Linguistic More: Taking a Further Step toward Efficient and Accurate Scene Text Recognition

Boqiang Zhang, Hongtao Xie*, Yuxin Wang, Jianjun Xu, Yongdong Zhang
University of Science and Technology of China, Hefei, China
{cyril, wangyx58, xujj1998}@mail.ustc.edu.cn, {htxie, zhyd73}@ustc.edu.cn

Abstract
Vision model have gained increasing attention due to their simplicity and efficiency in Scene Text Recognition (STR) task. However, due to lacking the perception of linguistic knowledge and information, recent vision models suffer from two problems: (1) the pure vision-based query results in attention drift, which usually causes poor recognition and is summarized as linguistic insensitive drift (LID) problem in this paper. (2) the visual feature is suboptimal for the recognition in some vision-missing cases (e.g. occlusion, etc.). To address these issues, we propose a Linguistic Perception Vision model (LPV), which explores the linguistic capability of vision model for accurate text recognition. To alleviate the LID problem, we introduce a Cascade Position Attention (CPA) mechanism that obtains high-quality and accurate attention maps through step-wise optimization and linguistic information mining. Furthermore, a Global Linguistic Reconstruction Module (GLRM) is proposed to improve the representation of visual features by perceiving the linguistic information in the visual space, which gradually converts visual features into semantically rich ones during the cascade process. Different from previous methods, our method obtains SOTA results while keeping low complexity (92.4% accuracy with only 8.11M parameters). Code is available at https://github.com/CyrilSterling/LPV.

1 Introduction
Scene Text Recognition (STR) is a meaningful task in computer vision that aims to understand the textual information from the cropped image of natural scenes [Long et al., 2021; Shi et al., 2016a; Fang et al., 2021; Xu et al., 2022; Sheng et al., 2019]. Due to the lack of language modal information of other perception tasks, STR is widely used in Visual Questions and Answers (VQA), automatic pilots, etc.

*Corresponding Author

Figure 1: Our motivation. (a) The basic structure of previous vision-only models with an attention-based decoder and a visual query to decode the characters, which has the problem of attention drift. (b) The visualization of the dot similarity between the query vectors at each position of ABINet [Fang et al., 2021].

eeling, a CTC-based [Graves et al., 2006] or attention-based decoder to obtain the predicted characters. The attention-based decoder uses a visual query to decode the position of each character, so it is accurate for arbitrary shape text recognition. In addition, due to its simplicity and effectiveness, the attention-based decoder is currently the mainstream solution for vision model [Wang et al., 2021; Fang et al., 2021; Wang et al., 2022a; Zheng et al., 2021; Wang et al., 2022b]. Such methods have a simple structure that can be efficient in most application scenarios. Though these vision-only methods have achieved promising results, there are still two problems.

The first problem is the attention drift in the attention-based decoders, which is not received enough attention in recent researches. Attention drift is when the area of attention region is not aligned with the target character (Figure. 1 (a)). RobustScanner [Yue et al., 2020] deeply analyzed attention drift and proved that the query vectors in the decoder encode not only context but also positional information. However, this positional information is easily drowned out by the in-
troduction of other information. Note that recent methods use a pure vision-based query, which is fixed when inputting different images. We further visualize the dot similarity between the query vectors at each position in ABI Net [Fang et al., 2021]. As shown in Figure 1(b), it is observed that the query at each position is similar to that at neighboring positions. This will lead to similar features when decoding the attention map of neighboring characters, thereby causing attention drift. Therefore, we indicate that the attention drift comes from decoding different images with the fixed vision-based query, which is linguistic insensitive. We summarize this issue as the Linguistic Insensitive Drift (LID) problem. Thus, how to eliminate the LID issue and obtain an accurate attention map is the key for robust text recognition.

Another problem is that visual feature is suboptimal for recognition in some vision-missing cases. To solve this problem, recent methods introduce the linguistic knowledge to assist the vision model. However, it is hard for vision model to obtain linguistic information efficiently and accurately. VisionLAN [Wang et al., 2021] designed a masked language-aware module to randomly occlude a character in the training stage which guides the vision model to utilize the linguistic information in text images. But the model introduces additional modules and requires separate pre-training. MGP-STR [Wang et al., 2022a] proposed a multi-granularity prediction strategy to inject information from the language modality into the model in an implicit way. However, this network requires a huge number of parameters. Thus, how to perceive linguistic information with an efficient structure and a simple training strategy is a great challenge for text recognition.

To enhance the linguistic perception of both query and feature in a simple way, we propose a concise Linguistic Perception Vision model (LPV). The pipeline of our LPV is shown in Figure 2. The pipeline mainly consists of two parts: the GLRM branch and the CPA branch. The GLRM branch continuously enhances the features of the input image. In this branch, the visual features $F^0$ are firstly extracted from the backbone. Then, Global Linguistic Reconstruction Module (GLRM) enhances the features of the previous stage $F^{i-1}$ into the features of the current stage $F^i$ using the mask generated by the attention map $A^{i-1}$. In this way, linguistic information is aggregated in GLRM and the visual features can be gradually transformed into semantic-rich features. Meanwhile, the GLRM ensures the simplification of the pipeline and there are no redundant modules. The CPA branch hierarchically optimizes the attention map and the query using the cascade position attention mechanism. Each Position Attention Module (PAM) takes the visual features $F^i$ as input, and obtains the attention map $A^i$ and the features $R^i$ of each character. Note that the prior query of the first PAM is initialized to 0, which means we have no prior to each character. PAM at $i^{th}$ stage uses $R^{i-1}$ as the prior query. Such an operation can take the recognition result of the previous stage as a priori and re-perform the similarity calculation for the enhanced features to obtain more accurate attention positions. Meanwhile, positional and linguistic information is constantly introduced in the PAM, so as to alleviate the linguistic insensitive drift problem. Compared with previous methods, we have a more concise structure and training strategy, while achieving better performance.

The main contributions of our work are as follows:

- We are the first to point out the attention drift due to lack of linguistic information, which is called Linguistic Insensitive Drift (LID) problem, and propose a Cascade Position Attention mechanism to effectively handle the LID problem.
- We propose a Global Linguistic Reconstruction Module to reconstruct the features of each character by aggregating global linguistic information during the process of sequence modeling. The method does not introduce extra parameters.
- Our method achieves state-of-the-art performance while keeping a very low parameter quantity with a simple end-to-end training strategy.

2 Related Work

2.1 Scene Text Recognition

Scene Text Recognition (STR) has been a significant research term in computer vision. Early methods use a backbone and a sequence modeling network for feature extraction and use a Connectionist Temporal Classification (CTC) [Graves et al., 2006] decoder or attention decoder for prediction [Qiao et al., 2020; Lyu et al., 2019a]. CTC-based decoder aims to maximize the probability of all the paths for final prediction, while attention-based decoder aims to localize the position of each character by attention mechanism. To further extract linguistic information of the visual predictions, SRN [Yu et al., 2020] proposed a language model to learn the relationship between each character. ABI Net [Fang et al., 2021] further proposed a stronger bi-directional language model for autonomous linguistic modeling. We believe that a powerful recognizer must have the ability of contextual linguistic modeling, but explicit language models have a large number of parameters, which severely limits recognition efficiency.

Recently, the simplicity of model reasoning has been emphasized. Considering the CTC-based decoder has an advantage in speed while the attention-based decoder has an advantage in precision, GTC [Hu et al., 2020] used a powerful attention-based decoder to guide the training of a CTC-based decoder. MGP-STR [Wang et al., 2022a] used ViT as the backbone to achieve high performance, which proved that the structure of ViT is applicable to STR. Further, SVTR [Du et al., 2022] proposed a faster and more lightweight backbone for STR task. We think that in order to design a more powerful recognizer, the model must have linguistic modeling capability while keeping the structure simple. Thus, we propose the Global Linguistic Reconstruction Module, which can aggregate the contextual linguistic information during the process of sequence modeling. Such a design ensures the simplicity of the model.

2.2 Attention Drift

The visual attention drift problem in STR refers to the fact that when an attention-based decoder is used, the attention region of the decoder cannot be accurately aligned with the
target character. [Cheng et al., 2017] first identified this problem and proposed a Focusing Attention Network (FAN) that is composed of an attention network for character recognition and a focusing network to adjust the attention drift. RobustScanner [Yue et al., 2020] deeply investigated the decoding process of the attention-based decoder and empirically find that a character-level sequence decoder utilizes not only context information but also positional information. They further suggested that the drowning of position information leads to attention drift problems. Using the above analysis, they solve this problem by a position enhancement branch to introduce position information. We further point out that attention drift comes from decoding different images with a linguistic insensitive query, which also lacks positional information. Based on this, we propose a Cascade Position Attention mechanism to solve this problem, which has a concise framework and does not introduce extra modules.

3 Proposed Method

In this section, we first detail the pipeline of proposed method in Sec. 3.1, and then we introduce Cascade Position Attention and Global Linguistic Reconstruction Module in Sec. 3.2 and Sec. 3.3 respectively.

3.1 Pipeline

The pipeline of our LPV is shown in Figure 2. We can view the pipeline as two branches. Given an input image of size \(H \times W \times 3\), the features \(F^i\) are obtained by the GLRM branch, which continuously enhances the features to obtain long-distance contextual linguistic information using proposed GLRM. Meanwhile, in the CPA branch, \(F^i\) are fed into the \(i^{th}\) Position Attention Module (PAM) to get the attention map \(A^i\) and the feature \(R^i\) of each character. The CPA branch constantly rectifies the recognition results to alleviate linguistic insensitive drift using a linguistic-sensitive query. Note that the parameters in each PAM are NOT shared.

![Figure 2: The pipeline of our LPV. The pipeline mainly contains two branches: GLRM branch and CPA branch. GLRM branch continuously enhances the feature using proposed GLRM. CPA branch takes the feature as input and hierarchically decodes the attention map and feature of each character. CE Loss means the cross-entropy loss.](image)

3.2 Cascade Position Attention

The Cascade Position Attention mechanism hierarchically optimizes the recognition result using the enhanced feature \(F^i\) in each stage and outputs the attention map of each character.

As shown in Figure 3, a cross-attention mechanism is utilized to transcribe visual features into character sequences. Specifically, the attention map \(A^i \in \mathbb{R}^{T \times HW}\) and the features \(R^i \in \mathbb{R}^{T \times E}\) of each character is calculated by the queries, keys, and values as Eq. 1, where \(T\) is the maximum length of the character sequence. The prediction results \(Y^i \in \mathbb{R}^{T \times C}\) can be further obtained by a classification head (e.g. FC Layer), where \(C\) indicates the number of character classes. \(P(\cdot)\) is the classification head.

\[
A^i = softmax(K^iQ^iT/\sqrt{E})
\]

\[
R^i = A^iV^i
\]

\[
Y^i = softmax(P(R^i))
\]  

Concretely, \(K^i = G(F^i) \in \mathbb{R}^{HW \times E}\), where \(G(\cdot)\) is imple-
model learning linguistic knowledge, we design a novel way that reconstructs the features of each character by masking each character. To achieve this, PMG transforms the attention map $A^{i-1}$ into a parallel mask $M \in \mathbb{R}^{HW \times HW}$ as Eq. 3, where $U(x)$ is the unit stage function which takes the value of 1 for $x \geq 0$ and 0 for $x < 0$. $t$ is the threshold of foreground and background, which is set to 0.05 in our experiments. $\odot$ is matrix multiplication.

$$M^p = U(A^{i-1} - t)^T \odot U(A^{i-1} - t)$$

Then, the attention operation inside multi-head self-attention blocks can be formalized as follows:

$$[Q, K, V] = F^{i-1} W$$

$$F^i = softmax(QK^T \sqrt{E} + M)V$$

By using such a mask, the tokens in one character can not see the tokens in the same character during the self-attention operation, which means that features within each character region are reconstructed from features other than that character. Benefiting from such a design, the visual features are gradually transformed into semantic-rich features during the cascade stage.

Compared with BERT [Devlin et al., 2018] and Vision-LAN [Wang et al., 2021], though all approaches mask out the information in a certain time step, there are two differences: 1) BERT and VisionLAN mask the tokens of the input features, which leads to loss of origin features. But GLRM masks the tokens in the self-attention operation, which can ensure self-attention to model the global linguistic information. Meanwhile, the origin local features of each character are not lost due to the shortcut of the transformer encoder; 2) BERT and VisionLAN can only mask one character in a forward process, it can only guide the model to learn linguistic knowledge, but GLRM can mask all characters in a parallel way, which can reconstruct the features of each character and obtain an enhanced feature.

### 3.4 Training Objective

The final objective function of the proposed method is formulated in Eq. 5. $N$ is the number of cascade stages and $Y^i$ is the prediction at the $i$th stage. $g_t$ is the ground truth. $T$ is the max length of the character sequence which we set to 25 in our experiments.

$$L = -\frac{1}{NT} \sum_{i=0}^{N} \sum_{j=0}^{T} \log(P(Y^i|g_t))$$

### 4 Experiment

#### 4.1 Datasets

For fair comparison, we conduct experiments following the setup of [Wang et al., 2022a; Fang et al., 2021]. We use...
Table 1: Results on IC13, SVT, IIIT5k, IC15, SVTP and CUTE datasets. Following [Fang et al., 2021; Wang et al., 2022a], all the results are under NONE lexicon. The speed is the inference time on one NVIDIA 2080Ti GPU averaged over 1000 English image text.

<table>
<thead>
<tr>
<th>Method</th>
<th>IC13</th>
<th>SVT</th>
<th>IIIT5k</th>
<th>IC15</th>
<th>SVTP</th>
<th>CUTE</th>
<th>AVG</th>
<th>#Params (M)</th>
<th>Speed (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRNN [Shi et al., 2016b]</td>
<td>91.9</td>
<td>81.6</td>
<td>82.9</td>
<td>69.4</td>
<td>70.0</td>
<td>65.5</td>
<td>78.6</td>
<td>8.3</td>
<td>-</td>
</tr>
<tr>
<td>ASTER [Shi et al., 2018]</td>
<td>91.8</td>
<td>89.5</td>
<td>93.4</td>
<td>76.1</td>
<td>78.5</td>
<td>79.5</td>
<td>86.7</td>
<td>27.2</td>
<td>-</td>
</tr>
<tr>
<td>SEED [Qiao et al., 2020]</td>
<td>92.8</td>
<td>89.6</td>
<td>93.8</td>
<td>80.0</td>
<td>81.4</td>
<td>83.6</td>
<td>88.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RobustScanner [Yue et al., 2020]</td>
<td>94.8</td>
<td>88.1</td>
<td>95.3</td>
<td>77.1</td>
<td>79.5</td>
<td>90.3</td>
<td>88.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TextScanner [Wan et al., 2020]</td>
<td>92.9</td>
<td>90.1</td>
<td>93.9</td>
<td>79.4</td>
<td>84.3</td>
<td>83.3</td>
<td>88.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SRN [Yu et al., 2020]</td>
<td>95.5</td>
<td>91.5</td>
<td>94.8</td>
<td>82.7</td>
<td>85.1</td>
<td>87.8</td>
<td>90.4</td>
<td>54.7</td>
<td>-</td>
</tr>
<tr>
<td>VisionLAN [Wang et al., 2021]</td>
<td>95.7</td>
<td>91.7</td>
<td>95.8</td>
<td>83.7</td>
<td>86.0</td>
<td>88.5</td>
<td>91.2</td>
<td>32.8</td>
<td>21.73</td>
</tr>
<tr>
<td>ABINet [Fang et al., 2021]</td>
<td>97.4</td>
<td>93.5</td>
<td>96.2</td>
<td>86.0</td>
<td>89.3</td>
<td>89.2</td>
<td>92.3</td>
<td>36.7</td>
<td>46.86</td>
</tr>
<tr>
<td>MGP-Small [Wang et al., 2022a]</td>
<td>96.4</td>
<td>93.5</td>
<td>95.3</td>
<td>86.1</td>
<td>87.3</td>
<td>87.9</td>
<td>92.0</td>
<td>52.6</td>
<td>-</td>
</tr>
<tr>
<td>MGP-Base [Wang et al., 2022a]</td>
<td>97.3</td>
<td>94.7</td>
<td>96.4</td>
<td>87.2</td>
<td>91.0</td>
<td>90.3</td>
<td>93.3</td>
<td>148.0</td>
<td>-</td>
</tr>
<tr>
<td>SVTR-Tiny [Du et al., 2022]</td>
<td>96.3</td>
<td>91.6</td>
<td>94.4</td>
<td>84.1</td>
<td>85.4</td>
<td>88.2</td>
<td>90.8</td>
<td>6.03</td>
<td>4.11</td>
</tr>
<tr>
<td>SVTR-Small [Du et al., 2022]</td>
<td>95.7</td>
<td>93.0</td>
<td>95.0</td>
<td>84.7</td>
<td>87.9</td>
<td>92.0</td>
<td>91.6</td>
<td>10.3</td>
<td>4.81</td>
</tr>
<tr>
<td>SVTR-Base [Du et al., 2022]</td>
<td>97.1</td>
<td>91.5</td>
<td>96.0</td>
<td>85.2</td>
<td>89.9</td>
<td>91.7</td>
<td>92.3</td>
<td>24.6</td>
<td>5.80</td>
</tr>
<tr>
<td>LPV-Tiny (Ours)</td>
<td>96.7</td>
<td>92.9</td>
<td>96.3</td>
<td>86.4</td>
<td>86.7</td>
<td>90.6</td>
<td>92.5</td>
<td>8.11</td>
<td>5.17</td>
</tr>
<tr>
<td>LPV-Small (Ours)</td>
<td>96.8</td>
<td>93.7</td>
<td>96.7</td>
<td>87.1</td>
<td>89.8</td>
<td>92.4</td>
<td>93.3</td>
<td>13.99</td>
<td>5.77</td>
</tr>
<tr>
<td>LPV-Base (Ours)</td>
<td><strong>97.6</strong></td>
<td><strong>94.6</strong></td>
<td><strong>97.3</strong></td>
<td><strong>87.5</strong></td>
<td><strong>90.9</strong></td>
<td><strong>94.8</strong></td>
<td><strong>94.0</strong></td>
<td><strong>35.13</strong></td>
<td><strong>7.41</strong></td>
</tr>
</tbody>
</table>

4.3 Comparisons with State-of-the-Arts

We compare our method with previous state-of-the-art methods on 6 benchmarks in Table. 1. Our model shows significant performance in both regular (IC13, SVT and IIIT5k) and irregular (IC15, SVTP and CUTE) datasets while keeping a very low parameter quantity. Notably, LPV-Tiny has already outperformed most of the state-of-the-art methods with only 8.11M parameters while LPV-Small and LPV-Base obtain the performance of 93.3% and 94.0% with only 13.99M and 35.13M parameters respectively. For inference time, LPV-Tiny only needs 5.17ms, which is faster than most of the existing methods.

Many previous works, such as SRN [Yu et al., 2020], ABINet [Fang et al., 2021], VisionLAN [Wang et al., 2021], and MGP [Wang et al., 2022a] tried to introduce linguistic knowledge to assist recognition. Compared to them, LPV shows the best performance on all datasets. This result implies that our cascade position attention mechanism and GLRM are effective. Additionally, compared with SVTR, our tiny, small, and base model obtains 1.6%, 1.7%, and 1.7% improvement respectively.

4.4 Ablation Study

The Effectiveness of Cascade Position Attention

We propose the Cascade Position Attention (CPA) mechanism to alleviate the linguistic insensitive drift problem. To prove the effectiveness of CPA, we perform ablation from two aspects.

From the aspect of model performance, we conduct several experiments to evaluate the effect of the number of stages $N$ in Table. 2. Especially, $N = 1$ means no extra stage to optimize the recognition result. The first row of Table 2 is the baseline with a CTC-based decoder instead of attention-based decoder. From the statistics we can conclude: 1) Our
model with 3 stages outperforms 1.581% more improvement.
2) Due to the concise structure of the hierarchical optimization strategy, the increase of stages will result in great gains in average accuracy while little increase in parameter quantity.
3) As the number of stages increases gradually, performance improvement is limited. For the trade-off between parameter quantity and accuracy, we choose 3 stages in our model.

From the aspect of attention drift, we further visualize it. In the position attention mechanism, the query guides the decoder to find the position of each character. As described in Sec. 1, the high similarity of queries between neighboring locations leads to the problem of attention drift. Based on LPV-Tiny, we visualize the similarity between query vectors at different positions in Figure. 5. Due to stages 2 and 3 having a linguistic-sensitive query that is different when inputting different images, we calculate the average similarity with all images in IC15 [Karatzas et al., 2015] of each sequence length. As shown in Figure 5, the query in the first stage has no position and linguistic prior about the input image so it does not have a centralized similarity. In stages 2 and 3, the queries consist of a linguistic prior query Q\textsubscript{pri}, and the position encoding is introduced again to enhance position sensitivity. Therefore, the similarity is centered in the diagonal, which means the position of each character is more certain. When decoding characters, the feature similarity of neighboring characters is reduced, so the attention drift is mitigated. Note that the similarity in stage 3 is more concentrated than that in stage 2 due to the stronger prior and more position information. Additionally, the difference is even more pronounced with long text, because attention drift is more likely to occur in the case of long text.

The Effectiveness of GLRM
As described in Sec. 3.3, we argue that the input features F\textsuperscript{i} of each stage can not be the same and needs to be dynamically adjusted, so a sequence modeling network is necessary. To prove this inference, we first use a simple transformer encoder as the sequence modeling network to obtain dynamic features. For fair comparison, we place the transformer encoder before the CPA decoder to fix the features and keep the same parameter quantity. As shown in Table. 3, the performance is not good (91.763% vs 92.012%) when we place the transformer layer before the CPA and input the same features into each stage.

Furthermore, to acquire linguistic knowledge, we propose GLRM as the sequence modeling network which uses a parallel mask to enhance the feature and obtain the contextual linguistic information. As shown in Table. 3, for LPV-Tiny and LPV-Small, the proposed mask obtains 0.469% and 0.539% improvement on average accuracy respectively.

The Layer Number of GLRM
Our GLRM consists of a Parallel Mask Generator and L×Masked Transformer Encoder. To determine the number of layers L, we conduct several experiments. From the results in Table. 4 we can observe: 1) More layers in GLRM can provide stronger contextual modeling capacity and obtain higher performance. 2) We can obtain 0.69% improvement when L increases from 1 to 2, which is greatly larger than 0.234% when L goes from 2 to 3. That is because the masked transformer encoder can only model the area around each character at a shallow layer, and gradually model the global feature as it moves deeper. This conjecture can be verified by the visualization of the attention map in the masked transformer encoder. We calculate the average attention map of the pixels in the area masked and show the visualization in Figure. 6. From the attention map, we can find that the attention in the first layer of the first GLRM is limited around each character.
because there is no global feature in the input. When it goes deep, the attention area goes global.

Finally, considering the total parameter quantity, we set $L$ to 2 in LPV-Tiny and 3 in LPV-Small and LPV-Base.

4.5 The Qualitative Analysis

GLRM in Subword Perception

From the visualization in Figure 6, we can further analyze the attention area. As we all know, there are some sub-words that occur frequently in words (e.g., 'ing', 'pri', 'mer', 'tion', etc). Such knowledge can assist the model to obtain a more accurate result when the visual clue is confused. Our GLRM guides the model to reconstruct the feature of each character using the feature of other characters so it will be sensitive to the sub-words. As shown in Figure 6 (a), the sub-words 'din' and 'ing' pay attention to themselves individually. This demonstrates the ability of our GLRM to learn contextual linguistic knowledge.

Linguistic Inensitive Drift Problem

From the visualization in Figure 7, we can further analyze the attention area. As we all know, there are some sub-words that occur frequently in words (e.g., 'ing', 'pri', 'mer', 'tion', etc). Such knowledge can assist the model to obtain a more accurate result when the visual clue is confused. Our GLRM guides the model to reconstruct the feature of each character using the feature of other characters so it will be sensitive to the sub-words. As shown in Figure 6 (a), the sub-words 'din' and 'ing' pay attention to themselves individually. This demonstrates the ability of our GLRM to learn contextual linguistic knowledge.

4.5 The Qualitative Analysis

GLRM in Subword Perception

From the visualization in Figure 6, we can further analyze the attention area. As we all know, there are some sub-words that occur frequently in words (e.g., 'ing', 'pri', 'mer', 'tion', etc). Such knowledge can assist the model to obtain a more accurate result when the visual clue is confused. Our GLRM guides the model to reconstruct the feature of each character using the feature of other characters so it will be sensitive to the sub-words. As shown in Figure 6 (a), the sub-words 'din' and 'ing' pay attention to themselves individually. This demonstrates the ability of our GLRM to learn contextual linguistic knowledge.

Linguistic Inensitive Drift Problem

From the visualization in Figure 7, we can further analyze the attention area. As we all know, there are some sub-words that occur frequently in words (e.g., 'ing', 'pri', 'mer', 'tion', etc). Such knowledge can assist the model to obtain a more accurate result when the visual clue is confused. Our GLRM guides the model to reconstruct the feature of each character using the feature of other characters so it will be sensitive to the sub-words. As shown in Figure 6 (a), the sub-words 'din' and 'ing' pay attention to themselves individually. This demonstrates the ability of our GLRM to learn contextual linguistic knowledge.

5 Conclusion

This paper first notices the Linguistic Inensitive Drift (LID) problem and analyzes the linguistic perception of the model. To find an efficient and accurate method, LPV is proposed to enhance the linguistic information of both query and feature (Linguistic More). To be specific, LPV introduces CPA to obtain an accurate attention map by using linguistic-sensitive query instead of visual query, and designs GLRM to aggregate the global linguistic information to enhance the visual feature. Compared with previous methods, our LPV is able to take a further step toward efficient and accurate recognition, which obtains dominant recognition performance while maintaining a concise pipeline. We believe that LPV will inspire recent works in simple network design and efficient linguistic perception, and we will further explore its potential in the future.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2022YFB3104700), the National Nature Science Foundation of China (62121002, 62022076, U1936210, 62232006).
References


