Supplementary: Introducing Language Guidance in Prompt-based Continual Learning

Muhammad Gul Zain Ali Khan\textsuperscript{1,2} Muhammad Ferjad Naeem\textsuperscript{3} Luc Van Gool\textsuperscript{3} Didier Stricker\textsuperscript{1,2} Federico Tombari\textsuperscript{4,5} Muhammad Zeshan Afzal\textsuperscript{1,2}

\textsuperscript{1}RPTU\textsuperscript{2} DFKI\textsuperscript{3} ETH Zürich\textsuperscript{4} TUM\textsuperscript{5} Google

1. Experiment Details

We provide experimental details of all experiments in this section. We implement LGCL using PyTorch on Ubuntu 20.0 workstation. We conduct all experiments on A100-40GB GPUs. We use a constant learning rate of 0.005 using Adam Optimizier \cite{3} with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We use batch size 24 for DualPrompt \cite{6} and batch size 16 for L2P \cite{7}. We resize all input images to 224$\times$224. For L2P, we use 50 epochs for Split-ImageNet-R \cite{6} and 5 epochs for CIFAR-100 \cite{4} datasets. For DualPrompt \cite{6} we use 50 epochs for Split-ImageNet-R and 20 epochs for L2P \cite{7} on Split-CIFAR100 \cite{4} dataset. We set $M = 10$, $L_p = 5$ and $N = 5$ for DualPrompt \cite{6} on Split-CIFAR100 \cite{4} dataset. We set $M = 10$, $L_p = 20$, $L_q = 5$ and $N = 1$ for Dualprompt \cite{6} on Split-ImageNet-R \cite{6} dataset. We set $M = 30$, $L_p = 20$ and $N = 5$ for L2P \cite{7} on Split-ImageNet-R \cite{6}. We use CLIP \cite{5} text encoder “ViT-L/14” for generating language representations ($L_t$ and $L_c$). Specifically, we use “ViT-L/14” for L2P \cite{7} and “ViT-L14/336px” for DualPrompt \cite{6}. Following \cite{7, 6}, we use ViT-B/16 pre-trained backbone.

For Split-ImageNet-R \cite{6}, we set 0.3 loss weight for task-level language guidance loss given in Eq. 3 and 0.7 loss weight for class-level language guidance loss given in Eq. 4. For Split-CIFAR100 \cite{4}, we set 0.4 loss weight for task-level language guidance loss given in Eq. 3 and 0.6 loss weight for class-level language guidance loss given in Eq. 4. For L2P \cite{7}, we set 0.1 loss weight for task-level language guidance loss given in Eq. 3 and 0.9 loss weight for class-level language guidance given in Eq. 4 on Split-CIFAR100 \cite{4} dataset. For Split-ImageNet-R \cite{1}, we set 0.5 loss weight for both task-level language guidance loss given in Eq. 3 and class-level language guidance loss given in Eq. 4 with L2P \cite{7}

2. Evaluation Metrics

We compute Average Accuracy at Task $t$ denoted by $A_t$ and Forgetting at Task $t$ denoted by $F_t$. Let $E_{t,T}$ be the accuracy of task $t$ when evaluated at task $T$. Then $A_t$ is shown below:

$$A_t = \frac{1}{t} \sum_{T}^{t} E_{t,T}$$

Average accuracy represents an average of the accuracy of all the tasks at a given task $t$. However, this does not represent how much the model has forgotten from the previous tasks. Forgetting $F_t$ aims to quantify the catastrophic forgetting of Neural Networks in Continual Learning. Forgetting $F_t$ formulation is given below.

$$F_t = \frac{1}{t-1} \sum_{\tau}^{t-1} max_{\tau' \in \{1,...,t-1\}} (E_{\tau', \tau} - E_{t,\tau})$$

3. Comparison with L2P on Prompt Length and Selection Size

We show ablation on Prompt Length “L” and Selection Size “N” on L2P \cite{7} in Table. 1. For comparison with baseline method L2P \cite{7}, we report average accuracies on Split-CIFAR100 \cite{4}. We show that LGCL outperforms baseline method L2P \cite{7} on most configurations. LGCL achieves the highest accuracy at N=5 and L=20. However, the highest difference in
performance is with N=20 and L=20 where LGCL outperforms baseline method L2P [7] by 1.51%. LGCL is comparable to baseline method L2P [7] where L2P [7] outperforms LGCL with highest difference of 0.8% at N=1 and L=10. We observe that LGCL is fairly robust to changes in prompt configuration since LGCL outperforms baseline method L2P [7] on most configurations and produces comparable results in configurations where L2P [7] outperforms LGCL. We achieve this consistent performance without any addition in the number of parameters.

![Table 1: Ablation on Prompt Length “L” and Selection Size “N” on L2P [7]](image_url)

<table>
<thead>
<tr>
<th>L</th>
<th>N</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77.63/77.01</td>
<td>81.49/82.49</td>
<td>82.92/83.39</td>
<td>83.34/83.15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>82.24/82.88</td>
<td>83.85/83.50</td>
<td>83.90/83.39</td>
<td>83.79/82.84</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>82.48/83.28</td>
<td>83.68/83.18</td>
<td>83.52/83.13</td>
<td>81.98/81.84</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>83.86/83.40</td>
<td>84.01/82.60</td>
<td>82.65/81.10</td>
<td>81.19/79.65</td>
<td></td>
</tr>
</tbody>
</table>

4. Limitations

LGCL does not explore other modalities since LGCL builds on previous prompting based continual learning methods [7, 6]. LGCL assumes that robust VIT [2] based pre-trained feature extractor is available. LGCL also assumes that a pre-trained text encoder that can generate robust text embeddings of the classes is available during training. Furthermore, LGCL explores language modelling on VIT [2] based networks. We leave the exploration of LGCL and language modelling on conventional ConvNet-based networks as future work. While we have shown that Incremental Class setting of continual learning can benefit from language modelling and LGCL, we leave the exploration of LGCL in other domains of continual learning, such as task-agnostic continual learning, as future work.

References


