

# Supplementary Materials: Spatiotemporal Graph Guided Multi-modal Network for Livestreaming Product Retrieval

Anonymous Authors

## 1 INTRODUCTION

In this document, we provide the following materials that are not included in our main paper due to space constraints:

- More dataset analysis.
- Experimental details on MF dataset.
- Principle of loss function.
- More model analysis.
- More visualization results.
- Source code and pretrained model.

## 2 MORE DATASET ANALYSIS

Both MF<sup>1</sup> and LPR4M<sup>2</sup> datasets used in our paper are publicly available. We analyze the properties of the two datasets, including data modality, number of product categories, number of videos and images, and number of pairs of (video, image). The quantitative comparison results are shown in Tab. 1. MF is a dataset containing only one category of fashion clothing, with two modalities, video (V) and image (I). And LPR4M is the largest publicly available LPR dataset that covers 34 commonly used live e-commerce categories, with three modalities, video, image, and text (T). Our proposed SGMN can efficiently and comprehensively utilize multiple modal information while maintaining the perception of fine-grained features. The capture of long-range spatiotemporal correlation enables our model to identify intra-class differences in clothing data and accurately distinguish inter-class diversity among multiple classes, achieving SOTA performances on both datasets above.

## 3 EXPERIMENTAL DETAILS ON MF DATASET

Regarding the fine-tuning details on the MF dataset, we initialize the weights using the model pretrained on the LPR4M dataset. The initial learning rate is  $1 \times 10^{-4}$ . The model is trained for 20 epochs on 8 NVIDIA Tesla V100 GPUs within only 4 hours. Other settings are maintained the same as those used in training on LPR4M, as described in the Experiments section of the main paper. To facilitate thorough testing, we redeveloped the evaluation code, which will also be included in the code submission. Evaluation is conducted using the model from the last epoch. For both training and testing, we extract 10 evenly spaced frames from each video as input.

## 4 PRINCIPLE OF LOSS FUNCTION

Due to space limitations, the calculation of some loss functions are simplified in the main paper. We provide the details of triplet loss [1] and symmetric cross-entropy loss in Eq.5, Eq.15, and Eq.21 of the main paper.

**Triplet loss.** The principle of triplet loss is to optimize the distance between anchor examples and positive examples to be smaller than the distance between anchor examples and negative examples,

<sup>1</sup><https://github.com/humatics/Seam-Match-RCNN>

<sup>2</sup><https://github.com/adxcreative/RICE>

Table 1: Analysis of MF and LPR4M datasets.

|          | Property       | MF         | LPR4M           |
|----------|----------------|------------|-----------------|
| Content  | Modal Category | V & I<br>1 | V & I & T<br>34 |
| Training | Videos         | 15,045     | 3,955,181       |
|          | Images         | 14,855     | 272,063         |
|          | Pairs          | 15,045     | 3,955,181       |
| Testing  | Videos         | 1,342      | 20,079          |
|          | Images         | 1,341      | 66,358          |

which is calculated as:

$$\mathcal{X}(a, p, n) = \max(d(a, p) - d(a, n) + \text{margin}, 0), \quad (1)$$

where  $a$  is an anchor,  $p$  and  $n$  are positive and negative samples,  $d(\cdot)$  is distance metric function and margin is a constant greater than 0, and we set the margin as 0.2.

In Eq.5 of main paper, for the obtained global video representations  $V_{\text{visual}} = \{v_1, v_2, \dots, v_N\} \in \mathbb{R}^{N \times D}$  and image representations  $I_{\text{visual}} = \{i_1, i_2, \dots, i_N\} \in \mathbb{R}^{N \times D}$ , we define each sample in  $V_{\text{visual}}$  and  $I_{\text{visual}}$  as  $v_j \in \mathbb{R}^{1 \times D}$  and  $i_k \in \mathbb{R}^{1 \times D}$ . Then we compute the cosine similarity matrix between  $V_{\text{visual}}$  and  $I_{\text{visual}}$  as follow:

$$M_{\text{sim}} = V_{\text{visual}} I_{\text{visual}}^T, \quad (2)$$

where each element in  $M_{\text{sim}}$  is  $m_{jk} = v_j i_k$ ,  $j, k \in [0, N]$ . Then we determine the position of the positive and negative samples in the  $M_{\text{sim}} \in \mathbb{R}^{N \times N}$ . When  $j = k$ ,  $(v_j, i_k)$  is a pair of positive samples, and the rest are negative samples:

$$\begin{aligned} (v_j, i_k)^+ &= 1 \text{ if } j = k, \\ (v_j, i_k)^- &= 0 \text{ if } j \neq k, \end{aligned} \quad (3)$$

We optimize the distance between positive and negative samples so that positive samples are close to each other and negative samples are far away from each other:

$$\begin{aligned} d(v, i, i^+) &= \max([\text{margin} - s(v, i) + s(v, i^+)], 0) \\ d(v, i, i^-) &= \max([\text{margin} - s(v, i) + s(v, i^-)], 0), \end{aligned} \quad (4)$$

where  $s(v, i)$  is denoted as the cosine similarity. Thus the triplet loss between  $V_{\text{visual}}$  and  $I_{\text{visual}}$  is:

$$\begin{aligned} \mathcal{T}(V_{\text{visual}}, I_{\text{visual}}) &= \mathcal{X}(a, (v, i)^+, (v, i)^-) \\ &= d(v, i, i^+) + d(v, i, i^-). \end{aligned} \quad (5)$$

Similarly, we calculate  $\mathcal{T}(V_{\text{text}}, I_{\text{text}})$  as the same way.

**Symmetric Cross-entropy Loss** In the SMF module, we calculate the symmetric cross-entropy loss based on the fusion feature  $\mathcal{L}_m = \mathcal{E}(\text{AvgPool}(V_{\text{cross}}))$  in Eq.21. Each training batch  $\Omega$  consists of  $N$  video-image pairs  $(\hat{v}_m, \hat{i}_n)$ , and they are discriminated in cross-entropy loss as follows:

**Table 2: Analysis of the weighted factor  $\lambda$  in similarity loss.**

| $\lambda$ | 0    | 0.3  | 0.5         | 0.7  | 0.9  | 1.0  |
|-----------|------|------|-------------|------|------|------|
| R@1       | 38.7 | 40.4 | <b>41.5</b> | 41.2 | 39.9 | 40.9 |
| R@5       | 66.5 | 67.8 | <b>68.5</b> | 68.1 | 66.8 | 67.3 |
| R@10      | 76.2 | 77.9 | <b>79.0</b> | 78.1 | 77.1 | 77.6 |

**Table 3: Performance of different feature fusion ways.**

| Fusion                        | R@1         | R@5         | R@10        |
|-------------------------------|-------------|-------------|-------------|
| None                          | 39.7        | 66.7        | 76.2        |
| Addition                      | 38.7        | 65.8        | 75.0        |
| Concatenation                 | 40.3        | 68.3        | 78.0        |
| <b>Cross-attention (Ours)</b> | <b>41.5</b> | <b>68.5</b> | <b>79.0</b> |

$$\mathcal{F}_{i2v} = -\frac{1}{N} \sum_{m \in \Omega} \log \frac{\exp(u_{mm})}{\sum_{n \in \Omega} \exp(u_{mn})},$$

$$\mathcal{F}_{v2i} = -\frac{1}{N} \sum_{n \in \Omega} \log \frac{\exp(u_{nn})}{\sum_{m \in \Omega} \exp(u_{mn})},$$
(6)

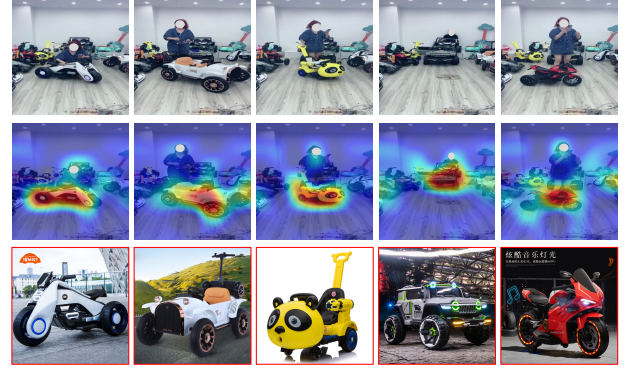
where  $u_{mn}$  is the element in  $\hat{V}_{cross}$ . Thus, the weighted sum of these two cross-entropy losses is the symmetric cross-entropy loss, corresponding to Eq.21 of the main paper:

$$\mathcal{E}(\text{AvgPool}(V_{cross})) = \frac{1}{2} (\mathcal{F}_{i2v} + \mathcal{F}_{v2i}).$$
(7)

## 5 MORE MODEL ANALYSIS

**Analysis of Weighting Factor.** We analyze the weight factor  $\lambda$  in the similarity loss function (Eq.5 of the main paper) to verify the contribution of textual information in the TE module. Specifically, we set the weight factor as  $\lambda = \{0, 0.3, 0.5, 0.7, 0.9, 1\}$  and show the results in Tab. 2. It can be seen that the use of text domain information can obtain considerable performance gains. Since the automatic speech recognition (ASR) information in the livestreaming domain usually contains excessive verbal habits and unrelated clutter information, using more weight to the text embedding will degrade the performance.

**Multi-modal Feature Fusion Mechanism.** We analyze different ways of fusing features from visual and text modalities, including feature addition, feature concatenation, and cross-attention fusion, and present their results in Tab. 3. Due to the heterogeneity of visual and textual features, the practice of directly adding multi-modal features will make the global embedding misaligned, resulting in suboptimal performance. The way of feature aggregation using feature concatenation can make features from different modalities be projected into different spaces and optimized to obtain performance gains. However, this aggregation in the feature dimension cannot perform cross-domain interactions, so the benefits to performance are limited. Therefore, we design a cross-attention layer to introduce adaptively learned parameters to optimize the cross-modal fusion of text and visual features. Experimental results demonstrate that the cross-attention-guided feature fusion layer alleviates cross-domain heterogeneity and recalibrates the multi-modal information, achieving the highest performance gain (1.8% in R@1).



**Figure 1: A real-world livestreaming scenario in the LPR4M dataset that the salesperson puts all the products to be sold but explains them one by one. Images with red boundary denote the Top-1 ranking results from our SGMN.**

## 6 MORE VISUALIZATION RESULTS

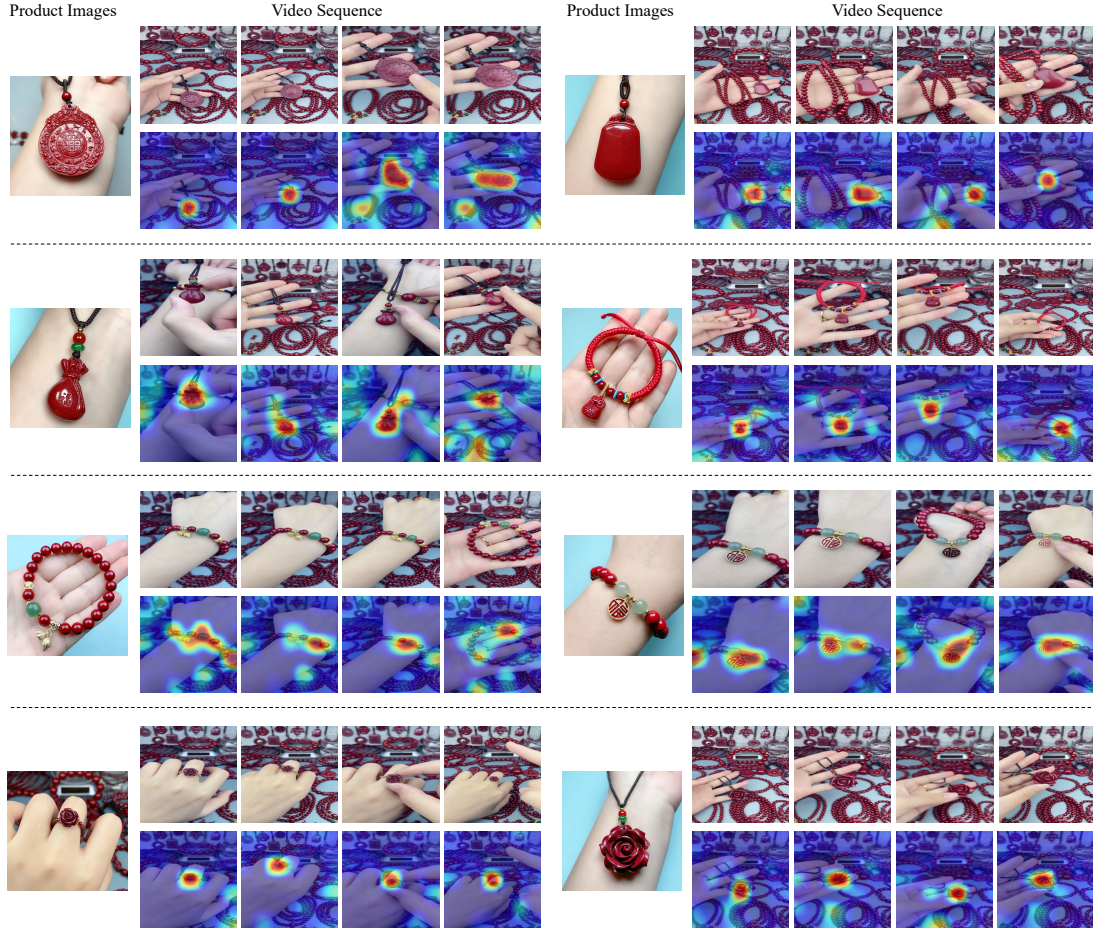
To better demonstrate the practicability of our method in real-world application scenarios of livestreaming, we provide more visualization results of livestreaming domain.

**Products in cluttered background.** In the live clip shown in Fig. 1, the salesperson showed a variety of toy cars at the same time, but only sold one product at a specific time. The dual guidance of text and visual features enables our method to accurately retrieve the expected product even if there are multiple similar products. As shown in Fig 2, we show the attention visualization of 8 similar-looking Zisha ornaments from the LPR4M dataset in the live domain. Specifically, we project the feature values of the last layer in the video encoder in Global Representation Alignment (GRA) module to  $[0,1]$ , and draw the heatmaps for the corresponding video frames. The attention maps show which areas the model focuses on, and the significant regions are marked as highlighted, where red means significant, orange, green, and blue indicate that the importance gradually decreases in order. Our SGMN still has higher discriminative ability for intended products even if the background has multiple highly similar and easily confused products.

**Products with Appearance Variations.** The results in Fig. 3 show that our model can still accurately localize regions of intended products that encounter appearance distortion due to occlusion, motion, scaling, or illumination variations. The results in Fig. 3 show that our model can guide the model to focus on the intended products from the LPR4M dataset in the cluttered background and distinguish highly similar products with fine-grained features.

**Ranking Results on the LPR4M Dataset.** We randomly select several video sequences in the LPR4M dataset and show the top 5 shop images in the ranking results of models with and without (w/o) the TE module in cluttered background products (Fig. 4(a)) and models with and without (w/o) the SMF module in highly similar background products (Fig. 4(b)). The TE model guides the network to accurately retrieve the intended products related to the video ASR keywords among various distracting products. The SMF module assists our model in capturing sufficient fine-grained features to distinguish similar samples.

**Ranking Results on the MF Dataset.** We randomly select several video sequences in the MF dataset and show the top 5 shop images



**Figure 2: Attentional visualization of highly similar products in the cluttered background scenario. Even if similar-looking Zisha ornaments appear simultaneously in the live domain, our model can still focus on the intended product in each frame.**

in the ranking results in Fig. 5. It can be seen that even if the MF dataset does not have the text modal, our SGMN without text domain features can still accurately retrieve the products that best match the current video among a large number of clothing images.

## 7 SOURCE CODE AND PRETRAINED MODEL

We submit the source code of this paper in the supplementary material, including the source code files of our SGMN and the rewritten training and test code for the MF dataset.

Please refer to the 'README.md' file to download the dataset and run our scripts. All the source code and pre-trained models will be publicly available for further research after the paper is accepted. The directory structure of our submitted code is listed as:

```

-- SGMN_CODE
-- |   train.py
-- |   eval_lpr4m.py
-- |   eval_mf.py
-- |   util.py
-- |   metric.py
-- |   README.md
-- |   dataloaders
-- |   data_dataloaders.py
-- |   dataloader_lpr4m_retrieval.py

```

```

-- |   dataloader_mf_retrieval.py
-- |   rawframe_util.py
-- |   rawvideo_util.py
-- |   modules
-- |   modeling_sgmn.py
-- |   module_clip.py
-- |   module_cross.py
-- |   module_tmc.py
-- |   module_mofusion.py
-- |   tokenization_clip.py
-- |   until_config.py
-- |   until_module.py
-- |   util_module.py
-- |   optimization.py
-- |   graph_loss.py
-- |   file_utils.py
-- |   mca.py

```

## REFERENCES

- [1] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* (2017).



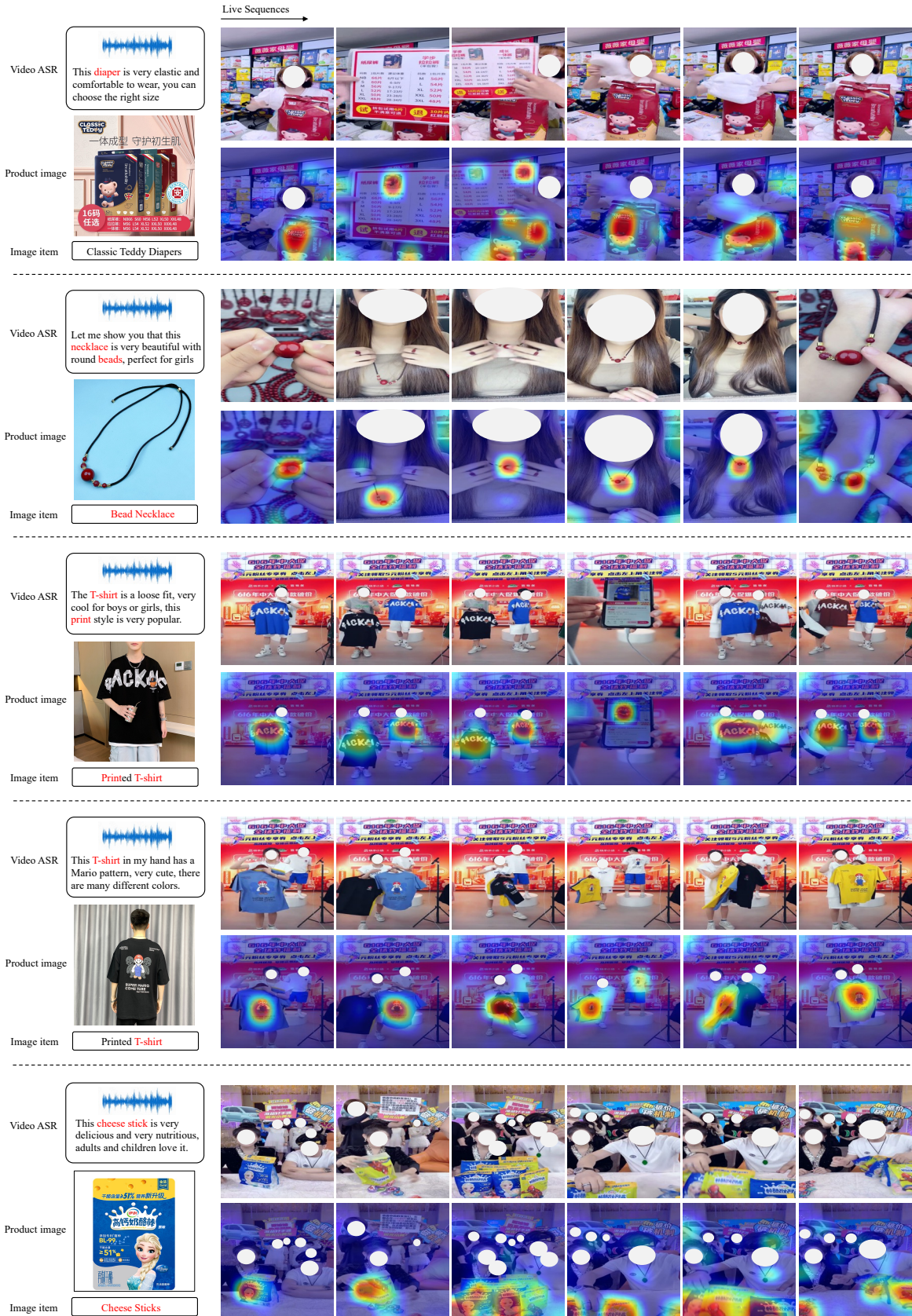


Figure 3: Representative examples to demonstrate that our method consistently focuses on the correct region for retrieving intended products in the livestreaming domain, even with appearance variations and structural deformation.



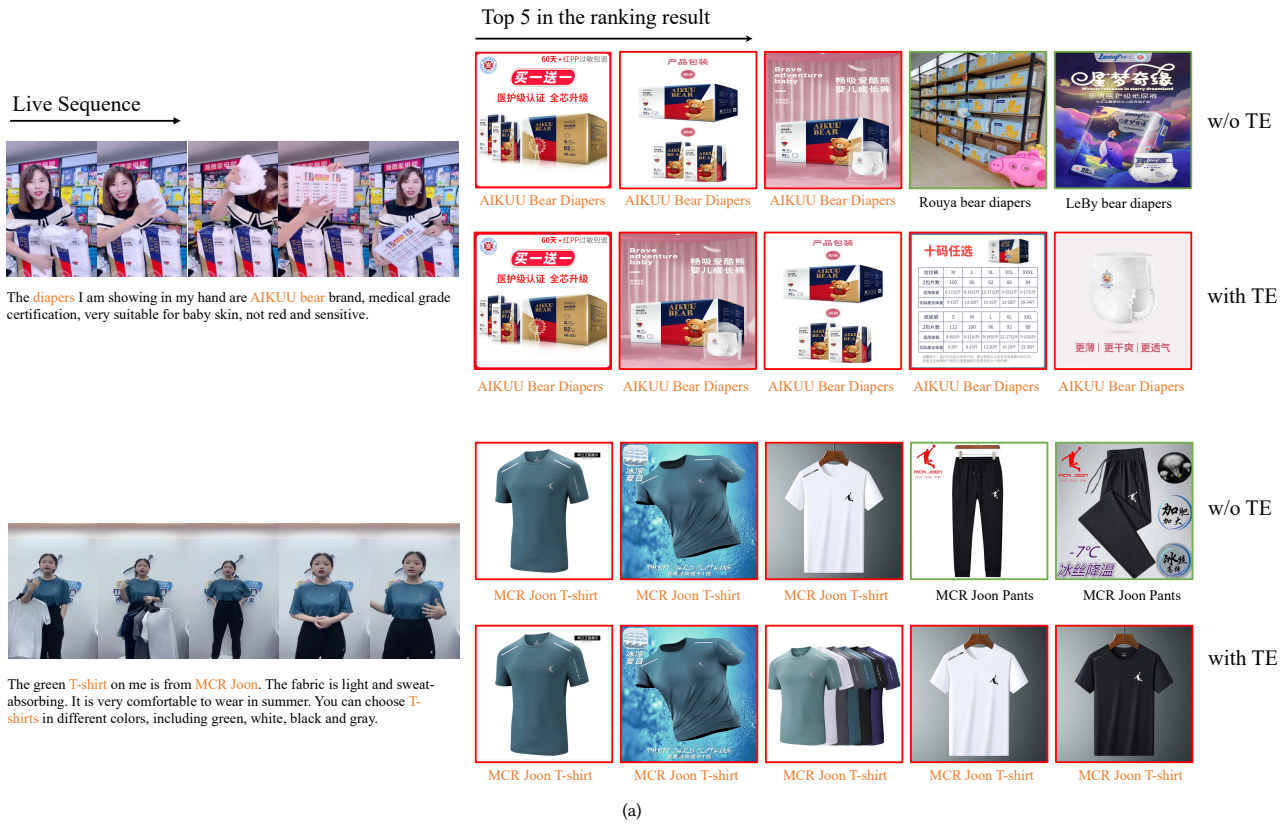


Figure 4: The ranking results of the LPR4M dataset. (a) The top 5 shop images in the ranking results of models with and without (w/o) the TE module. The texts below the video and image are ASR text and product titles, respectively. Images with red and green boundaries denote true positives and false positives. (b) The top 5 shop images in the ranking results of models with and without (w/o) the SMF module in highly similar background products.

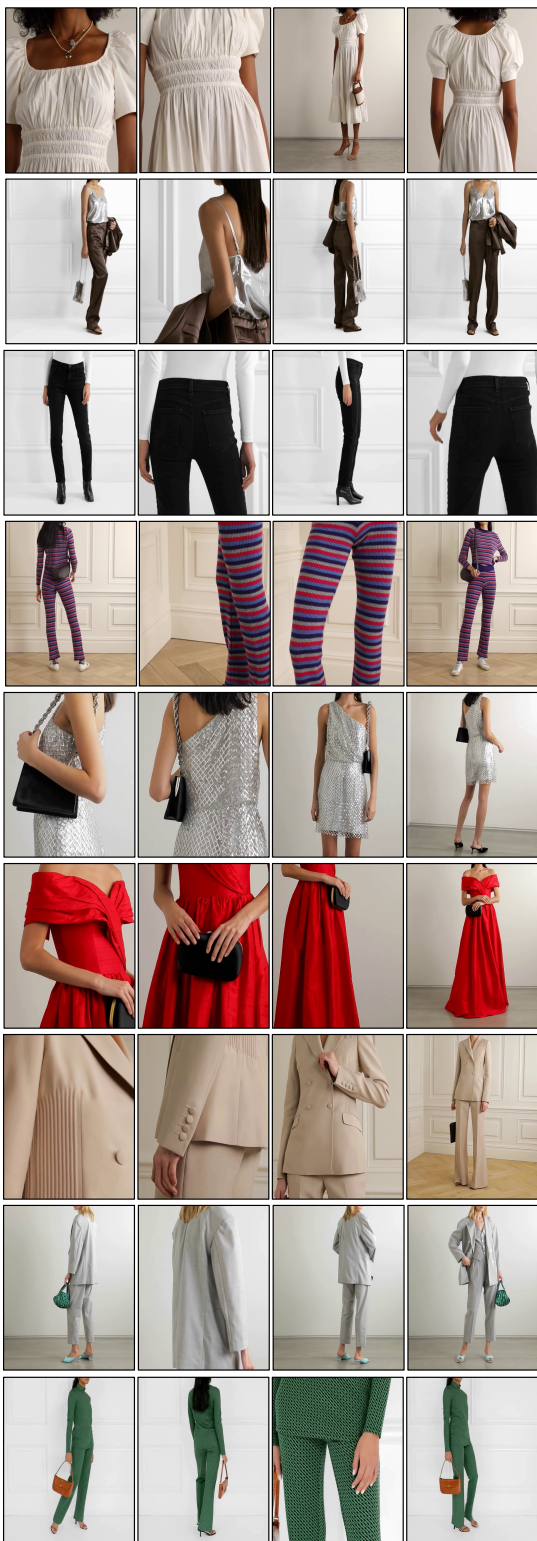


Figure 5: The top 5 shop images in the ranking results of several live sequences in the MF dataset. Images with red and green boundaries denote true positives and false positives.

