Compositional Video Understanding with Spatiotemporal Structure-based Transformers

Supplementary Material

In the supplementary material, we provide a detailed explanation of the ST-GT module and the data split.

8. Implementation details of ST-GT

In this section, we present a detailed implementation of constructing the input token of a spatiotemporal graph transformer in Section 3.2. Firstly, symbolic graph construction to obtain $\mathcal{G} = \{G_t\}_{t=1,\dots,N}$, which is a sequence of snapshot graphs represented as $G_t = (A_t, X_t, E_t)$, is explained. Then, a spatiotemporal graph tokenizing method to obtain the input $\mathbf{X}$ for ST-GT is explained.

8.1. Spatiotemporal Symbolic Graph Construction

For the attribute matrix of the node X_t in snapshot graph $G_t = (A_t, X_t, E_t)$, we concatenated shape, color, size, and material features. Each word expressing the shape, color, size, and material is transformed into 50-dimensional vectors using the GloVe embedding method[25]. Based on the distance between two objects, the edges of the snapshot graph are defined. In Figure 6 (a), we consider two objects to be connected by an edge if their distance is less than *ED*, which we have set to 1.5. For the edge attribute E_t, we concatenate the attributes of the connected nodes. Finally, the adjacency matrix A_t is calculated using the edges.

8.2. Spatiotemporal Graph Tokenization

The input data for the ST-GT (Spatio-Temporal Graph Transformer) model, denoted by $\mathbf{X}$, consists of three token types. As shown in Figure 6 (b), each token is added with three types of information: feature, positional embeddings, and token type identifier. In our model, when dealing with the spatial edge type token and the temporal edge type token, we concatenate their feature vectors and positional embeddings with those of the connected node type tokens. To extract a 128-dimensional time feature vector for each object, we use the time mapping function denoted as f_t in Figure 3, which was introduced in TGAT [39]. For the pose feature, we use the 6 degrees of freedom (6 DoF) information to obtain a 12D pose vector containing a specific object's 3D global position and local coordinate information. Eigenvectors of graph laplacian L^{total} are used as positional embeddings where L^{total} is calculated from A^{total} defined in Section 3.1.

9. Detailed explanation on the Compositional Generalization Test for Composite Action Recognition Dataset

In this section, we provide a more detailed explanation of the conditions used to divide the label set for Task 2 in Section 4.2.2. The label set for Task 2 has been divided into three cases, each of which is subject to different temporal relationships. The temporal relationships for each case are as follows.

Before or After relationship A randomly selected label *X before Y*, where $X \neq Y$, is defined as an element in L_A, then the opposite label *Y before X* is defined as the element in L_B. This condition can be expressed mathematically as follows:

$$\text{`}X \textit{ before } Y\text{'} \in L_A \iff \text{`}Y \textit{ before } X\text{'} \in L_B. \tag{4}$$

The same principle holds for the *after* relationship.

During relationship A randomly selected label A_1 *during* A_2, where $A_1 \neq A_2$, is defined as an element in L_A, then B_1 *during* B_2, where $B_1 \neq B_2$, is defined the element in L_B. In this case, L_A and L_B satisfy the following conditions:

$$\begin{aligned} &\forall \text{`}A_1 \textit{ during } A_2\text{'} \in L_A, \exists \text{`}B_1 \textit{ during } B_2\text{'} \in L_B \\ &\text{s.t.} \quad |\{A_1, A_2\} \cap \{B_1, B_2\}| = 1. \end{aligned} \tag{5}$$

In other words, only one of the two actions in any label included in L_A paired with a label in L_B is the same.

Same action labels If any temporal label between identical actions is included in L_A, the corresponding label in L_B also represents a temporal label between identical actions. An element in L_A is defined as (X, t, X), then the element in L_B is defined as (Y, t, Y), where t is an element of the set of total temporal relation type, and $X \neq Y$. When dividing L_A and L_B, we approached it by considering each action with object and action types as in Task 1. Therefore, if we express two actions X and Y with object and action types, we have $X = (o_X, a_X)$ and $Y = (o_Y, a_Y)$. Here, $o_X, o_Y \in S$, and $a_X, a_Y \in A$, where S and A represent the sets of total object and action types. In this context, L_A and L_B satisfy the following conditions:

$$\begin{aligned} &\forall((o_X, a_X), t, (o_X, a_X)) \in L_A, \\ &\exists((o_Y, a_Y), t, (o_Y, a_Y)) \in L_B \\ &\text{s.t.} \quad (o_X = o_Y) \vee (a_X = a_Y). \end{aligned} \tag{6}$$

In other words, the temporal relation type between the elements in L_A and L_B is the same, and the two actions X and

Y share only one of an object or action type. As L_A and L_B are disjoint sets, $X \neq Y$ is implied.

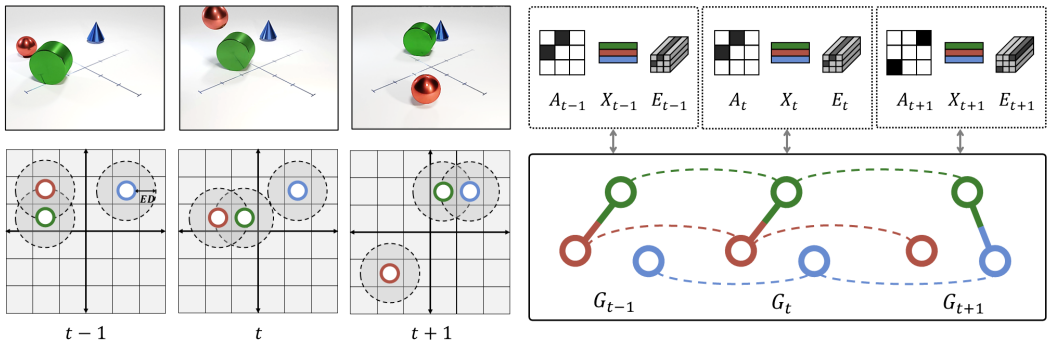
According to the defined L_A and L_B, it holds that $L_C = \mathcal{A}_c - (L_A \cup L_B)$, where $\mathcal{A}_c$ represents the set of total composite actions. Ultimately, the conditions $|L_A| = |L_B| = 98$ and $|L_C| = 105$ are satisfied. The labels provided in the CATER dataset for Task 2 include seven labels that are part of the Task 2 labels but have never appeared in the CATER dataset videos. Consequently, these seven labels are included in L_C. Therefore, L_C contains seven more labels than L_A and L_B. Details about these seven labels can be found in [10].

10. Implementation details of MOMA-LRG dataset experiments

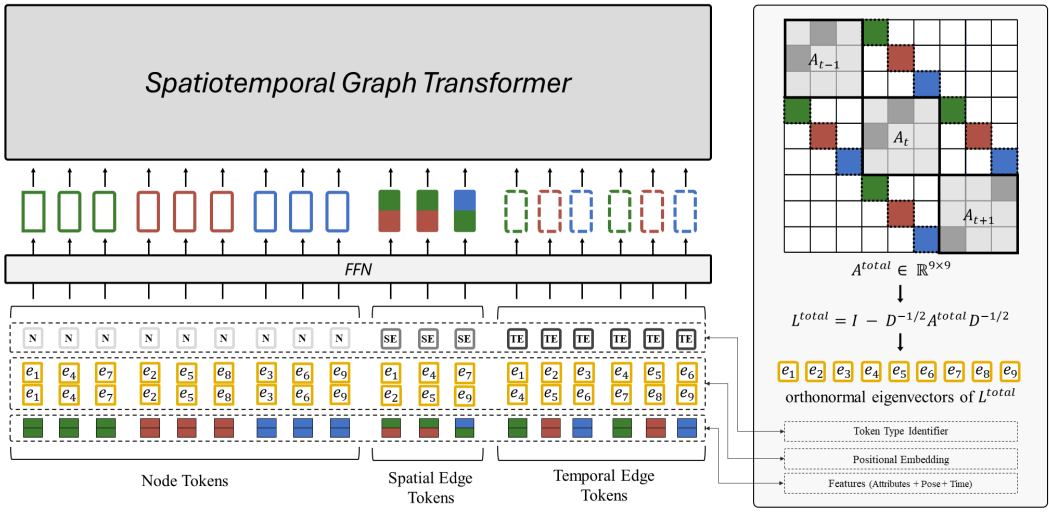
As alluded to in section5.4, we conducted experiments on a multi-label classification problem (subactivity level classification) using the MOMA-LRG dataset. As the MOMA-LRG dataset provides a frame-level symbolic graph, we defined the object and actor as nodes and the relation between nodes as edges. We used sentence-bert[26] for nodes and edges as our input feature. For the formulation of the multi-label classification problem, we excluded videos without sub-activity and those with an excessive number of objects, that exceeded the model's input token limit (e.g., basketball-playing videos). Therefore, the number of videos for train, valid, and test sets is 757 / 182 / 227.

11. Qualitative for unseen types of videos

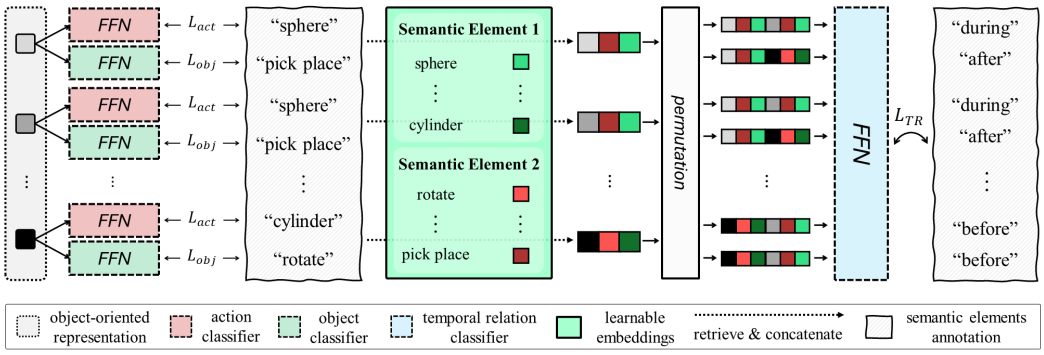
We examined how our model makes predictions for entirely unseen types of videos. As discussed in Section 4.1, Task 1 of the CATER dataset contains only 14 labels among 20 possible combinations(5 types of object and 4 types of action), due to certain actions not being applicable to specific objects (for example *cone rotate, cube contain*). For the zero-shot compositional generalization setting, we synthesized new videos corresponding to *cone rotate, cube contain* labels(Figure 7 (a)-(b)). In Figure 7 (c), the embeddings for two synthesized objects are plotted with other embeddings. From this Figure, we can observe that embeddings for objects never seen before are located in proximity to embeddings that possess similar semantic units, despite never having been trained together. Based on these analyses, we anticipate that the proposed model will exhibit robust performance even in zero-shot settings.



(a)



(b)



(c)

Figure 6. (a) Details of Symbolic Graph Construction. At the timestamp $t-1$, when the distance between the red sphere and the green cylinder is less than *ED*(1.5), two objects are connected with an edge as shown in G_{t-1}. These edges enable the identification of action patterns related to both objects, such as *contain*. (b) Example of Spatiotemporal Graph Tokenization. Calculating graph laplacian when $N = 3$ and $n = 3$. Positional embeddings are composed with the eigenvectors from the total adjacency matrix which is denoted as $A^{\text{total}} \in \mathbb{R}^{9\times9}$. (c) Details of Embedding Disentangling Module. As mentioned in the main paper, we defined three semantic elements (action, object, temporal relation) and obtained corresponding semantic element annotations. For two of these(action, object), the object-oriented representations incoming into the embedding disentangling module are supervised by two classifiers. On the other hand, embeddings for the relationship between two atomic actions are required for temporal relations. Therefore, we first retrieved embeddings corresponding to the predicted semantic elements from the two classifiers and concatenated them to obtain embeddings for each atomic action. Subsequently, we introduced a permutation process to obtain embeddings for the relationships between atomic actions, and trained temporal relations using another classifier.

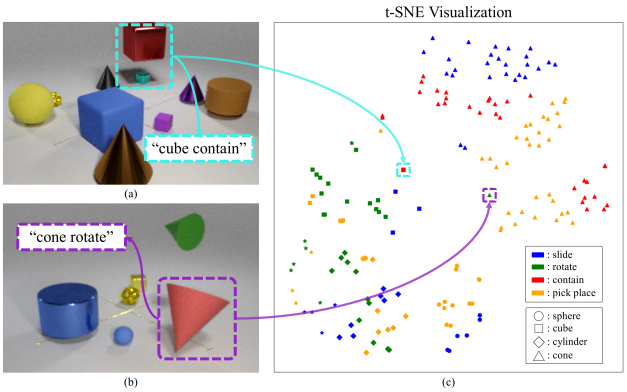


Figure 7. Embeddings with unseen composite labels. (a) and (b) show generated videos containing unseen labels *cube contain*(green box), *cone rotate*(yellow box) respectively. (c) demonstrates how the embeddings for *cube contain*(green box) are positioned between the representations of *cubes* (square data points) and *contain* (red data points). Also, the embeddings for *cone rotate* (yellow box) are located between the representations of *cone* (triangle data points) and *rotate* (green data points). This illustrates the model's ability to accurately predict outcomes for entirely unseen labels by combining known features.