SP-Net: Slowly Progressing Dynamic Inference Networks

Huanyu Wang[1], Wenhu Zhang[2], Shihao Su[1], Hui Wang[1], Zhenwei Miao[3], Xin Zhan[3], and Xi Li[1,4,5]

[1] College of Computer Science and Technology, Zhejiang University
[2] Polytechnic Institute, Zhejiang University
[3] Alibaba Group, HangZhou, Zhejiang, China
[4] Shanghai Institute for Advanced Study, Zhejiang University
[5] Shanghai AI Laboratory
{huanyuhello, wenhuzhang, shihaocs, wanghui_17}@zju.edu.cn,
{zhenwei.mzw, zhanxin.zx}@alibaba-inc.com, xilizju@zju.edu.cn

A Clarify the performance gain of multi-stage classification more than feature evolution loss.

The performance gain of feature evolution loss seems limited because EXP-3 is conducted based on EXP-2. To better illustrate the performance gain of the two modules, we apply feature evolution regularization without multi-stage classification obtaining a performance of 94.96% with 0.46 GMACC on CIFAR-10 and 74.94% with 0.46 GMACC on CIFAR-100. The result shows that the feature evolution regularization does contributes much to the gain.

B Sensitivity analyses on hyper-parameters α, β, and γ.

We study parameters on CIFAR-10 based on ResNet-110 and show the grid search results in Table 1.

Table 1: Parameter Ablation Study on α, β, and γ

study on α			study on β			study on γ		
α	Acc. (%)	GFLOPs	β	Acc. (%)	GFLOPs	γ	Acc. (%)	GFLOPs
0.6	94.32	0.46	0.1	93.49	0.44	1e-5	95.06	0.46
0.8	94.67	0.46	0.3	94.76	0.46	1e-4	95.22	0.46
1.0	95.22	0.46	0.5	95.22	0.46	1e-3	94.33	0.42
1.2	94.75	0.47	0.7	95.03	0.45	1e-2	93.59	0.38

C Comparison of the result variants with other methods.

As shown in Table 2, we run the following four methods five times under the same settings: 0.4(±0.03)GFLOPs, based on ResNet-110.

Table 2: Result variants Comparison.

Methods	ConvAIG [32]	Skip-Net [41]	CoDiNet [35]	SP-Net
CIFAR-10	94.22±0.11	93.86±0.07	94.35±0.13	95.09±0.04
CIFAR-100	72.86±0.17	72.41±0.12	74.13±0.26	75.02±0.06

D Visualization on samples or metrics of how feature map changes with/without feature evolution loss.

First, we visualize the feature map changes in Fig. 1. Second, we measure the average KL-div in Fig. 2. of features from adjoining stages in epochs. A smaller value means a stable evolution.

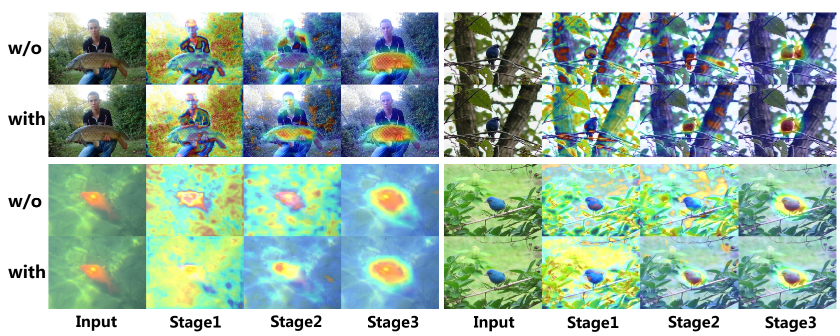

Fig. 1: Visualization on feature changes across the network

E Specifying the routers used in Dynamic ResBlocks.

We also show the structure of the router in Fig. 3.

F Experiments on other dynamic network backbones.

In principle, our method can facilitate dynamic backbones that assign a router to each computational block. We conduct the proposed method on three relevant networks in Table 3.

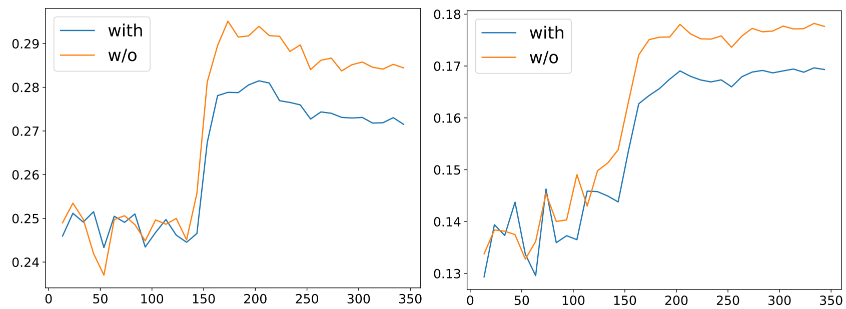

Fig. 2: Metric on feature changes across the network.

Table 3: Experiments on other backbones.

	CIFAR-10			CIFAR-100		
Methods	ConvAIG	CoDiNet	RDI-Net	ConvAIG	CoDiNet	RDI-Net
w/o.	94.24	94.47	95.10	72.90	72.90	74.31
with	94.97(0.73)	94.96(0.49)	95.16(0.06)	74.92(2.02)	74.18(1.28)	74.77(0.46)

G How do prior methods fix the instability issues?

First, pre-training or warm-up the network. SkipNet [41] proposes a supervised pre-train phase. BlockDrop [38] and RDI-Net [34] apply uniform sampling warm-up strategies. Second, data augmentation for similar samples. CoDiNet [35] utilizes self-supervised augmentations for each sample. RDI-Net [34] also measure the distance between samples and regularizes the corresponding inference paths.

H The instability may come from a bad router of pathways instead of the 'unstable feature evolution'.

The instability in dynamic inference network are two-fold. First, the unstable training phase might come from the unpowerful routers. To solve this problem, we use the auxiliary module for guided paths and labels. Second, the unstable feature evolution across network is also prone to instability. We invite the evolution loss to fix this issue.

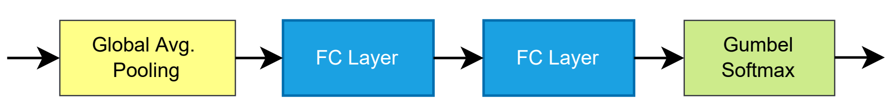

Fig. 3: The structure of the router in use