SLAN: Self-Locator Aided Network for Vision-Language Understanding

Jiang-Tian Zhai1* Qi Zhang2* Tong Wu2 Xing-Yu Chen2 Jiang-Jiang Liu† Ming-Ming Cheng1†
1VCIP, CS, Nankai University 2Tencent Youtu Lab
{jtzhai30,j04.liu}@gmail.com, townswu@tencent.com, cmm@nankai.edu.cn

Abstract

Learning fine-grained interplay between vision and language contributes to a more accurate understanding for Vision-Language tasks. However, it remains challenging to extract key image regions according to the texts for semantic alignments. Most existing works are either limited by text-agnostic and redundant regions obtained with the frozen region proposal module, or failing to scale further due to their heavy reliance on scarce grounding (gold) data to pre-train detectors. To solve these problems, we propose Self-Locator Aided Network (SLAN) for vision-language understanding tasks without any extra gold data. SLAN consists of a region filter and a region adaptor to localize regions of interest conditioned on different texts. By aggregating vision-language information, the region filter selects key regions and the region adaptor updates their coordinates with text guidance. With detailed region-word alignments, SLAN can be easily generalized to many downstream tasks. It achieves fairly competitive results on five vision-language understanding tasks (e.g., 85.7% and 69.2% on COCO image-to-text and text-to-image retrieval, surpassing previous SOTA methods). SLAN also demonstrates strong zero-shot and fine-tuned transferability to two localization tasks. The code is available at https://github.com/scok30/SLAN.

1. Introduction

Recent years have witnessed growing interest in exploring relationships between vision and language modalities. A wide range of applications have been boosted by its rapid development, such as multi-modal search engines [3, 7, 12] and recommender systems [6, 34, 35]. It motivates researchers to find semantic correspondence between two modalities and bridging their visual-semantic discrepancy. Some earlier works [14, 16, 24, 31] focused on learning joint embeddings for the two modalities, while more recent ones [17, 25, 47, 48] have turned to considering latent vision-language alignments at the level of regions and words.

In order to achieve fine-grained vision-language alignments, some works [20, 21, 26] use object detectors to extract key regions in images. Treated as black boxes, the detectors only support for fixed vocabulary object detection. Meanwhile, the extracted regions cannot adapt to different text information due to the freezing parameters of the detectors. To alleviates the problem, VinVL [47] applies a pre-trained object detector with more than 2000 classes and attributes to enrich local visual representations. However, the extended label set still limits the perceptive capability of object detectors for vision-language understanding compared to free-form text from real-world scenes.

Recently, more works have attempted to apply learnable region locators for vision-language tasks, which extract regions of interest conditioned on different texts.
Unlike previous methods using frozen object detectors, MDETR [17] builds an end-to-end framework on datasets with region-to-word annotations. GLIP [25] directly proposes grounded language-image pre-training for learning object-level, language-aware, and semantic-rich visual representations. These methods demonstrate their effectiveness in vision-language reasoning by introducing trainable locators. However, in order to supervise the training of locators, these methods require a certain amount of region-to-word grounding annotations (gold data), which are based on burdensome and expensive annotation efforts. It limits their applications on existing larger scale of vision-language datasets which have abundant but coarse-grained image and text pairs.

To address the problems above, we propose Self-Locator Aided Network (SLAN) for vision-language understanding. The designed self-locator is capable of accurately locating regions of interest based on different texts. Specifically, the self-locator consists of a region filter to select important regions and a region adaptor to update coordinates of regions with text guidance. By incorporating the self-locator into our framework, SLAN performs context-aware region extraction and vision-language feature fusion. Moreover, SLAN is trained solely on datasets with paired images and texts, making it scalable to larger pre-training settings for further performance improvements. With fine-grained region-word alignments, SLAN has a more detailed understanding of interactions in vision and language modalities.

To sum up, our contributions have three aspects:

- We propose a framework termed SLAN to capture fine-grained interplay between vision and language modalities. A self-locator is introduced to perform text-guided region adaptation, enabling dynamic region-word alignments for vision-language understanding tasks, as shown in Fig. 1.

- We demonstrate that SLAN can be easily applied to large-scale pre-training on vision-language datasets for being free from training with gold data. SLAN can also be naturally generalized to typical localization tasks, such as object detection and phrase grounding, due to its ability to locate key regions in images.

- Experiments on five vision-language understanding and two localization tasks demonstrate the effectiveness of our method. For example, SLAN achieves state-of-the-art performance on COCO image-text retrieval.

2. Related Work

2.1. Vision-language Task

Previous research has explored the relationship between visual and textual modalities and applied this knowledge to various downstream multi-modal tasks. Methods such as DeVISE [13], TBNN [36], and [49] have proposed loss functions and network structures to learn semantic visual-language alignments. Other approaches like SGG [41] and ViSTA [8] leverage prior tools or knowledge for image-text matching analysis.

Recently, leveraging visual backbone networks [11, 15, 40] and language encoders [18], vision-language pre-training on larger datasets has become increasingly popular. CLIP [31] pre-trains using 400M image-text pairs from the web, establishing global relations between images and texts. BLIP [23] benefits from extensive web data for vision-language understanding and generation tasks. Beit-3 [37] adopts mask-then-predict self-supervised training on large-scale monomodal and multi-modal data to learn internal vision-language dependencies.

However, these methods are constrained by the expense of fine-grained region-word datasets, making it challenging to directly provide local matching signals during pre-training for more accurate cross-modal knowledge. This knowledge enables models to precisely localize objects according to corresponding words, providing cues for downstream tasks.

2.2. Localization for Vision-language Task

Localization of image regions and words in sentences helps models learn local alignment. There are two kinds of methods based on whether the region proposal module is frozen or trained for vision-language tasks.

The first kind uses a frozen object detector (e.g., Faster R-CNN) pre-trained on Visual Genomes to extract detailed visual representations. Some later works (e.g.,VinVL [47], Oscar [26]) increase the number of detection labels and introduce attribute information to complement visual concepts.

The other kind relies on fine-grained annotations of the vision-language dataset for pre-training. MDETR [17] introduces a modulated detector with multi-modal datasets that have precise alignments between phrases in text and objects in images. GLIP [25] applies grounded pre-training to learn object-level, language-aware, and semantic-rich visual representations. However, these methods require vision-language data with fine-grained annotations, limiting their application on larger-scale pre-training settings.

3. Self-Locator Aided Network (SLAN)

The framework of SLAN is shown in Fig. 2. We first briefly introduce the two unimodel encoders and then the detailed structures of other components. SLAN adaptively proposes and selects informative regions with text guidance, as described in Fig. 3. Finally, we list our pre-training objectives. The relevant symbols are described in Tab. 1.
Different from most traditional object detection tasks that use the pre-defined label set, vision-language tasks usually have a wider vocabulary and free-form textual expressions. Therefore, our self-locator introduces a region filter for region importance prediction and a region adaptor for progressive region regression. By replacing fixed vocabulary prediction with region importance prediction, our self-locator assigns each region a saliency score $S_i$ to estimate the probability that the region is useful for the alignment process. For traditional detection settings, the regression targets are annotated region coordinates. Since there is no grounding (gold) annotations in our setting, we propose progressive region regression in the multi-stage region adaptor, producing intermediate updated regions in each level. These updated regions are then used for supervising the internal region proposal module. As shown in Fig. 3, SLAN dynamically adapts region embeddings in $L = 3$ levels, yielding more flexible and accurate visual representations than the global visual feature maps, or patch embeddings from the vision transformer.

### 3.2.2 Region Filter: Region Importance Prediction

When describing images, people usually focus on limited salient regions in the images [9, 10]. However, region proposal module [32] typically outputs a large number of re-
Cross Attention

![Figure 4](image-url)

Figure 4. The i-th stage of the region adaptor. The region adaptor update each region’s coordinate with text guidance. We use the feature map from vision decoder to extract region embeddings and explore latent region-word alignments.

...region proposals (e.g., 100) for an image. Directly selecting all regions will lead to unnecessary computational cost and may also cause the model to learn from some meaningless region-to-word pairs. The strategy to control the maximum number of selected regions has three steps. (a) Normalize all saliency scores of the regions. After this process, the scores are represented as $S = \{S_1, \ldots, S_k\}, S_i \in [0, 1]$. (b) Sort these regions in descending order according to their saliency scores. (c) We pick no more than top $T$ regions with saliency scores above a threshold $h$. Finally, we weight region embeddings by the scores. The saliency score of each proposed region is updated with gradients from downstream vision-language supervision, which will be described in Sec. 3.3.

### 3.2.3 Region Adaptor: Progressive Region Regression

The region adaptor aims at adjusting the coordinates of proposed regions to align with words with the same semantics. The difficulty comes from no annotated text-referenced regions as ground truths. We turn this problem into a $L$-level cascaded coarse-to-fine progressive regression process, with $L = 3$ by default. As shown in Fig. 4, the i-th level of the region regression process receives three inputs: word embeddings $E_i^T \in \mathbb{R}^{N_T \times D}$, region embeddings $E_i^G \in \mathbb{R}^{N_G \times D}$ with their coordinates $G_i \in \mathbb{R}^{N_G \times 4}$, and a global decoder feature map $F_i \in \mathbb{R}^{H_i \times W_i \times D}$, where $N_T$ and $N_G$ denotes the number of words and selected regions, respectively. $D$ denotes the dimension of embeddings.

The detailed procedure of progressive region regression is described in Algorithm 1. The vision-language multi-head attention layers fuse region and word embeddings and model their interactions as follows:

$$A_i = \frac{E_i^G E_i^G^T}{\sqrt{D}},$$

$$E_{i+1}^G = \text{Softmax}(A_i) E_i^T,$$

$$E_{i+1}^T = \text{Softmax}(A_i^T) E_i^G.$$  \hspace{1cm} (1)

With vision-language semantics, the updated vision-aware word embeddings $E_i^T$ are able to guide region coordinate updates by searching for highly correlated regions around the original one. Specifically, the neighborhood of region $g = (x, y, w, h)$ is defined as a region of size $(N_i^h, N_i^w)$ centered on it, where $N_i^h$ and $N_i^w$ are predefined parameters for the i-th level region regression process. The neighborhood is split to $K \times K$ regions to compute region-word similarities. As shown in Fig. 4, each region embedding is extracted with RoIAlign and then averaged pooling from $F_i$.

With different response scores to words, neighbor regions aggregate context information to the central one. The coordinate update for the central region is in the form of weighted summation of coordinates of its neighbor center points, as shown in Eq. (2):

$$\Delta x = \sum_{j=0}^{K^2-1} M_j N_j^h (j \mod K - \lfloor \frac{K}{2} \rfloor),$$

$$\Delta y = \sum_{j=0}^{K^2-1} M_j N_j^w (j \mod K - \lfloor \frac{K}{2} \rfloor),$$

$$x' = x + \Delta x, \quad y' = y + \Delta y,$$

$$w' = p_w w, \quad h' = p_h h,$$

where $\lfloor \cdot \rfloor$ is the round down operation. Every region in all levels of the region adaptor has its own $p_w$ and $p_h$, which are set as learnable parameters. $M_j$ is the maximum cosine similarity between the embedding of the $j$-th neighbor region and all word embeddings. The purpose of the last term in the first two lines of Eq. (2) is...
to map the 1D index to a 2D index (e.g., from \{0, 1, ..., 8\} to \{(0, 0), (0, 1), ..., (2, 2)\}).

For each original region \(g_i\), let \(g_i'\) denotes its updated version after the \(i\)-th layer in region regression. We take the average of them as the ground truth and apply the \(L_1\) and GIoU regression loss:

\[
\mathcal{L}_{reg}(g) = \mathcal{L}_{L_1}(g, \bar{g}) + \mathcal{L}_{GIoU}(g, \bar{g}).
\]

3.3. Pre-training Objectives with SLAN

SLAN is pre-trained on image-text pairs and learns fine-grained region-word alignments with the supervision from three common losses.

Image-Text Matching Loss (ITM) predicts whether a given image-text pair is positive or not, which can be viewed as a binary classification problem. The visual and textual tokens \((T_v, T_t)\) are concatenated and sent to a linear layer \(f_c\). The ITM loss is formalized as follows:

\[
\mathcal{L}_{itm}(I, T) = H(f_c(cat(T_v, T_t)), y_{v,t}),
\]

where \(y_{v,t}\) denotes the matching relation (1 for matched and 0 for unmatched), and \(H\) is the cross-entropy loss for classification. We directly select positive pairs from the dataset and build hard negative samples with batch sampling, following ALBEF [24].

Image-Text Contrastive Loss (ITC) ensures that visual and textual embeddings share the same semantic space and the positive (matched) image-text pairs are pulling closer than negative (unmatched) ones. We use two queues \(I_q, T_q\) to save the latest visited image and text samples. For each image-text pair \((I, T)\), the softmax-normalized vision-language similarity is computed as:

\[
p_{21}(I, T, T_q) = \frac{\exp(sim(T_v, T_t)/\tau)}{\sum_{T'_v \in T_q} \exp(sim(T_v, T'_t)/\tau)},
\]

\[
p_{21}(T, I, I_q) = \frac{\exp(sim(T_t, T_v)/\tau)}{\sum_{T'_t \in I_q} \exp(sim(T_t, T'_v)/\tau)},
\]

where \(\tau\) is a temperature parameter and \(sim(\cdot)\) measures vision-language similarity, which is implemented by the dot product between the image and text embeddings. Following ALBEF [24], we compute ITC loss as:

\[
\mathcal{L}_{ite}(I, T) = -log(p_{21}(I, T, T_q)) - log(p_{21}(T, I, I_q)).
\]

Language Modeling Loss (LM) encourages the model to predict masked words with context information. We randomly mask 15% text tokens and apply the masked language modeling loss as follows:

\[
\mathcal{L}_{lm}(I, T) = H(p_{mask}(T_v, T_t), y_{mask}),
\]

where \(y_{mask}\) denotes the masked word to predict and \(p_{mask}(I, T)\) is its predicted probability. \(\mathcal{L}_{ds}\) is the downstream loss, which is computed by the sum of previous three losses.

\[
\mathcal{L}_{ds}(I, T) = \mathcal{L}_{itm}(I, T) + \mathcal{L}_{ite}(I, T) + \mathcal{L}_{lm}(I, T).
\]

The full pre-training objective is the combination of the downstream loss and our constraint on progressive region regression, computed as follows:

\[
\mathcal{L} = \mathcal{L}_{ds} + \mathcal{L}_{reg}.
\]

\(\mathcal{L}_{reg}\) denotes the summation of the regression loss in Equ. (3) for all regions. The model is supervised by \(\mathcal{L}\) during training.

4. Experiments

SLAN is first pre-trained on a combined dataset of 14M image-text pairs from five datasets: COCO [28], Visual Genome [19] (excluding COCO images), Conceptual Captions [5], Conceptual [5], and SBU Captions [29]. We evaluate SLAN by comparing it to other state-of-the-art cross-modal methods on several downstream tasks. We also conduct extensive ablation studies to investigate how each component of SLAN influences the performance.

4.1. Implementation Details

We choose BERT\(_{base}\) [18] as our text encoder, which is initialized from HuggingFace [39]. For the vision encoder, we explore four design choices: one CNN-based model (i.e., ResNet50) and three transformer-based models (i.e., ViT-Base, ViT-Large and ViT-Huge), which are all random initialized. As for the neighbour size for each region adaptor, we use a ratio \(r_i\) to denote them: \((N^h_i, N^w_i) = (r_iH_i, r_iW_i)\), where \(r_1, r_2, r_3 = 1, 0.5, 0.25\), respectively. We pre-train SLAN for 20 epochs. For different choices of the vision encoder, the batch size is set to 1280, 960, 640 for ResNet50, ViT-Base, ViT-Large and ViT-Huge, respectively. The AdamW optimizer is adopted with an initial learning rate of 3e-4, and the learning rate is linearly decayed to 0. We resize the input images to \(224 \times 224\).

4.2. Comparison on Downstream Tasks

We compare SLAN with other state-of-the-art methods on five challenging vision-language understanding tasks, including image-text retrieval, image captioning, visual question answering, natural language visual reasoning, zero-shot video-text retrieval. We also generalize SLAN to two localization tasks: object detection and phrase grounding. The default vision encoder is ViT-Huge, if not specified.
4.2.1 Image-Text Retrieval

Given an image, the retrieval task expects to retrieve the corresponding text from the text gallery through the input image, and vice versa. We evaluate our method on Flickr30k [30] under zero-shot and fine-tune settings with Karpathy split and the performance is evaluated in terms of Recall@k. The comparative results are shown in Tab. 2. Specifically, on the same pre-training setting, SLAN outperforms BLIP [23] by 3.3% in average recall@1 on COCO.

4.2.2 Image Captioning

Given an input image, the captioning task generates a sentence description to describe the image in detail. We use COCO Karpathy split to fine-tune and evaluate. SLAN outperforms most existing methods under this efficient setting, as shown in Tab. 3.

4.2.3 Visual Question Answering

Visual Question Answering (VQA) [1] requires the model to predict an answer from an image-question pair. We follow [23] and treat VQA as an open-ended question-generation task. We fuse the image embedding with the question embedding and send them to the question decoder to get the result. As shown in Tab. 3, SLAN achieves higher performance than Beit-3 on the VQAv2 test-dev and test-standard (test-std) splits, which adopts a larger vision backbone and requires more pre-training data.

4.2.4 Natural Language Visual Reasoning

Natural Language Visual Reasoning (NLVR2) [33] measures whether a sentence describes a pair of images. We extract the image and text embeddings from the image-text input, which are then fused with a cross-attention layer. We use a binary classification module to predict their relations. SLAN surpasses most existing methods by a large margin, and achieves comparable performance with Beit-3, showing the importance of learning fine-grained vision-language alignments.
<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Pretrain Data (M)</th>
<th>Object Detection (COCO)</th>
<th>Phrase Grounding (Flickr30k)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Image-Text</td>
<td>Region-Word</td>
<td>Zero-shot</td>
</tr>
<tr>
<td>DETR [4]</td>
<td>ResNet50</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>MDETR [17]</td>
<td>ResNet101</td>
<td>0.2</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>GLIP [25]</td>
<td>Swin-Large</td>
<td>24</td>
<td>3</td>
<td>49.8</td>
</tr>
<tr>
<td>GLIPv2 [46]</td>
<td>Swin-Huge</td>
<td>16</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>Beit-3 [37]</td>
<td>ViT-Giant</td>
<td>21</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>ViT-Base</td>
<td>14</td>
<td>0</td>
<td>46.9</td>
</tr>
<tr>
<td></td>
<td>ViT-Large</td>
<td>14</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>ViT-Huge</td>
<td>14</td>
<td>0</td>
<td>48.5</td>
</tr>
</tbody>
</table>

Table 5. Comparison on two localization tasks: object detection on COCO and phrase grounding on Flickr30k. The pre-training data includes image-text pairs and word-specific region annotations. We evaluate both the zero-shot and fine-tune settings on object detection. We use Recall@k scores to evaluate the phrase grounding task.

<table>
<thead>
<tr>
<th>Trainable Region Proposal</th>
<th>Adaptor Number</th>
<th>COCO</th>
<th>Flickr30k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>TR@1</td>
<td>IR@1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TR@1</td>
<td>IR@1</td>
</tr>
<tr>
<td>✗</td>
<td>0</td>
<td>68.5</td>
<td>53.8</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>69.1</td>
<td>53.8</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>70.0</td>
<td>57.2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>70.8</td>
<td>57.5</td>
</tr>
<tr>
<td>✓</td>
<td>3</td>
<td>72.1</td>
<td>58.3</td>
</tr>
</tbody>
</table>

Table 6. Ablations on the trainable region proposal module and region adaptor in SLAN. ✗ in the first column denotes applying a frozen region proposal module and no self-locator. TR@1 and IR@1 denote recall@1 of image to text and text to image retrieval, respectively. To evaluate the effect of the self-locator against a frozen region proposal module, we load the weights pre-trained on COCO detection task and compare it with our method (Row 1 vs. 2). The remaining experiments are trained from scratch. ViT-Base is used as the vision encoder.

### 4.2.5 Zero-shot Video-Text Retrieval

Besides the image-text tasks mentioned above, SLAN can generalize to the video-text retrieval task. We randomly select \( m \) frames from the video input and concatenate them to get an image-text sequence, which are then directly fed into our image-text retrieval model. As shown in Tab. 4, SLAN achieves comparable performance to the other methods, demonstrating the vision-language knowledge learned in SLAN is semantic-rich.

### 4.2.6 Localization Tasks

We conduct experiments on two localization tasks: object detection on COCO, and phrase grounding on Flickr30k. For the text input in the object detection task, we use a prompt composed of concatenated labels from COCO (e.g., “detect: person, bicycle, car, ... , toothbrush”). We adopt the output from the last layer of the region adaptor. Tab. 5 shows exciting performance of SLAN on localization tasks. For example, in the task of object detection with ViT-Base as the backbone, SLAN achieves comparable results to GLIP requiring a larger backbone and 3M gold data. Though not designed for localization tasks, SLAN with ViT-Huge as backbone outperforms almost all comparative methods.

### 4.3. Ablation Study

#### 4.3.1 Effectiveness of Self-locator

**Importance of learnable region proposal module.** As shown in Tab. 6, the 1st row represents replacing self-locator with a frozen detector pre-trained on the COCO detection task, and the 2nd row is our learnable region proposal module. We do not initialize the region proposal module with pre-trained weights, but only fine-tune them on the downstream task’s datasets. Our method improves on average about 0.5% and 2% on COCO and Flickr30k’s image-to-text and text-to-image retrieval tasks, respectively.

**Number of region adaptors for region regression.** The region adaptor performs progressive regression on the regions outputted by the region proposal module to provide more accurate region localization for vision-language understanding tasks. As shown in Tab. 6, when the number of region adaptors increases from 0 to 3, the retrieval performance can be significantly improved by an average of more than 3%.


<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Params(M)</th>
<th>FLOPs(G)</th>
<th>COCO TR@1</th>
<th>IR@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLIP</td>
<td>ViT-Base</td>
<td>370</td>
<td>558</td>
<td>81.9</td>
<td>64.3</td>
</tr>
<tr>
<td>BLIP</td>
<td>ViT-Large</td>
<td>810</td>
<td>1594</td>
<td>82.4</td>
<td>65.1</td>
</tr>
<tr>
<td>Coca</td>
<td>ViT-Giant</td>
<td>2100</td>
<td>4103</td>
<td>83.0</td>
<td>65.5</td>
</tr>
<tr>
<td>Beit-3</td>
<td>ViT-Giant</td>
<td>1900</td>
<td>-</td>
<td>84.8</td>
<td>67.2</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet50</td>
<td>322</td>
<td>324</td>
<td>85.1</td>
<td>68.9</td>
</tr>
</tbody>
</table>

Table 8. Comparison on number of parameters and FLOPs on the vision-language retrieval task. The FLOPs is calculated with an input image resolution of 384x384. “Backbone” denotes the vision encoder.

Region filter for saliency prediction. Tab. 7 illustrates how the region filter affects the performance on COCO and Flickr30k retrieval tasks. Learnable region proposal module is trained from scratch and the number of region adaptors is set to 3. The first two rows show that when the regions are sorted by their saliency scores and only selected a certain number (top K), we can achieve an performance gain of ~ 2% on each dataset. When using in combination with saliency score threshold, our region filter is able to remove redundant regions that negatively affect vision-language adaptation and achieves even higher performance.

4.3.2 Computational Cost

Tab. 8 shows the comparison on computational cost of SLAN and other state-of-the-art methods. As can be seen, SLAN has the smallest amount of parameters and FLOPs for that in this experiments our vision backbone is a relatively lightweight ResNet50. However, our retrieval performance significantly outperforms other methods. We believe that the above phenomena demonstrate the efficiency and effectiveness of our proposed SLAN.

4.4. Visualization Analysis

4.4.1 Text-guided Region Adaptation

As shown in Fig. 5, our region adaptor produces text-specific results with relatively high confidence. When we change the detailed description of the sentence, e.g., “a man in a red coat” to “a man in black pants”, the interesting phenomenon is that the attention regions of our self-locator are also shifted accordingly with relatively high confidence.

4.4.2 Coarse-to-fine Region Adaptation

To verify the calibration effect of region adaptation, we visualize an image with its text in Fig. 6. Model locates more accurate regions of interest with higher similarity scores after three levels of region adaptor. It shows that our self-locator can hierarchically refine the relevant regions corresponding to the provided words.

5. Conclusions and Future Work

In this paper, we introduce the Self-Locator Aided Network (SLAN), which leverages a self-locator to adapt the proposed regions for vision-language alignments without the need for extra grounding (region-to-word) annotations. We aim to further investigate and optimize the self-locator’s performance for various localization applications.

Acknowledgements. This research was supported by the NSF (NO. 62225604, 62176130) and the Fundamental Research Funds for the Central Universities (Nankai University, 070-63233089). The Supercomputing Center of Nankai University supports computation.
References


[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Int. Conf. Mach. Learn., 2021. 1, 6


[34] Rui Sun, Xuezhi Cao, Yan Zhao, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. Multi-modal knowledge graphs for recommender systems. In CIKM, 2020. 1


