# Compositional Video Understanding with Spatiotemporal Structure-based Transformers

Hoyeoung Yun[1]*, Jinwoo Ahn[2]*, Minseo Kim[1], Eun-Sol Kim[1, 2]†
[1]Department of Computer Science, Hanyang University
[2]Department of Artificial Intelligence Application, Hanyang University
{yhy17520, jinwooahn, simon1011, eunsolkim}@hanyang.ac.kr

## Abstract

*In this paper, we suggest a new novel method to understand complex semantic structures through long video inputs. Conventional methods for understanding videos have been focused on short-term clips, and trained to get visual representations for the short clips using convolutional neural networks or transformer architectures. However, most real-world videos are composed of long videos ranging from minutes to hours, therefore, it essentially brings limitations to understanding the overall semantic structures of the long videos by dividing them into small clips and learning the representations of them. We suggest a new algorithm to learn the multi-granular semantic structures of videos, by defining spatiotemporal high-order relationships among object-based representations as semantic units. The proposed method includes a new transformer architecture capable of learning spatiotemporal graphs, and a compositional learning method to learn disentangled features for each semantic unit. Using the suggested method, we resolve the challenging video task, which is compositional generalization understanding of unseen videos. In experiments, we demonstrate new state-of-the-art performances for two challenging video datasets.*

## 1. Introduction

Recently, as video-content-based services have obviously proliferated, human-level video understanding using artificial intelligence has been regarded as one of the fundamental problems in the computer vision field. However previous methods of video understanding [15, 31, 32] have been focused on short-term clips ranging from seconds to a few minutes, and, it is considered one of the challenging problems to understand the semantic structures in long video due to the complex and high-order dependencies between scenes or events throughout the spatial and temporal axis.

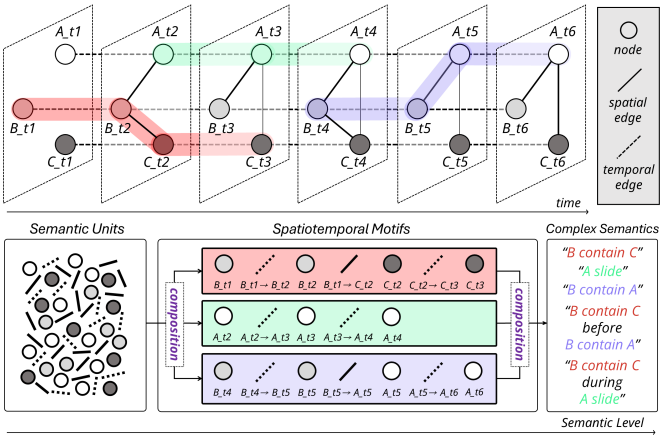



Figure 1. Overall scheme of proposed compositional learning strategy. We introduce an object-centric spatiotemporal graph as an alternative representation of the given video and decompose it to obtain fine-grained semantic units. Subsequently, by composing these semantic units, we can reproduce higher-level semantics corresponding to the complex semantics present in the given video.

Due to the complexity, most conventional video understanding methods [15, 31, 32] represent each frame (or uniformly segmented short-term clips) as a real-valued vector which makes it hard to consider high-order relationships between objects within the frame. Moreover, the methods have limitations in modeling various lengths of actions inherent within videos and the complex relationships between the actions as the length of the video increases. In other words, it is hard to understand multi-granular semantic structures encompassing objects, scenes, and video-wide contexts with conventional video understanding methods. To tackle this problem, we suggest a novel video understanding method based on object-oriented representations that can learn the spatiotemporal semantic structures of the videos in compositional ways.

The two main goals of this paper are 1) to propose a

*These authors contributed equally to this work
†Corresponding author

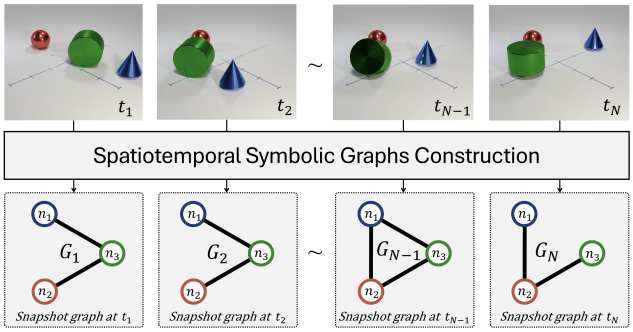



Figure 2. To represent the semantic structure for a given video, we construct a symbolic graph with $N$ number of graphs corresponding to $N$ frames, and each graph, denoted as $G_t$, encodes information about $n$ number of objects as nodes and the relationship between two nodes as edges at time $t$.

new transformer-based algorithm to learn object-oriented semantically disentangled representation of videos and, 2) to tackle the most challenging video-related task, which is a compositional generalization task using the proposed method. Firstly, we introduce a new object-oriented video understanding algorithm consisting of four modules: spatiotemporal graph construction, spatiotemporal graph transformer, object-oriented video encoder, and compositional learning modules. The proposed algorithm demonstrated outstanding performance on the CATER[10] and MOMA-LRG[23] datasets, which can be considered the most challenging datasets in terms of the length of the video and the complexity of the elements constituting the actions in the videos. Furthermore, we argue that compositional generalization ability is a crucial element of video understanding algorithms to understand long videos and suggest a new dataset to tackle this problem. Based on the CATER videos, we present new data splits for comparing the compositional generalization abilities between models.

Through experiments, we demonstrate that the proposed algorithm achieves high video understanding performance in both CATER (synthetic data) and MOMA-LRG (real-world data) datasets. Specifically, we show that our proposed method achieves significantly higher performance in compositional generalization tasks compared to existing comparative methods using object-oriented video representation.

Our contribution can be summarized as follows.

- A novel object-oriented video understanding method to learn the multi-granular semantic structure of long videos is suggested.
- A novel data split for compositional generalization test of video understanding algorithms is proposed.
- In experiments with both synthetic and real-world videos, we achieve new state-of-the-art performances.

## 2. Related Work

### 2.1. Video Representation Learning

Video representation learning is the most fundamental task for various video-related tasks such as action recognition [4, 15, 36], video retrieval [8, 30], and video captioning [19, 41], which aims to learn meaningful representation from raw video data. While image representations could only consider spatial visual features, understanding temporal dependencies between image frames is crucial for video representation learning. To tackle this problem, previous research such as STIP [20], HOG3D [17], and iDT [35] have tried to design hand-crafted descriptors to capture spatiotemporal patterns from the videos.

Along with the remarkable achievements of convolutional neural networks (CNNs), video representation learning methods based on CNN have been proposed. Two-stream CNNs [7, 28] suggested an additional temporal stream that uses dense optical flow between consecutive video frames to exploit the motions of videos. I3D [4] improved existing two-stream CNNs [28] by inflating 2D CNNs into 3D [31]. Moreover, the non-local block was proposed to help models understand long-range temporal dependencies [37].

Recently, transformers [33], which are powerful in learning long-range temporal dependencies, have been applied because the ConvNet-based models do not have good temporal dependencies on long-term videos [1, 6]. However, existing methods based on CNNs or transformers focus on learning temporal dependencies between image sequences, so complex relationships between objects in a spatiotemporal manner are hard to contain.

### 2.2. Compositional Video Learning

Compositional generalization ability, which refers to the capability for understanding unseen novel data composed of concepts or components learned during training, can play an important role in long video understanding. AVT [9], which is a model for action anticipation, forecasts the next action based on understanding the previous part of the video. Video synthesis models like AG2Vid [3] predict the following circumstance and synthesize the next frame composing concepts and components shown. In the video captioning task, DCC [13] shows compositionality explaining novel objects without paired data, which do not exist in caption corpora. VISA [22] suggests a novel task called compositional temporal grounding to assess the compositional generalizability in temporal grounding task with testing on the queries of new combinations of seen words during training.

Recently, a number of benchmark datasets such as CLEVRER [40], TVQA [21], and AGQA [11] have been suggested for compositional video question answering tasks. In this paper, the most challenging dataset CATER

[10] in terms of compositional action recognition for long videos is used for the experiments.

### 2.3. Transformers for Graphs

As transformers[33] have achieved remarkable performance on a wide range of domains, various application methods for graphs have emerged[14, 42].

Earlier transformer methods for graphs applied self-attention only for locally close neighbors[5, 34] or with additional message-passing modules[27]. However, these approaches have limitations in representing the overall graph structure and are prone to over-smoothing. Consequently, self-attention on nodes has been applied to handle edges and integrate graph structures by incorporating heuristic adaptation methods[18, 24]. Especially, TokenGT[16] maintains the self-attention mechanism of pure transformer[33] without modifications and furnishes both nodes and edges with specific token-wise embeddings as input.

Although there has been extensive exploration of transformer architectures designed to process graph inputs recently, there is a notable absence of discussions on algorithms specifically designed for learning temporal graphs.

## 3. Method

The core idea of the proposed algorithm is to learn representations of videos at the object level, defining spatiotemporal high-order relationships among the object-level representations as semantic units. The suggested method consists of the following four modules.

### 3.1. Constructing Spatiotemporal Symbolc Graphs

In this paper, we define a spatiotemporal symbolic graph as an input representation of a given video. A spatiotemporal graph $\mathcal{G} = \{G_t\}_{t=1,\dots,N}$ consists of a sequence of snapshot graphs, and each snapshot graph $G_t = (A_t, X_t, E_t)$ denotes the attributed graph where $A_t \in \mathbb{R}^{n\times n}$ represents the symmetric adjacency matrix with $n$ nodes, $X_t \in \mathbb{R}^{n\times p}$ is the attribute matrix of $p$ attributes per node, $E_t \in \mathbb{R}^{n\times n\times q}$ is the attribute tensor of $q$ attributes per edge. In that, as described in Figure 2, a graph for a video with $N$ frames consists of $N$ number of snapshot graphs, and each snapshot graph $G_t$ encodes information about $n$ number of objects by considering the attributes such as color, shape, and material using the $X_t$. The information about the relationship, such as distance and direction between nodes $n_i, n_j$, is encoded in $E_{i,j}$. The process of constructing the graph $\mathcal{G}$ from a video can be easily implemented using well-established methods such as object detection and attribute classification, at least for CATER videos.

### 3.2. Spatiotemporal Graph Transformer(ST-GT)

As discussed in Section 2, while transformer architectures that can take graphs as input have been widely proposed recently, there is a significant lack of discussion on algorithms capable of learning temporal graphs. In this paper, we propose a novel transformer algorithm that takes a temporal graph given as a sequence of static graphs as input, enabling the learning of spatiotemporal correlations among nodes.

The suggested method, Spatiotemporal Graph Transformer(ST-GT), introduces a total adjacency matrix $A^{total} \in \mathbb{R}^{(N\times n)\times(N\times n)}$ which involves temporal auxiliary edges describing the temporal connection between two adjacent static graphs. In other words, it involves additional temporal edges between nodes representing the same object in adjacent static graphs. This allows the representation not only of relationships between objects at a single time point but also the temporal changes of objects across connected frames. By introducing temporal auxiliary edges, it is anticipated that the attention mechanism of the transformer can learn relationships between objects in adjacent frames.

The total adjacency matrix $A^{total}$ with temporal auxiliary edges can be defined as follows.

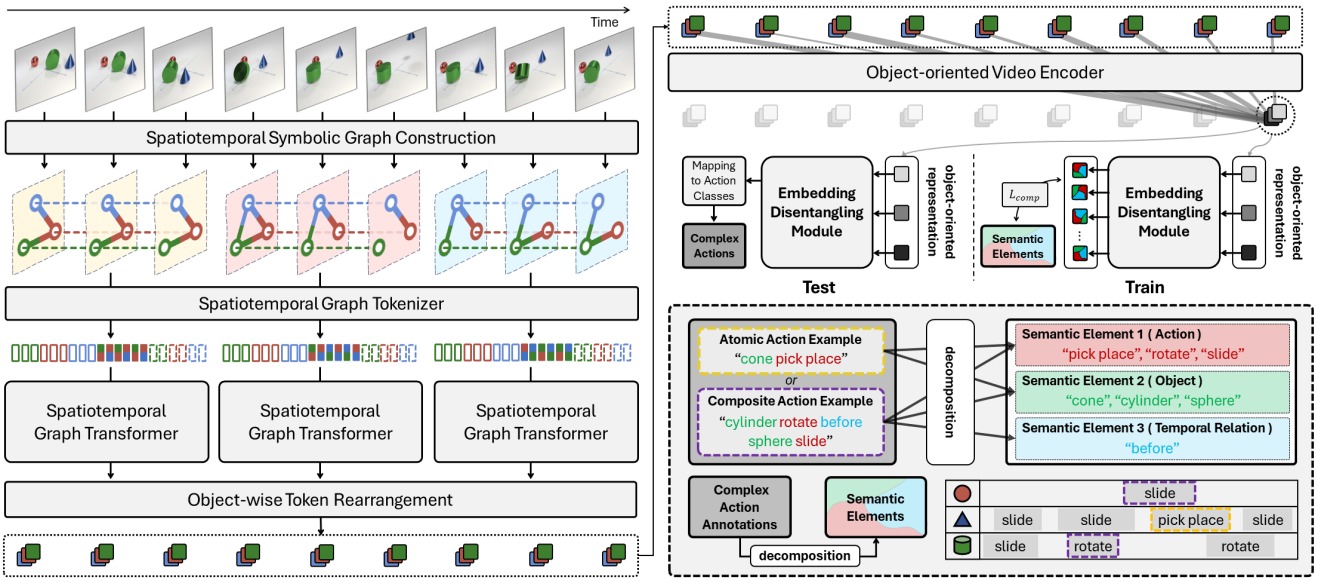
$$A^{total}_{i,j} = \begin{cases} A_t, & \text{for diagonal blocks} \\ 1, & \text{for (i, n+i)} \\ 1, & \text{for (n+j, j)} \end{cases} \tag{1}$$

It can be seen as a static graph consisting of $N$ subgraphs, the temporal relationship between adjacent frames is encoded in the off-diagonal blocks.

Inspired by recent work [16], all elements in the $\mathcal{G}$ with $A^{total}$, which are nodes, spatial edges, and temporal auxiliary edges, are tokenized to be fed into the transformer as inputs. If the edge is defined as an input token, information about the connection between tokens might be lost. To maintain this information, position embeddings representing spatial and temporal relationships are defined and added to the input tokens. The elements composing the input tokens of the suggested ST-GT can be summarized as follows.

The input $\mathbf{X}$ of ST-GT consists of three types of tokens, which are node, spatial edge, and temporal auxiliary edge. Each token contains three kinds of information: feature, position embeddings, and type identifier. The position embeddings represent the structural information of each node or edge in the total adjacency matrix $A^{total}$. The type identifier is introduced to indicate which of the three components of the spatiotemporal graph each token represents.

**Node-type Token** In addition to the attribute feature defined in Section 3.1, information about time and object's pose is added. The feature vector of the node-type token is represented with $d_f$-dimensional vectors. To consider the structural information of each node, the $d_p$-dimensional eigenvector of graph Laplacian of $A^{total}$ is considered as the positional embedding vector. Finally, $d_t$-dimensional learnable embedding is added to indicate the type of tokens.




(a) (b)

Figure 3. (a) Overall architecture of our model. First, through the graph construction and graph tokenizing process, we obtain fine-grained semantic units within the raw video. These tokens are passed to ST-GT, which captures spatiotemporal motifs. Next, we rearrange the output tokens of ST-GT by object and learn higher-order semantics through an object-oriented video encoder. In the final stage of the embedding disentangling process, the complex semantic structure information contained in each object-oriented representation is disentangled to multiple semantic elements. (b) Details of the process for obtaining semantic elements. (The dotted box in the bottom right corner) The CATER dataset defines two types of complex actions: atomic actions (yellow dotted box) and composite actions. (purple dotted box) These complex action labels are decomposed into multiple semantic elements(e.g. object, action, temporal relation) and used in the training process.

**Spatial/Temporal Edge-type Token** As the edge connects two nodes, the feature vector for the edge-type token is devised by concatenating two feature vectors of nodes. Also, as the positional embedding, two positional embedding vectors of each node are concatenated. Similar to the node-type token, two $d_t$-dimensional learnable embeddings are devised for spatial and temporal edges respectively.

Overall, three types of features as inputs of ST-GT are constructed: $\mathbf{X}_{node} \in \mathbb{R}^{K\times(d_f+d_p+d_t)}$, $\mathbf{X}_{edge_s} \in \mathbb{R}^{L\times(2\times d_f+2\times d_p+d_t)}$, $\mathbf{X}_{edge_t} \in \mathbb{R}^{M\times(2\times d_f+2\times d_p+d_t)}$, where $K, L, M$ are the number of nodes, spatial edges, and temporal edges included in total graph (connecting $N$ number of static graphs). To get the equal dimension size of three types of feature vectors, a single-layer fully connected network is introduced. Finally, the input of the ST-GT can be defined as $\mathbf{X} \in \mathbb{R}^{(K+L+M)\times d}$.

Furthermore, although the size of the $A_{total}$ matrix itself is large, it is a highly sparse matrix. Since temporal auxiliary edges are defined only between adjacent frames, even when breaking down a long video into multiple segments, it is sufficient to define $A_{total}$ within each segment. This results in a manageable increase in computational complexity.

### 3.3. Object-oriented Video Encoder

We suggest an object-oriented video encoder, which learns object-level action representations by grouping each object's representations obtained in Section 3.2. To observe the movement of objects within the video, output tokens of spatiotemporal graph transformers are grouped based on their object indices in the object-wise token rearrangement module in Figure 3. This allows us to obtain tokens corresponding to each object over time. These tokens are then fed as input to the object-oriented video encoder, which consists of causal masked self-attention, feed-forward layer. Through an object-oriented video encoder, the temporal movement of each object is encoded, resulting in object-specific representations.

### 3.4. Compositional Learning by Disentangling Embeddings

The object-oriented action representation obtained in Section 3.3 contains information on changes in the motion of each object in the video. The information combines two types of information, objects and actions. Rather than directly using the information obtained in Section 3.3 for classification, we define a new learning method that disentangles combined information into subspaces defined by

each semantic element(objects, and actions). This involves predicting the final label(composition of each semantic element) through a combination of predictions within each subspace. This approach is essential to achieve the zero-shot compositional generalization ability targeted in this paper, as it requires understanding newly encountered data (compositional labels) as combinations of previously learned semantic elements. To facilitate this, we introduce a method where predictions are made for each element based on disentangled features for object and action, rather than using features where information about object and action is entangled for label prediction.

To resolve this, entangled feature embeddings are projected onto two independent subspaces(Figure 3). Two learnable embedding layers are introduced for each object and action subspace. By concatenating the disentangled two features, a multi-label classification head is applied to predict labels according to the tasks defined in CATER.

Considering each subspace embedding of objects, actions, and output of object-oriented video encoder, we compute the overall loss of our model $\mathcal{L} = \mathcal{L}_{vid} + \mathcal{L}_{comp}$. Where $\mathcal{L}_{comp} = \mathcal{L}_{obj} + \mathcal{L}_{act} + \mathcal{L}_{TR}$. $\mathcal{L}_{vid}$ is derived from the binary cross entropy of presence action classes in the video using video-level representations. Video-level representations are obtained by mean pooling object-specific representation of object-oriented video encoder. For strong compositional learning, we calculate $\mathcal{L}_{comp}$ by predicting the object, action, and temporal relation of each composite action label with binary cross entropy. Since temporal relation does not exist in task 1, we applied $\mathcal{L}_{TR}$ only at task 2.

# 4. Compositional Generalization Test for Videos

## 4.1. CATER dataset

For experiments, a most challenging dataset CATER [10] is used to check the compositional video understanding ability of the suggested method. CATER dataset[10] consists of 5.5k videos and each video includes 300 frames with 320 by 240 pixels. Through 300 frames, multiple actions with various objects having different durations are included. Specifically, the number of objects appearing in each video ranges from 5 to 10 and each object has four types of attributes such as shape, size, material, and color.

With the CATER dataset, three different tasks are defined. The first task is to predict atomic actions within the trimmed video. There are 14 different atomic actions which are combinations of object shapes(cube, cone, etc.) and action types(rotate, slide, pick-place, and contain). The second task is to predict compositional action labels, which consider a temporal combination of two atomic actions. Based on Allen's temporal algebra, three types of temporal dependency (after, during, and before) are considered to make compositional action labels. The third task is the localization of the position in the last frame of the snitch defined as a special object in CATER. It is a classification problem about which cell in the frame has the snitch, also challenging because the snitch is fixed at a small size and easily contained by other objects, making it hard to see it in a certain range of frames.

## 4.2. A New Split for Compositional Generalization

Based on the CATER dataset, we constructed a new data split to check the compositional generalization ability of the video understanding models. The main idea behind constructing the new split for compositional generalization is that the label comprises combinations of two or more semantic units, where individual semantic units are seen during the training phase, but the combinations of labels observed during the test(or validation) phase are novel. In other words, the aim is to ascertain whether, even though a (compositional) label is seen for the first time, it can be understood by decomposing the label into semantic units and combining the units to comprehend the novel label. Since the CATER dataset provides two-level action labels(tasks 1 and 2), we introduce new data splits to each level for compositional generalization tests as follows.

### 4.2.1 Compositional Generalization Test for Atomic Action Recognition

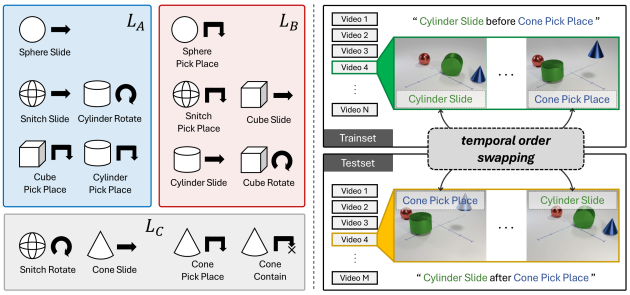
The labels provided by the CATER dataset for Task 1 consist of combinations of object and action types. While there are five and four types of object and action, there are only 14 labels in total due to certain actions not being applicable to specific objects (for example *cone rotate*). We divide these 14 labels into three disjoint label sets $L_A, L_B, L_C$, as illustrated in the following Figure 4 (a), to create splits for the compositional generalization test.

Based on the label splits $L_A, L_B, L_C$, video clips corresponding to $L_A$ and $L_C$ are defined as the training set, while videos of label $L_B$ are defined as the test set. With this training and test split, each object and action component consisting of the test labels is included in the training data, while the specific label combinations of the test set are novel.

Formally, the characteristics of the new data split can be represented as follows. For training and test dataset $\mathcal{D}^1_{train}, \mathcal{D}^1_{test}$,

$$\begin{aligned} O &= \text{SEEN}_O(\mathcal{D}^1_{train}) = \text{SEEN}_O(\mathcal{D}^1_{test}), \\ A &= \text{SEEN}_A(\mathcal{D}^1_{train}) = \text{SEEN}_A(\mathcal{D}^1_{test}). \end{aligned} \tag{2}$$

where, $\text{SEEN}_O(\mathcal{D})$ and $\text{SEEN}_A(\mathcal{D})$ represent the set of object and action types included in dataset $\mathcal{D}$, and $O, A$ represent the set of total object and action types.




(a) (b)

Figure 4. (a) Venn diagram representing the set of 14 labels of the CATER dataset for Task 1. $L_A, L_B, L_C$ refers to the three disjoint sets that we have divided. (b) is an example of a before and after relationship of new data split for Task 2. As shown in the figure, if *cylinder slide before cone pick place* only appears in the training set, the corresponding opposite label *cylinder slide after cone pick place* only appears in the test set.

#### 4.2.2 Compositional Generalization Test for Composite Action Recognition

Task 2 of the CATER dataset considers the temporal relationships between two actions, where each action consists of object and action types. Therefore, we devise another compositional generalization split according to temporal relationships. As discussed in 4.1, three types of temporal relationships (i.e., before, during, after) are proposed according to Allen's temporal algebra As the compositional action labels consist of 14 types, 301 unique temporal action labels are provided as classification labels of Task 2. Similar to the procedure of the first data split, three different disjoint label sets $L_A, L_B, and L_C$ were constructed from the 301 temporal action labels. The detailed procedure to construct three label sets is as follows.

**Before or After relationship** After selecting a randomly selected label *X before Y* and defining it as an element of $L_A$, the opposite label *Y before X* is defined as the element of $L_B$. Then, other labels having the same temporal relationships (e.g., *M before N, K before L*) are defined as $L_C$. The same procedure applied to *after* relationship.

**During relationship** If *X during Y* is selected as an element of $L_A$, then any label with only one action label between X and Y is defined as an element of $L_B$. In that, labels such as *X during A, A during Y* are included in $L_B$. Other labels having the during relationship are defined as $L_C$.

**Same action labels** For the temporal labels having the same action labels, e.g. *A before A, B during B*, $L_A, L_B, L_C$ are constructed in the level of actions. In that, as discussed in Section 4.2.1, object and action types are considered to construct compositional generalization split.

After constructing the $L_A, L_B, L_C$, video clips corresponding to $L_A$ and $L_C$ are defined as the training set, while videos of label $L_B$ are defined as the test set. The data split ensures the following characteristics.

For training and test dataset $\mathcal{D}^2_{train}, \mathcal{D}^2_{test}$,

$$\begin{aligned} \mathcal{A}_a &= \text{SEEN}_{\mathcal{A}_a}(\mathcal{D}^2_{train}) = \text{SEEN}_{\mathcal{A}_a}(\mathcal{D}^2_{test}), \\ T &= \text{SEEN}_{T}(\mathcal{D}^2_{train}) = \text{SEEN}_{T}(\mathcal{D}^2_{test}). \end{aligned} \tag{3}$$

where, $\text{SEEN}_{\mathcal{A}_a}(\mathcal{D})$ and $\text{SEEN}_{T}(\mathcal{D})$ represent the set of atomic action and temporal relation types included in dataset $\mathcal{D}$, and $\mathcal{A}_a, T$ represent the set of total atomic action and temporal relation types.

The two types of data split are available at https://github.com/hy0Y/ST-GT/tree/main/CG_data_split.

## 5. Experimental Results

We evaluate the performance of the suggested algorithm with the original CATER dataset and the newly suggested split in section 4. Also, to demonstrate our algorithm in the real-world dataset, we experimented with the MOMA-LRG[23] video dataset.

### 5.1. Evaluation Metric and Implementation

According to the evaluation protocol of previous studies, the mean average precision score (mAP) is used as an evaluation metric for Tasks 1 and 2.

**Architecture details** To implement the ST-GT module, two transformer layers with 4 heads are applied. For the object-oriented video encoder, one layer (for task 1, three layers for task 2) of GPT-2 architecture is adopted. For both modules, 128-dimensional and 256-dimensional embeddings for Task 1 and 2 are used, respectively.

During training, our model takes 2 hours for 50 epochs with a batch size of 8 on 1-A100 machine. We use the AdamW optimizer with $\beta_1$=0.9, $\beta_2$=0.99, and a weight decay of 0.01, with a learning rate of 1e-4.

### 5.2. Quantitative Results

Across tasks 1 and 2, the suggested method achieves a new state-of-the-art performance with meaningful margins. In particular, our model significantly outperformed comparative methods for task 2, which requires understanding the temporal relationship between two actions. Based on these quantitative results, we can argue that the disentangled object-level representations are crucial for understanding complex patterns defining actions present in the long videos.

**Ablation Study** To demonstrate our proposed method, we conducted two ablation studies. First, to assess the impact of $\mathcal{L}_{comp}$ mentioned in Section 3.4, we conducted experiments by excluding $\mathcal{L}_{comp}$. As shown in Table 3, excluding $\mathcal{L}_{comp}$ and using only $\mathcal{L}_{vid}$ results in a significant performance degradation in both tasks. This demonstrates that

Table 1. mAP score of comparative and our model on Task 1.

| Method | mAP |
|---|---|
| FasterRCNN [29] | 63.85 |
| Single stream SCI3D [29] | 91.82 |
| SCI3D [29] | 96.77 |
| R3D [4, 12] | 98.8 |
| R3D + NL [37] | 98.9 |
| Ours | **99.88** |

Table 2. mAP score of comparative and our model on Task 2.

| Method | mAP |
|---|---|
| FasterRCNN [29] | 25.45 |
| R3D [4, 12] | 44.2 |
| R3D + NL [37] | 45.9 |
| R3D + LSTM [10] | 53.4 |
| R3D + NL + LSTM [10] | 53.1 |
| SCI3D + LSTM [29] | 66.71 |
| Single stream SCI3D + LSTM [29] | 69.76 |
| ViViT[1] | 66.18 |
| FROZEN[2] | 66.64 |
| Ours | **75.40** |

structurally decomposing and comprehending complex actions, facilitated by $\mathcal{L}_{comp}$, is effective in video understanding.

Table 3. Ablation study of the loss components

| | Task 1 | | Task 2 | |
|---|---|---|---|---|
| | val | test | val | test |
| $L_{vid}$ | 90.42 | 90.38 | 56.16 | 53.53 |
| $L_{vid} + L_{comp}$ | 99.89 | 99.88 | 76.07 | 75.40 |

Finally, we provide an ablation study about the three different token types of the ST-GT in Table 4. As can be seen from the experimental results, it is crucial to define spatial and temporal connectivity information in spatiotemporal graphs as distinct types for learning to understand the semantic structure of the entire video.

**Compositional Generalization Test** To check the compositional generalization ability, the suggested and comparative methods are evaluated with the newly suggested splits.

As a comparative method for the first compositional generalization split, the R3D-based method[4, 10, 12, 37] is used because the code for the SCI3D method[38] was not published. In Table 5, it can be observed that our proposed method demonstrates remarkable performance improvements even for unseen labels ($\mathcal{D}^1_{test}$) than the comparative method. Because the comparative method performs classification based on labels defined by combinations of action and object types, it is limited to consider the separate influence of each element(action and object types) on the classification decision. However, the suggested method, by considering information separately for each action and object type during training, can readily learn even novel combinations. Furthermore, not only for the compositional generalization setting but also the suggested method outperforms the comparative method in a setting similar to the traditional machine learning problem, which is learning and evaluating with the same label set($\mathcal{D}^1_{train}$).

In the data split based on Task 2(considering temporal relationships between actions), we observed even greater performance improvements. As a comparative method for this setting, two R3D-based methods (R3D and R3D with two LSTM layers) are used because the code for the SCI3D method was not published. This compositional generalization setting, compared to the previous one, involves a large number of class labels and is more complex, comprising three elements to form the label(object, action, and temporal relationship types). As a result, the comparative method shows very low performance, around 2%. On the other hand, the suggested method demonstrated the ability to pre-

Table 4. Ablation study of 3 token types composing ST-GT's input

| Token Types | | | Dataset | |
|---|---|---|---|---|
| N | SE | TE | CATER | MOMA-LRG |
| ✓ | - | - | 72.49 | 49.76 |
| ✓ | ✓ | - | 74.49 | 66.47 |
| ✓ | - | ✓ | 74.48 | 66.78 |
| ✓ | ✓ | ✓ | **75.40** | **72.83** |

Table 5. Compositional generalization result of Task 1

| | Validation | | Test | |
|---|---|---|---|---|
| | Baseline | Ours | Baseline | Ours |
| $\mathcal{D}^1_{test}$ | 25.78 | 72.95 | 25.05 | 71.53 |
| $\mathcal{D}^1_{train}$ | 93.95 | 98.9 | 93.04 | 99.1 |

Table 6. Compositional generalization result of Task 2

| Split | | R3D | R3D+LSTM | Ours |
|---|---|---|---|---|
| Validation | $\mathcal{D}^2_{test}$ | 4.31 | 4.00 | 69.45 |
| | $\mathcal{D}^2_{train}$ | 60.32 | 76.49 | 81.1 |
| Test | $\mathcal{D}^2_{test}$ | 3.47 | 3.51 | 69.2 |
| | $\mathcal{D}^2_{train}$ | 58.23 | 74.84 | 80.9 |

dict unseen label sets in this problem with an accuracy of over 80% (Table 6).

### 5.3. Qualitative Results

To thoroughly understand the characteristics of our proposed model, we conducted two qualitative experiments. Firstly, in the CATER Task 2 setting, we examined how the accuracy changes with variations in the time difference between the occurrences of the two actions that constitute the label (for example, for a label *X before Y*, the time difference between the occurrence of X and Y). As can be seen in Figure 5 (a), we confirmed that the proposed method exhibits excellent performance overall, particularly achieving remarkably high accuracy in predicting long video clips. Conventional video understanding methods usually extract features from uniformly segmented video clips and comprehend the overall video by averaging them or adding simple temporal models. Therefore, this approach has a crucial disadvantage; as the video lengthens, the information becomes aggregated. On the other hand, the suggested method efficiently extracts information corresponding to semantic units in long videos and can comprehend complex labels through the relationships between these units, making it robust to changes in length.

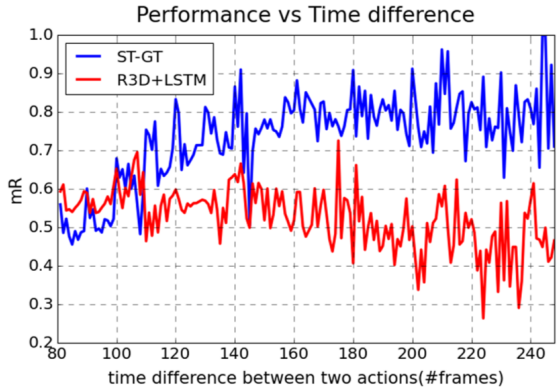
Similarly, we examined how the number of interfering actions between two actions affects the prediction. As can be seen in Figure 5 (b), similar to the time difference case, the suggested method demonstrated high performance regardless of the presence of a disrupter between the two actions.

### 5.4. Experiments with Real-world Dataset

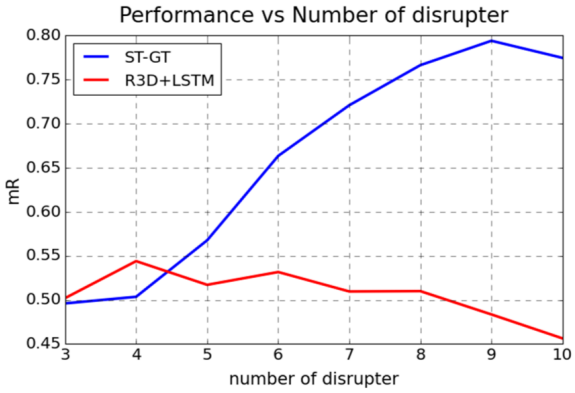
Finally, additional experiment results with a real-world video dataset are presented. The MOMA-LRG dataset[23] is proposed to recognize human actions included in videos, providing annotations in the form of symbolic graphs representing relationships between objects and humans in the video. With the symbolic graph provided in the dataset, we show the results of the action classification tasks (sub-activity level labels in the MOMA-LRG dataset) in a similar setting to the previously conducted with the CATER dataset. As shown in Table 7, our model achieved remarkable performance with mAP 72.83, outperforming comparative models, and showed the adaptability and effectiveness of our method on a real-world dataset.

Table 7. Performance comparison and model details on MOMA-LRG dataset

| Method | mAP | # params | Inference time (ms) |
|---|---|---|---|
| ViViT [1] | 65.09 | 99M | 23 |
| FROZEN [2] | 69.36 | 115M | 40 |
| Ours | **72.83** | **7.3M** | **20** |




(a)




(b)

Figure 5. Performance with variations in times and disrupters. In (a), we demonstrated how the mean recall performance varies with the temporal difference between two actions in videos. The conventional model(R3D+LSTM) that extracts features on a frame-by-frame basis and averages them exhibits a decrease in performance as the temporal gap between actions increases. However, our model maintains strong performance even in long videos. Also in (b), the R3D+LSTM model suffers when the number of interfering actions between two actions increases.

## 6. Conclusion with Future Direction

We have introduced a novel algorithm effective in understanding complex semantic structures within long videos. We anticipate that this research will contribute to achieving human-level video understanding for long videos such as movies or TV shows and human-level reasoning performance.

## 7. Acknowledgements

This work is supported by IITP grant funded by MSIT (Grant No. 2022-0-00264/40%, 2022-0-00612/20%, 2022-0-00951/20%,) and IITP Artificial Intelligence Graduate School Program for Hanyang University funded by MSIT (Grant No. 2020-0-01373/20%).

## References


[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 2, 7

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 7

[3] Amir Bar, Roei Herzig, Xiaolong Wang, Anna Rohrbach, Gal Chechik, Trevor Darrell, and Amir Globerson. Compositional video synthesis with action graphs. *arXiv preprint arXiv:2006.15327*, 2020. 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 7

[5] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020. 3

[6] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021. 2

[7] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. 2

[8] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 214–229. Springer, 2020. 2

[9] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13505–13515, 2021. 2

[10] Rohit Girdhar and Deva Ramanan. Cater: A diagnostic dataset for compositional actions and temporal reasoning. *arXiv preprint arXiv:1910.04744*, 2019. 2, 3, 5, 7

[11] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11287–11297, 2021. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[13] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, and Trevor Darrell. Deep compositional captioning: Describing novel object categories without paired training data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2016. 2

[14] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710, 2020. 3

[15] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012. 1, 2

[16] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems*, 35:14582–14595, 2022. 3

[17] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008. 2

[18] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021. 3

[19] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017. 2

[20] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64:107–123, 2005. 2

[21] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 2

[22] Juncheng Li, Junlin Xie, Long Qian, Linchao Zhu, Siliang Tang, Fei Wu, Yi Yang, Yueting Zhuang, and Xin Eric Wang. Compositional temporal grounding with structured variational cross-graph correspondence learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3032–3041, 2022. 2

[23] Zelun Luo, Zane Durante, Linden Li, Wanze Xie, Ruochen Liu, Emily Jin, Zhuoyi Huang, Lun Yu Li, Jiajun Wu, Juan Carlos Niebles, et al. Moma-lrg: Language-refined graphs for multi-object multi-actor activity parsing. *Advances in Neural Information Processing Systems*, 35:5282–5298, 2022. 2, 6, 8

[24] Wonpyo Park, Woonggi Chang, Donggeon Lee, Juntae Kim, and Seung-won Hwang. Grpe: Relative positional encoding for graph transformer. *arXiv preprint arXiv:2201.12787*, 2022. 3

[25] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 1

[26] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 2

[27] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph

transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020. 3

[28] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 2

[29] Akash Singh, Tom De Schepper, Kevin Mets, Peter Hellinckx, José Oramas, and Steven Latré. Deep set conditioned latent representations for action recognition. *arXiv preprint arXiv:2212.11030*, 2022. 7

[30] Cees GM Snoek, Marcel Worring, et al. Concept-based video retrieval. *Foundations and Trends® in Information Retrieval*, 2(4):215–322, 2009. 2

[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1, 2

[32] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3

[35] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 2

[36] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[37] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2, 7

[38] Zhuoyuan Wu, Jian Zhang, and Chong Mou. Dense deep unfolding network with 3d-cnn prior for snapshot compressive imaging. *arXiv preprint arXiv:2109.06548*, 2021. 7

[39] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020. 1

[40] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019. 2

[41] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016. 2

[42] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019. 3