</th>
<th>Parameters</th>
<th>Test time</th>
</tr>
</thead>
<tbody>
<tr>
<td>With LoCAte</td>
<td>75.6 ± 4.1</td>
<td>1.13M</td>
<td>0.57 V/s</td>
</tr>
<tr>
<td>With LoCAte-GAT</td>
<td>76.0 ± 2.7</td>
<td>5.85M</td>
<td>0.54 V/s</td>
</tr>
</tbody>
</table>
Table V
Impact of different sub-components in LoCATe on top-1 accuracy (CZSAR setting on UCF-101). d denotes the dilation factor.

<table>
<thead>
<tr>
<th>Component</th>
<th>Inclusion</th>
<th>Top-1 (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of LCA branches</td>
<td>4 (d=1, 2, 4)</td>
<td>76.0</td>
</tr>
<tr>
<td></td>
<td>2 (d=2, 4)</td>
<td>75.2</td>
</tr>
<tr>
<td></td>
<td>1 (d=4)</td>
<td>74.9</td>
</tr>
<tr>
<td>CBAM</td>
<td>Yes</td>
<td>76.0</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>75.4</td>
</tr>
</tbody>
</table>

Fig. 6. Benefit of capturing local context. The small objects of interest are enclosed in yellow boxes. Action classes are taken from UCF-101.

conveys that capturing context at smaller scales is beneficial. For a class-level comparison, we randomly pick four actions involving small objects of interest and plot the top-1 accuracy with different architectures in Fig. 6. Even without GAT, a 3-branch LoCATe easily outperforms recent works like AURL [48] on most occasions. With a 3-branch LoCATe-GAT, state-of-the-art results are achieved in all cases. However, a LoCATe-GAT framework with fewer branches fares below a 3-branch LoCATe, indicating that only enhancing the semantic representations of actions is inadequate for ZSAR; multi-scale local context for empowering the visual embeddings is essential as well.

3) Role of CBAM in LCA: Table V shows that the exclusion of CBAM modules from the LCA block yields a lesser accuracy, suggesting that feature refinement using channel and spatial attention helps produce better video embeddings. Moreover, while comparing in a 3-branch LCA setting (Fig. 6), excluding CBAM consistently shows an inferior performance to both LoCATe and LoCATe-GAT frameworks with CBAM, proving its worth within the framework.

4) Different text representations: We compare the semantic representations learned by GAT when initialized with word embeddings (Word2Vec [16]), sentence embeddings (Sent2Vec [18]), and CLIP text encodings [8] in Table VI. It is evident that CLIP text encodings are more powerful than word embeddings. Moreover, they are better as node initializers for our knowledge graph since we beat LGKT [6] comprehensively (Table I) on CZSAR, who use Sent2Vec [18].

In fact, Sent2Vec worsens the visual-semantic alignment with video embeddings obtained from LoCATe, yielding an accuracy of 30.1%. This is similar to a previous Sent2Vec-based framework [13] (32.8%) but is heavily outperformed using our framework based on CLIP text encodings by 45.9%. Hence, learning from CLIP text encodings clearly has superior compatibility with our video embeddings.

5) Significance of LCA over MLP: Traditionally, transformer encoder blocks employ MLPs as feed-forward networks (FFNs) that constitute about two-thirds of the total trainable parameters. These FFNs are position-wise functions, and [54] recently showed that they emulate neural memory and act as pattern detectors over an input sequence of textual data. However, with video frames (images) where spatial relationships are crucial, convolutional layers can play a similar role due to their advantageous compatibility with image data by design. Experimentally, we justify the merit of our LCA block (that has convolutional layers) in the transformer LoCATe as opposed to the traditional MLP from two aspects – classification accuracy and number of trainable parameters (Fig. 7). Using only LoCATe as our ZSAR framework, LCA achieves an absolute gain of 4% over MLP, with just about one-third of the number of trainable parameters than the MLP setting. With LoCATe-GAT as the ZSAR framework, LCA again overtakes MLP by 4.5% by using just about 1.5 times less trainable parameters. Hence, using LCA makes the framework lightweight while improving accuracy.

6) Mitigating polysemy: Figure 8 presents the top-1 accuracy achieved on four consecutive pairs of action classes that encounter polysemy due to the words playing, shot, jump, and diving. Recent works like AURL [48] rely simply on word embeddings for semantic descriptions and are unable to mitigate polysemy, yielding poor results for such classes. However, LoCATe produces much more distinguishable visual representations despite polysemy, owing to the powerful CLIP visual embeddings as inputs and accounting for action context. Using...
a GAT additionally induces even greater distinguishability in the semantic space, handling polysemy better and yielding massive improvements over recent methods like AURL.

7) Is GAT necessary for ZSAR?: The node aggregation strategy of a GAT aids zero-shot knowledge transfer between classes. This is visualized in Fig. 5, where neighbors are connected to a node with different statistical strengths (attention) after training the GAT, as shown by different edge colors. Additionally, semantically-similar neighbors highlight a meaningful construction of our knowledge graph. Figs. 5(a) and 5(b) show similar sports/musical instruments as neighbors of hammer throw/playing dhol. Considering hammer throw, GAT learns to give more attention to throw discus due to similar spatiotemporal motion of the body in these sports, followed by javelin throw (due to object shape similarities). Dhol and tabla are both two-surface percussions and hence receive more attention, unlike drum kits. Fig. 5(c) presents the interesting case of high jump, where polysemy of the verb “jump” can be seen among the connected neighbors. However, long jump and balance beam are given lesser attention than sky diving and trampoline jumping, both of which involve greater heights like high jump. Finally, transcending objects, environments, and even body motions, writing on board and typing (Fig. 5(d)) are strongly connected.

V. CONCLUSION

Motivated by the dual role of object-centric and environmental context for zero-shot action recognition, we present a novel framework for improving the visual-semantic alignment of action classes. It consists of a temporal transformer that focuses on the multi-scale local context during temporal modeling, as well as a graph attention network for capturing action relationships, empowered by the powerful image and text encodings from a pretrained CLIP. Extensive experiments on two challenging benchmarks demonstrate state-of-the-art results in conventional and generalized zero-shot settings.

REFERENCES

M. Jain, J. C. Van Gemert, T. Mensink, and C. G. Snoek, “Ob-
J. Gao and C. Xu, “Ci-gnn: Building a category-instance graph for zero-


