STVGFormer: Spatio-Temporal Video Grounding with Static-Dynamic Cross-Modal Understanding

Zihang Lin
Sun Yat-sen University
Guangzhou, China
linzh59@mail2.sysu.edu.cn

Chaolei Tan
Sun Yat-sen University
Guangzhou, China
tANCHlei@mail2.sysu.edu.cn

Jian-Fang Hu
Sun Yat-sen University
Guangzhou, China
hujf5@mail.sysu.edu.cn

Zhi Jin
Sun Yat-sen University
Guangzhou, China
jinz26@mail.sysu.edu.cn

Tiancai Ye
Tencent
Guangzhou, China
tiancaiye@tencent.com

Wei-Shi Zheng
Sun Yat-sen University
Guangzhou, China
wzheng@ieee.org

ABSTRACT
In this technical report, we introduce our solution to human-centric spatio-temporal video grounding task. We propose a concise and effective framework named STVGFormer, which models spatio-temporal visual-linguistic dependencies with a static branch and a dynamic branch. The static branch performs cross-modal understanding in a single frame and learns to localize the target object spatially according to intra-frame visual cues like object appearances. The dynamic branch performs cross-modal understanding across multiple frames. It learns to predict the starting and ending time of the target moment according to dynamic visual cues like motions. Both the static and dynamic branches are designed as cross-modal transformers. We further design a novel static-dynamic interaction block to enable the static and dynamic branches to transfer useful and complementary information from each other, which is shown to be effective to improve the prediction on hard cases. Our proposed method achieved 39.6% vIoU and won the first place in the HC-STVG track of the 4th Person in Context Challenge.

CCS CONCEPTS
• Computing methodologies → Visual content-based indexing and retrieval. Scene understanding.

KEYWORDS
spatio-temporal video grounding, cross-modal learning

1 METHOD
Human-centric spatio-temporal video grounding (HCSTVG) task [5] aims to localize the target person spatially and temporally according to a language query. We propose a framework named STVGFormer to tackle this problem. As illustrated in Figure 1, our framework mainly consists of a static branch and a dynamic branch to model static and dynamic visual-linguistic dependencies for complete cross-modal understanding. The static branch performs cross-modal understanding for static contextual information, i.e., finding the target person that matches the query text in a still frame according to static visual cues, like appearance which is important for achieving accurate spatial grounding. The dynamic branch performs cross-modal understanding for dynamic contextual information, i.e., finding the temporal moment that best matches the query sentence according to dynamic visual cues, like action which is important for achieving accurate temporal grounding. In order to enable information transition between the two branches, we further develop a novel static-dynamic interaction block to exchange complementary information between the static and dynamic branches.

In this way, both the static and dynamic branch can absorb useful and complementary information from the other one, which can help greatly reduce the uncertainty in the ambiguous and hard cases. Finally, we employ a bounding box prediction head on top of the static branch and a temporal moment prediction head on top of the dynamic branch to predict spatial and temporal grounding result, respectively. Overall, our framework is concise but effective. In the following, we will introduce each component in detail.

1.1 Static Branch
The static branch is employed to perform cross-modal static context understanding. Following MDETR[5], we define our static branch as a stack of $N$ cross-modal transformer encoder layers and $N$
Figure 1: An overview of the proposed framework. Our framework mainly consists of a static branch and a dynamic branch. The static branch learns to predict the spatial location (i.e., bounding boxes $\hat{b}_1, \hat{b}_2, ..., \hat{b}_n$) of the target object according to static cues like human appearance. The dynamic branch learns to predict the starting and ending time $(t_c, t_e)$ for the target moment according to dynamic cues like human action. We further devise a static-dynamic interaction block which enables the two branches to query useful and complementary information from the opposite branch.

Figure 2: The detailed architectures of the proposed framework. (a): The cross-modal transformer in the static branch. (b): The architecture of each space-time cross-modal transformer layer in the dynamic branch.

The dynamic branch performs cross-modal understanding for dynamic contextual information. As illustrated in Figure 1, we first extract clip-level visual features $F_v \in \mathbb{R}^{T \times H \times W \times d}$ from $T$ uniformly sampled video clips by a pretrained 3D-CNN[2] and employ an FC embedding layer to project the channel dimension from $c$ to $d$. Then the features after projection are fed into a Space-Time Cross-Modal Transformer (STCMT) which consists of $N$ layers to model the visual-linguistic dependencies from a dynamic perspective. The detailed architecture of each layer in STCMT is illustrated in Figure 2 (b). In each layer, we first perform intra-modality self-attention for the dynamic visual features and linguistic features. For visual features, in order to reduce computation cost, we follow TimesFormer[1] to split the spatio-temporal attention into separate attentions. Denoting the visual features after self-attention as $V_{\text{att}}$, we perform the following fusion operation: $V_{\text{att}}^f = \sum_{i=1}^{T} V_{\text{att}}^i$, where $V_{\text{att}}^i$ denotes the i-th attentions. The fused feature $V_{\text{att}}^f$ is then projected into the channel dimension via a fully-connected layer $F_{\text{proj}}$. The features after fusion are then used to attend over the linguistic features, and vice versa.
After computing the cross-attention between the visual features \( F_v \) and \( F_l \) as:

\[
\begin{align*}
Q^l_i &= W_{qV}^l F^l_i, K_v &= W_{kV} F_v, V_v &= W_{vV} F_v, \\
Q_l &= W_{ql} F_l, K_l &= W_{kl} F_l, V_l &= W_{vl} F_l, \\
\tilde{F}_v^{(h,w)} &= F_v^{(h,w)} + \text{Attention}(Q^l_i, K_v, V_v), \\
F_v &= \tilde{F}_v + \text{Attention}(Q_l, K_l, V_l), \\
\text{Attention}(Q,K,V) &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,
\end{align*}
\]

where \( W_{qV}, W_{kV}, W_{vV}, W_{ql}, W_{kl}, W_{vl} \) are learnable weighted matrices for computing queries, keys and values for the attention mechanism. \( F_v \in \mathbb{R}^{T \times d} \) is obtained by conducting mean pooling on \( F_v \) along the spatial dimensions. \( d_k \) is the dimension of the queries and keys. \( F_v^{(h,w)} \in \mathbb{R}^{T \times d} \) indicates the visual feature at spatial position \((h,w)\). \( F_v, F_l \) are the output visual and linguistic features after cross-attention, respectively. The cross-attention operation is designed in the way that rich interactions between the dynamic visual and linguistic features can be explored, which enables the model to learn a powerful cross-modal representation about the depicted dynamic cues. This is essential for grounding temporal moments. After computing the cross-attention between the visual features and linguistic features, we finally employ Feed-Forward Network (FFN) to process both features. In our dynamic branch, rich context is learned from the two modalities, and the visual dynamic features are guided to attend to the object-related regions. This can serve as a strong guidance to help the dynamic branch focus more on the dynamic variations around the object-related regions. In this way, the model can better capture the motion of the target object.

The dynamic-to-static interaction block is depicted in Figure 3(c). In each block, the object query \( O_M^i \) in the static branch first queries some dynamic information from \( F_v^{(1)}[t] \) (the output visual feature of the \( t \)-th layer in Space-Time Cross-Modal Transformer for frame \( t \)). Then, the object queries of different frames are mixed up by a temporal self-attention layer. This block enhances the object query representation with cross-frame dynamic information like object motion and human action. With the proposed static-dynamic interaction block, both the static and dynamic branches can effectively learn the complementary information from the other branch.

**1.4 Prediction Heads**

In this section, we introduce the prediction heads used in our static branch and dynamic branch. **Prediction Head for Static Branch.** The prediction head of the static branch is designed to predict the location of the target object. The input to the prediction head is the object query representation \( O^M_M \in \mathbb{R}^d \) outputted by the static branch at \( t \)-th frame. We implement a 3-layer MLP to regress the bounding box location related to the objects depicted in the query text, by utilizing the cross-attention matrices calculated in decoder layers of the static branch. Concretely, we employ static-to-dynamic interaction in each network layer as follows:

\[
\hat{F}_v^i = \text{LayerNorm}(F_v^i + A^i \odot FC(F_l^i)),
\]

where \( F_v^i \in \mathbb{R}^{T \times H \times W \times d} \) is the output visual feature of the \( i \)-th layer in dynamic branch and \( A^i \in \mathbb{R}^{T \times H \times W \times d} \) is the cross-attention weights (replicate to have \( d \) channels at the last dimension) calculated between the object queries \( \{O_M^t\}_{t=1}^T \) and corresponding encoded memories \( \{M_i^t\}_{t=1}^T \) in \( B_i^t \). \( \odot \) indicates hadamard product. The intuition behind this design is that the attention weights in the decoder layers of the static branch will attend to the object regions matching the language description, and this can serve as a strong guidance to help the dynamic branch focus more on the dynamic variations around the object-related regions. In this way, the model can better capture the motion of the target object.
(represented by center coordinates and size) \( \hat{b}_t \in \mathbb{R}^d \) of the target object. In order to make the static branch be aware of whether the input frame is well matched to the input text query, we further employ an FC layer to predict a score \( p_t \), which indicates whether frame \( t \) is inside the target temporal moment.

**Prediction Head for Dynamic Branch.** The prediction head in the dynamic branch is designed to predict the time span of the target temporal moment. Specifically, we first perform mean pooling on the features outputted by the dynamic branch at the spatial dimension to obtain a temporal feature \( F_d \in \mathbb{R}^{T \times d} \), then employ an FC layer to adjust the dimension from \( d \) to \( d_m \). We follow Aug2DTAN[8] to implement a 2D-proposal-based prediction head and construct the 2D proposal map \( M \in \mathbb{R}^{T \times T \times d_m} \) as:

\[
M_{ij} = \begin{cases}
\text{MeanPool}\left(\left\{F_{i'd}^{1}, F_{i'+1}^{1}, ..., F_{j'd}^{1}\right\}\right) & i \leq j \\
0 & i > j
\end{cases}
\]

where \( M_{ij} \) indicates the feature representation of temporal moment proposal \( C_{ij} \) which consists of clips \( C_{i}, C_{i+1}, ..., C_{j} \). Following the implementation in [8], we employ several convolutional layers to the 2D map \( M \) to obtain a score map \( S \in \mathbb{R}^{T \times T \times 1} \) where each element \( S_{ij} \) represents the matching score of the temporal proposal \( C_{ij} \). For inference, we take the proposal with the highest score as the prediction of target temporal time span.

Similar to the static branch, we implement an auxiliary head (3-layer MLP) to predict whether each clip is inside the target temporal time span. Formally, \( S_{aux} \) in \( \mathbb{R}^{T} \) and \( S_{aux} \) indicate the probability of clip \( i \) to be inside the target moment.

\[\text{MeanPool}\left(\left\{F_{i'd}^{1}, F_{i'+1}^{1}, ..., F_{j'd}^{1}\right\}\right)\]

\[0\]

### 1.5 Model Training

We train our model with loss \( L = L_s + L_d \), where \( L_s \) and \( L_d \) are the losses on the output of the static branch and dynamic branches, respectively. Specifically, they are defined as following:

\[
\begin{align*}
L_s &= \lambda_1 L_{t}(\hat{b}, b) + \lambda_2 L_{gIoU}(\hat{b}, b) + \lambda_3 L_{aux}^s, \\
L_d &= \lambda_4 L_{t}(\hat{S}, S) + \lambda_5 L_{aux}^d.
\end{align*}
\]

where \( L_{t} \) and \( L_{gIoU} \) are the loss weights of our static branch using pre-trained MDETR[5] as the video encoder. We train our model pre-trained on AVA[3] as the video encoder. We initialize the weights of our static branch using pre-trained MDETR[5] as done in [8, 11]. We train our model with a batch size of 8 and the loss weights are set as \( \lambda_1 = 5, \lambda_2 = 2, \lambda_3 = 0.5, \lambda_4 = 5, \lambda_5 = 1 \). For details on training strategies, we follow the practice in MDETR[5]. To reduce GPU memory usage, we uniformly sampled 48 and 96 frames for the static branch during training and testing, respectively. During testing, we inference a video multiple times with different sampled frames until all frames are sampled at least once.

### 2.2 Experimental Results

We first compare our method with state-of-the-arts on HCSTVG-v2 validation set[9]. As shown in Table 1, we outperform all previous methods by a considerable margin. Then we conduct ablation study by removing the static-to-dynamic interaction blocks (termed "w/o s-to-d interaction"), the dynamic-to-static interaction blocks (termed "w/o d-to-s interaction"), and the whole static-dynamic interaction blocks (termed "w/o interaction"). As shown in Table 1, both the interaction blocks bring some performance gains, which demonstrates their effectiveness.

**HC-STVG Challenge.** We submitted the prediction results of the proposed STVGFormer to the HC-STVG track of the 4th Person in Context Workshop. The naive STVGFormer achieved 36.8% vIoU on the test set and the 5-model-ensemble STVGFormer achieved 39.6% vIoU and won the first place in the HC-STVG track (results can be found on http://picdataset.com:8000/challenge/leaderboard/hevg2022).
REFERENCES

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is Space-Time Attention All You Need for Video Understanding? In Proceedings of the International Conference on Machine Learning (ICML).

[2] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF international conference on computer vision. 6202–6211.

[3] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudhendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 6047–6056.

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

[5] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. MDETR-modulated detection for end-to-end multimodal understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 1780–1790.

[6] Yitian Yuan, Tao Mei, and Wenwu Zhu. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 9159–9166.

[7] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. 2020. Where Does It Exist: Spatio-Temporal Video Grounding for Multi-Form Sentences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).