Improving Visual Prompt Tuning by Gaussian Neighborhood Minimization for Long-Tailed Visual Recognition

Mengke Li[1,2] Ye Liu[1,2] Yang Lu[3] Yiqun Zhang[4] Yiu-ming Cheung[5] Hui Huang[2*]

[1]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), China [2]Shenzhen University, China [3]Xiamen University, China [4]Guangdong University of Technology, China [5]Hong Kong Baptist University, China

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction & Motivation

Problem:

- The imbalanced distribution of long-tailed data impairs generalization performance of Parameter-efficient fine-tuning (PEFT).
- Sharpness-aware minimization (SAM) improves model generalization by flattening minima, but neglects class imbalance and increases computational costs.

Existing SAM-based methods:

- The generalization ability of head and tail classes cannot be enhanced equally.
- Additional computational costs are introduced.

Motivation:

- Flatten the loss landscape to enhance model generalization.
- Utilizing a distribution-independent SAM-based perturbation.

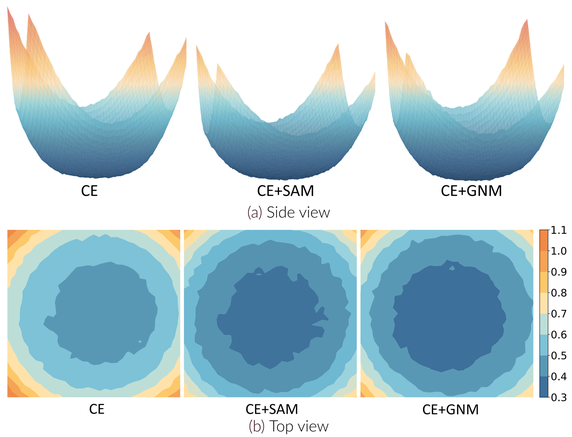

Figure 1. Loss landscape comparison with CE

Overview of GNM-PT:

Prompt Tuning with Gaussian Neighborhood Minimization (GNM-PT) aims to mitigate the presence of sharp minima and enhance the performance of VPT on long-tailed data by optimizing the parameters of VPT using our proposed Gaussian neighborhood minimization (GNM). GNM updates the model using the gradient at the Gaussian neighborhood of current model, which is independent of the data distribution schematically illustrated in Figure 2.

- More details at: http://arxiv.org/abs/2410.21042
- Code: https://github.com/Keke921/GNM-PT
- Contact: limengke@gml.ac.cn; zbdly226@gmail.com.

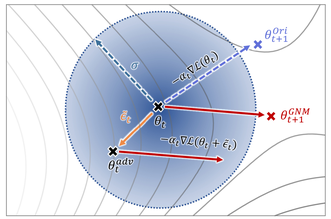

Figure 2. Schematic of optimization direction in GNM.

Table 1. Comparison of Native Execution Time.

Method	Acc. (%)	NET (s)
CE	81.02	39.78
CE+SAM	82.48	72.51
CE+GNM	**82.50**	40.16 (↓ 44.61%)
GCL+DRW	89.58	40.00
GCL+DRW+SAM	89.69	74.36
GCL+DRW+GNM	**90.28**	41.87 (↓ 43.69%)

Prompt Tuning with Gaussian Neighborhood Minimization

Methodology:

Gaussian neighborhood minimization (GNM) introduce the Gaussian neighborhood loss $L_{\mathcal{T}}^{GN}$ on $\mathcal{T}$, which is defined as:

$$L_{\mathcal{T}}^{GN}(\boldsymbol{\theta}) = \mathbb{E}_{\varepsilon_i \in \mathcal{N}(0,\sigma^2)} \left[L_{\mathcal{T}}(\boldsymbol{\theta} + \boldsymbol{\varepsilon}) \right]. \tag{1}$$

Optimizing $L_{\mathcal{T}}^{GN}$ is equivalent to optimizing an upper bound of the distribution $\mathcal{D}$ using the training set $\mathcal{T}$ sampled i.i.d. from $\mathcal{D}$, which is theoretically proven in our paper. When optimizing $L_{\mathcal{T}}^{GN}$, the parameters are updated as follows:

$$\tilde{\boldsymbol{\varepsilon}}_t = \rho_{GNM} \cdot [\varepsilon_i]_{i=1}^k, \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \tag{2}$$

$$\boldsymbol{\theta}_{t+1}^{GNM} = \boldsymbol{\theta}_t - \alpha_t \left(\nabla_{\boldsymbol{\theta}_t} L_{\mathcal{T}}(\boldsymbol{\theta}_t)|_{\boldsymbol{\theta}_t + \tilde{\boldsymbol{\varepsilon}}_t} + \lambda \boldsymbol{\theta}_t \right). \tag{3}$$

ρ_{GNM} represents the radius of the parameter neighborhood for GNM. Figure 2 schematically illustrates a single GNM parameter update.

Detail Analysis:

- GNM is better suited for long-tailed data. Eq. (2) in GNM is a sample-independent manner compared to the perturbation vector in SAM dominated by head classes.
- GNM saves computational overhead compared to SAM. The parameter update in GNM is computationally efficient without an additional forward and backward pass to calculate perturbations.
- GNM can achieve a flat loss landscape for VPT. Fig. 1 and Fig. 3 demonstrate the loss landscape obtained by GNM for VPT is flattened than the original VPT and SAM.

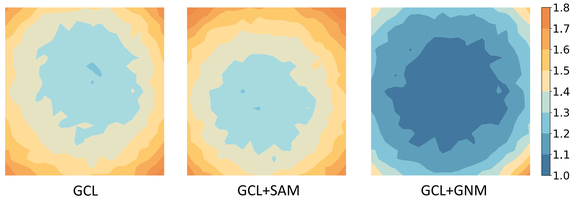

Figure 3. Loss landscape comparison with GCL

Comparison Results

Table 2. Acc. (%) comparison on iNat.

Method	Head	Med	Tail	Overall
DNN-based model (Backbone: ResNet50)				
LWS	72.9	71.2	69.2	70.5
RIDE	76.5	74.2	70.5	72.8
MisLAS	73.2	72.4	70.4	71.6
GCL	-	-	-	72.0
NCL	72.7	75.6	74.5	74.9
GPaCo	-	-	-	75.4
SHIKE	-	-	-	75.4
DNN-based model with SAM				
LDAM+SAM	64.1	70.5	71.2	70.1
CCSAM	65.4	70.9	72.2	70.9
ImbSAM	68.2	72.5	72.9	71.1
MHSA-based model (Backbone: ViT-B/16)				
Supplementary with linguistic data				
VL-LTR	-	-	-	**76.8**
RAC	75.9	80.5	81.1	<u>80.2</u>
Visual-only				
Decoder	-	-	-	59.2
LPT	-	-	79.3	76.1
LiVT	78.9	76.5	74.8	76.1
GNM-PT (ours)	61.5	77.1	79.3	<u>76.5</u>
GNM-PT (ours)	76.3	77.6	75.0	**76.3**

Table 3. Acc. (%) comparison on Places-LT.

Method	Head	Med	Tail	Overall
DNN-based model (Backbone: ResNet152)				
LWS	40.6	39.1	28.6	37.6
RIDE	44.4	40.6	33.0	40.4
MisLAS	39.6	43.3	36.1	40.4
GCL	38.6	42.6	38.4	40.3
NCL	-	-	-	41.8
GPaCo	39.5	47.2	33.0	41.7
SHIKE	43.6	39.2	44.8	41.9
DNN-based model with SAM				
CCSAM	41.2	42.1	36.4	40.6
MHSA-based model (Backbone: ViT-B/16)				
Supplementary with linguistic data				
VL-LTR	54.2	48.5	42.0	<u>50.1</u>
RAC	48.7	48.3	41.8	47.2
Visual-only				
Decoder	-	-	-	46.8
LPT	47.6	52.1	48.4	49.7
LiVT	48.1	40.6	27.5	40.8
GNM-PT (ours)	46.6	53.3	49.4	<u>50.1</u>
GNM-PT (ours)	48.6	52.1	47.9	**50.0**

Effectiveness comparison

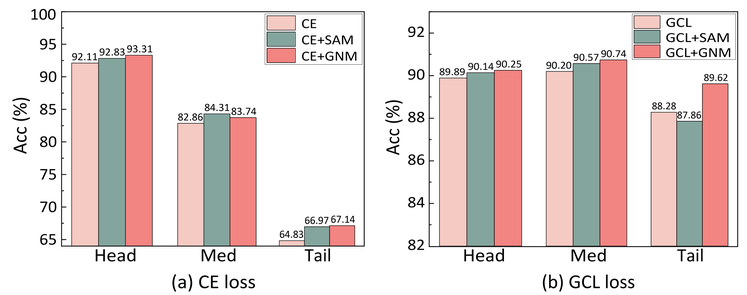

Figure 4. Effectiveness comparison of different classes

Concluding Remarks

By experiments and visualization, our GNM-PT has two-fold advantages:

- Balance the generalization capabilities of both head and tail classes.
- Almost no additional computational cost.

Limitation: Need to further re-balancing the classifier.