Boosting Residual Networks with Group Knowledge

Shengji Tang[1*], Peng Ye[1*], Baopu Li[4], Weihao Lin[1], Tao Chen[1†], Tong He[3], Chong Yu[2], Wanli Ouyang[3]

[1] School of Information Science and Technology, Fudan University, Shanghai, China
[2] Academy for Engineering and Technology, Fudan University, Shanghai, China
[3]Shanghai AI Laboratory, Shanghai, China
[4]Independent Researcher
eetchen@fudan.edu.cn

Abstract

Recent research understands residual networks from a new perspective of the implicit ensemble model. From this view, previous methods such as stochastic depth and stimulative training have further improved the performance of residual networks by sampling and training of its subnets. However, they both use the same supervision for all subnets of different capacities and neglect the valuable knowledge generated by subnets during training. In this paper, we mitigate the significant knowledge distillation gap caused by using the same kind of supervision and advocate leveraging the subnets to provide diverse knowledge. Based on this motivation, we propose a group knowledge based training framework for boosting the performance of residual networks. Specifically, we implicitly divide all subnets into hierarchical groups by subnet-in-subnet sampling, aggregate the knowledge of different subnets in each group during training, and exploit upper-level group knowledge to supervise lower-level subnet group. Meanwhile, we also develop a subnet sampling strategy that naturally samples larger subnets, which are found to be more helpful than smaller subnets in boosting performance for hierarchical groups. Compared with typical subnet training and other methods, our method achieves the best efficiency and performance trade-offs on multiple datasets and network structures. The code is at https://github.com/tsj-001/AAAI24-GKT.

Introduction

Residual structures, first introduced in (He et al. 2016), have become nearly indispensable in mainstream network architectures. It achieved great success in numerous architectures, such as convolutional networks (Tan and Le 2021; Ye et al. 2022b,a, 2023a; Mei et al. 2023), recurrent networks (Galshetwar, Patil, and Chaudhary 2022), MLP networks (Tolstikhin et al. 2021), and transformers (Vaswani et al. 2017; Huang et al. 2023; Liang et al. 2023). Considering the extraordinary performance of the residual structure, it is drawing increasing attention (He, Liu, and Tao 2020; Ding et al. 2022; Barzilai et al. 2022) to study the underlying mechanisms leading to their success. An interesting explanation is that residual networks can be regarded as an implicit ensemble of relatively shallow subnets, namely **unraveled view** (Veit, Wilber, and Belongie 2016; Barzilai et al. 2022). It opens up a new perspective to further improve the performance of residual networks. One of the common ways is to randomly sample subnets and train them individually. Stochastic depth (Huang et al. 2016) randomly drops a subset of layers and trains the remaining layers with ground truth labels. (Ye et al. 2022c) have observed a phenomenon known as "network loafing", where the standard training procedure fails to provide sufficient supervision and often causes subpar performance. To address the above problem, (Ye et al. 2022c) propose a stimulative training(ST) strategy by training randomly sampled subnets with outputs from the main network. However, these methods train various subnets with the same kind of supervision (i.e., stochastic depth always uses the ground-truth, and ST always uses

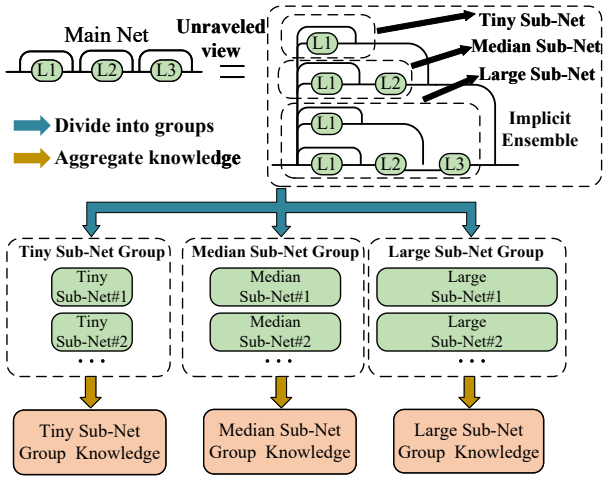


Figure 1: Illustration of unraveled view and group knowledge. Unraveled view shows that residual networks can be seen as an ensemble of numerous networks of different lengths. Inspired by this viewpoint, we allocate the subnets into subnet groups of different sizes, i.e. tiny, medium, and large subnet groups in the figure. Then we exploit the knowledge of subnet groups during training to boost the performance of given residual networks effectively and efficiently.

[*]These authors contributed equally.
[†]Corresponding author

the output of the main network), regardless of their unique capacities. It is straightforward to investigate **whether applying the same kind of supervision for diverse subnets is suitable**. Inspired by (Mirzadeh et al. 2020), we believe that "suitable supervision" should meet two important criteria: (1) **easy to be transferred (with a limited capacity gap),** (2) **containing rich and useful knowledge**.

To reduce the capacity gap, a common method (Mirzadeh et al. 2020) is to introduce extra intermediate models as teacher assistants. To get abundant knowledge, self knowledge distillation can be applied to learn from prior experience/knowledge, and ensemble knowledge distillation may further combine the supervisions of various teachers. Under the novel unraveled view, there are numerous subnets with various capacities, and the grouped assistant teachers naturally exist. Inspired by the observations above, we divide all subnets of a residual network into multiple groups by their capacity and aggregate their abundant knowledge during training, as shown in Figure 1. Interestingly, in the sociology field, transferring suitable knowledge is also important for improving the productivity of organizations (Baum and Ingram 1998). Group knowledge (Kane, Argote, and Levine 2005), collected from the same producing group, is considered easier to transfer to members in the neighboring group. Similar to the group knowledge in the field of sociology, we aggregate the knowledge from different subnets in the same group to build suitable supervision, and vividly call the aggregated knowledge as **network group knowledge**. Generally speaking, the knowledge produced by multiple subnet groups has two excellent properties: (1) it is naturally hierarchical and easy to be utilized to fill the capacity gap; (2) it is aggregated by numerous subnets containing abundant knowledge.

Based on the findings above, we further propose the **group knowledge based training (GKT)** framework, for boosting the performance of residual networks effectively and efficiently. In detail, during the training procedure, we first divide all subnets of a residual network into hierarchical subnet groups by a sampling strategy called subnet-in-subnet (SIS) sampling, then aggregate the knowledge of subnets in the same group by network logits moving average, and then supervise the subnet with an appropriate level of group knowledge. Moreover, we find that sampling and training larger subnets can better boost the performance of residual networks, thus we design an inheriting exponential decay rule to focus on the large or medium subnets. The proposed GKT framework can remarkably boost the network performance without any extra parameter (e.g., assistant teacher) or heavy computation cost (e.g., forwarding main net to obtain supervision). The efficacy and efficiency of GKT is shown in Figure 2. GKT does not require model topological modifications and only samples a part of a network (subnet) in the training procedure, resulting in less inference cost and training time compared with standard training (i.e., shown in the baseline of Figure 2). Because most CNN and transformer models adopt residual architecture and suffer from network loafing (Ye et al. 2023b), we further verify GKT on various CNN and transformer models.

In summary, our contributions are as follows:

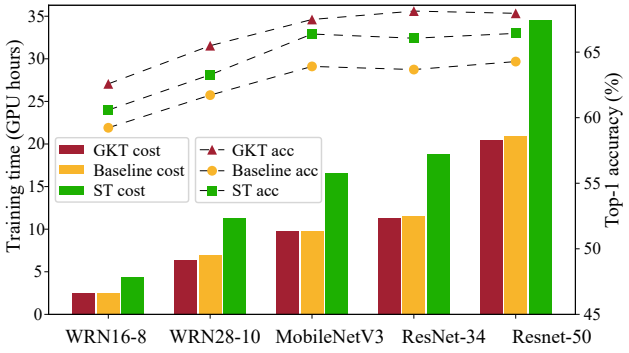


Figure 2: Training time and accuracy on TinyImageNet for group knowledge based training strategy, and other methods like standard training and stimulative training (ST).

- From the novel unraveled view of the residual network, we identify the hierarchical subnet group knowledge for the first time, which can provide better supervision for the diverse subnets of the residual network.
- We propose the GKT framework for boosting residual networks effectively and efficiently. In this framework, subnet-in-subnet sampling is adopted to implicitly divide all subnets into hierarchical subnet groups. Subnet logits' exponential moving average is exploited to aggregate the knowledge in the same subnet group.
- We experimentally verify sampling and training larger subnets can benefit residual networks more than smaller subnets. Thus we design an inheriting exponential decay rule to sample larger subnets for training subnets and preparing subnet groups.
- Comprehensive empirical comparisons and analysis show that GKT can reduce the capacity gap and efficiently improve the performance of various residual networks, including CNNs and transformers.

Related Works

Unraveled View

To better understand residual networks, (Veit, Wilber, and Belongie 2016) introduces a novel perspective named unraveled view, that interprets residual networks as an ensemble of shallower networks (i.e., subnets). Based on unraveled view, (Sun, Ding, and Guo 2022) verifies that shallow and deep subnets correspond to the low-degree and high-degree polynomials respectively, and shallow subnets play important roles when training residual networks. Then, (Barzilai et al. 2022) theoretically proves that the eigenvalues of the residual convolutional neural tangent kernel (CNTK) are made of weighted sums of eigenvalues of CNTK of subnets. Based on the unraveled view, (Huang et al. 2016) directly trains random subnets to improve the performance of residual networks. (Ye et al. 2022c) reveals the network loafing problem that standard training causes serious subnet performance degradation, and proposes stimulative training (ST) to supervise all subnets by the main net.

Following the research stream of unraveled view, we also focus our investigation on boosting residual networks by improving their subnets. Different from providing the same kind of supervision (e.g., ground truth (Huang et al. 2016) or main net logits (Ye et al. 2022c)) for all subnets, we pay attention to giving different subnets different supervisions according to their model capacities. Besides, we discover that the training of relatively large subnets can benefit the main net more, and we focus more on supervising the large or medium subnets instead of all subnets.

Knowledge Distillation

Conventional Knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015) transfers knowledge from a teacher network to a student network via logits (Kim, Park, and Kwak 2018; Shen and Xing 2022) or features (Bai et al. 2020; Jung et al. 2021), aiming at obtaining a compact and accurate student network. It usually requires additional cost because of training a larger teacher network. As a comparison, we do not require larger teachers or additional structures. And our target is to improve any given residual network effectively and efficiently by training its subnets well. Most related concepts among KD are described in detail as follows.

Ensemble Distillation. As the ensemble method (Dietterich 2000) is a useful technique to improve the performance of deep learning models, it is generally considered that, an ensemble of multiple teacher models can commonly provide supervision with higher quality compared with a single teacher. (Du et al. 2020) studies the conflicts and competitions among teachers and introduces a dynamic weighting method for better fusing teachers' knowledge. However, typical ensemble distillation methods need additional teacher models to provide supervision for a single student model. It requires huge computation cost and is not suitable for multiple coupled subnet students. Differently, we do not need extra teacher models or inference. For multiple unique coupled subnet students, we specifically provide suitable supervision by aggregating the hierarchical subnet group knowledge.

Self Distillation. To save the cost introduced by a larger teacher network, the self distillation (SD) attempts to provide supervision within the student network itself in training. (Yun et al. 2020) narrows down the predictive distribution deviation between different samples of the same label to provide the regularization supervision. (Deng and Zhang 2021; Kim et al. 2021) utilize the snapshot of the previous output logits as supervision to learn from prior experience. (Shen et al. 2022) rearranges the data sampling by including mini-batch from previous iteration, and uses the on-the-fly soft targets generated in the previous iteration to supervise the network. Similarly, we also consider that the historic information during the training contains abundant knowledge. However, previous SD methods only utilize the intermediate features or output logits of a single main net. Differently, motivated by the unraveled view, we focus on various subnets with distinct capacities and utilize their aggregated knowledge in different iterations.

Online Distillation. Online knowledge distillation (OKD) introduces extra multiple branches or models manually during the training procedure to extract knowledge. ONE (Zhu, Gong et al. 2018) introduces additional branches to create a native ensemble teacher and transfer the knowledge from the ensemble teacher to each branch. PCL (Wu and Gong 2021) builds ensemble teachers by integrating different branches and meaning them temporally to supervise each branch. OKDDip (Chen et al. 2020) proposes to enhance the diversity of multiple branches with attention-based weights. Different from OKD(Wu and Gong 2021; Chung et al. 2020) using the same knowledge-transfer strategies for each fixed student, we focus on providing tailored knowledge for dynamically sampled subnets with a lower capacity gap. Moreover, GKT aggregates intrinsic knowledge **without any extra architecture** causing easier implementation for different architectures and less training cost.

Capacity Gap. There is a counter-intuitive phenomenon (Cho and Hariharan 2019) called capacity gap, referring to the fact that a larger and more accurate teacher model does not necessarily teach the student model better. This phenomenon is attributed to the capacity mismatch, that a tiny student model has insufficient ability to mimic the behavior of a large teacher with huge capacity. For transferring knowledge better, numerous works are proposed to bridge the capacity gap. (Mirzadeh et al. 2020) introduces extra intermediate models as teacher assistants. (Li et al. 2022; Guo 2022) propose asymmetric temperature scaling for teacher and student to make larger teachers teach better. Since there are subnets with different capacities under the unraveled view, the capacity gap problem becomes more serious when transferring knowledge to various subnets. Different from the above methods, we bridge the capacity gap by aggregating the hierarchical subnet group knowledge, without additional models (Mirzadeh et al. 2020) or changes of hyperparameters like temperature (Li et al. 2022; Guo 2022).

Group Knowledge Based Training (GKT)

Framework

The overview of group knowledge based training (GKT) is shown in Figure 3. We divide the total number of training steps by the given number of subnet groups to get multiple equal training loops. For each loop, the operations of GKT consist of three parts: (1) Subnet Group Division; (2) Group Knowledge Aggregation; (3) Group Knowledge Transfer.

Subnet Group Division: At the beginning of each loop, we sample a subnet from the main net. Then, we continue to sample a subnet from the parent subnet, until the end of the loop. All subnets sampled in the same generation on different loops are divided into the same group. After sampling, we forward the subnet to obtain logits and compute the loss.

Group Knowledge Aggregation: The subnet logits, generated on the specific step of each loop, are used to update the subset's corresponding group knowledge by logit-level exponential moving average.

Group Knowledge Transfer: At each training step, the ground truth and neighboring larger group knowledge will be utilized to supervise the current subnet. The pseudo-code of GKT is shown in **Appendix B1**.

During testing, GKT forwards the residual network without modifying the structure or changing the testing pipeline.

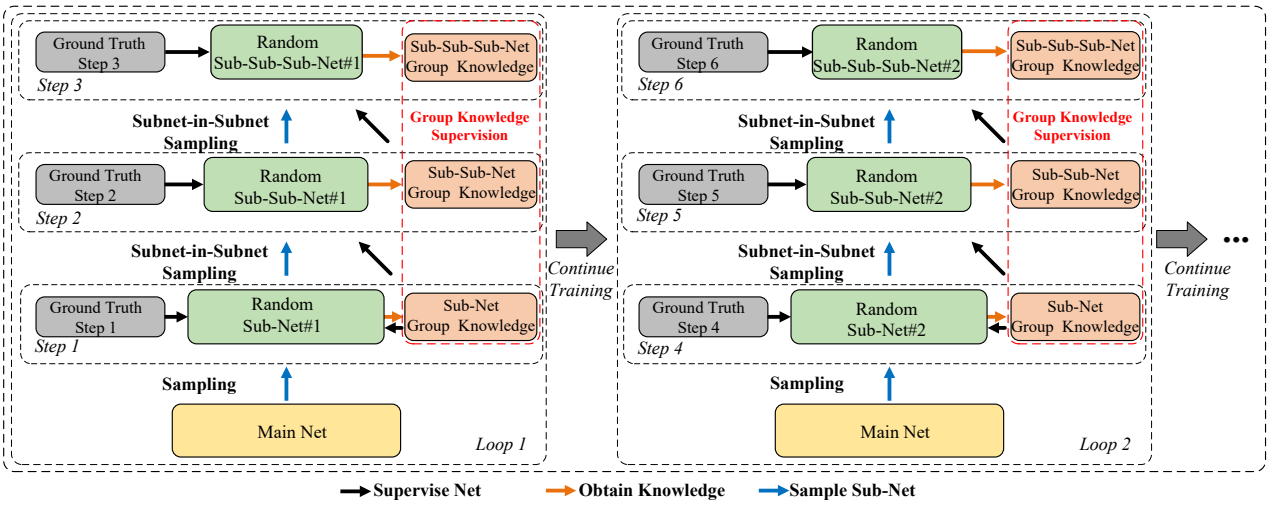


Figure 3: Overview of the group knowledge based training (GKT) framework. To give an example, we suppose there are three groups. GKT divides all subnets into hierarchical subnet groups by subnet-in-subnet sampling, aggregates the knowledge of different subnets of different steps in the same subnet group, and uses the aggregated group knowledge to supervise the neighboring subnet group. To avoid continuously sampling the tiniest subnet, multiple sampling loops are applied to successively sample subnets from the main network again, as expressed in the left part (i.e., loop 1) and the right part (i.e., the next loop 2).

We will introduce these three parts in detail as follows.

Group Division: Subnet-in-Subnet Sampling. To alleviate the capacity gap when supervising diverse subnets, we propose to divide subnets into hierarchical groups. An intuitive division method is to randomly sample subnets and directly divide them by their parameters or FLOPs. However, these direct methods introduce many hyper-parameters, such as partition bounds of each group, and may limit the number of possible subnets in a group. Therefore, we introduce subnet-in-subnet (SIS) sampling to naturally divide all subnets into hierarchical subnet groups. Specifically, SIS sampling strategy samples subnet from the parent subnet and divides the subnet sampled in the same generation into the same group. Besides, to avoid being restricted to tiny subnets due to an unending sampling, the total training steps are equally partitioned into several loops, and at the beginning of each loop, the sampling process starts from the main net.

Formally, we denote the subnet $\mathcal{N}_s$ belonging to the t-th group and sampled in the r-th loop as $\mathcal{N}_{s,t}^r$. Given a parent subnet $\mathcal{N}_{s,t}^r$, the next sampled subnet is generated as

$$\mathcal{N}_{s,t+1}^r = \pi(\mathcal{N}_{s,t}^r), \tag{1}$$

where $\pi(.)$ is the sampling operation representing randomly sampling a subnet from a network based on given sampling distribution. At the beginning of a loop, we sample a subnet from the main net, expressed formally as $\mathcal{N}_{s,1}^r = \pi(\mathcal{N}_m)$. Then, we continue to sample one subnet from the parent subnet at each step, as shown in Equation 1. The subnets sampled in the t-th generation step in all loops are regarded in the t-th group. After M sampling steps, the current training loop ends, and the sampling comes to the next training loop. It is noticed that a specific subnet might be sampled in any loop and divided into any groups. Subnet-in-subnet sampling utilizes the inheritance relationship of subnets as a criterion for loose group division, and its surpassing effectiveness is verified in **Appendix D2**.

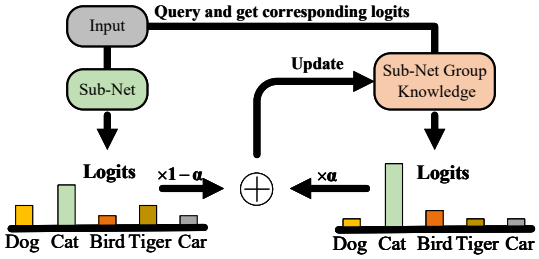


Figure 4: Updating mechanism of subnet group knowledge. We use the exponential moving average (EMA) of network logits to update the subnet group knowledge.

Group Knowledge Aggregation: Subnet Logits Exponential Moving Average. As network logits are the most commonly used supervision containing substantial high-level knowledge, we save and aggregate output logits of different subnets in the same group as group knowledge. However, considering that the subnets in the same group distribute in different temporal frames, it is unbearable to save all of their historic logits. Inspired by model parameter exponential moving average (EMA) in self-supervised learning (He et al. 2020), we adopt subnet logits EMA to aggregate group knowledge effectively and resource-friendly.

Formally, we denote $\mathcal{K}_t \in \mathbb{R}^{N\times k}$, where N and k are the data and class number of the dataset, as the t-th group knowledge. As shown in Figure 4, for mini-batch samples

$x \in \mathbb{R}^{b\times c\times h\times w}$, we denote $\mathcal{K}_t^I \in \mathbb{R}^{b\times k}$ as the corresponding group knowledge queried from $\mathcal{K}_t$ by indices I. Then the corresponding subnet group knowledge is updated as

$$\mathcal{K}_t^I := \alpha p(\theta_{\mathcal{N}_{s,t}}, x) + (1-\alpha)\mathcal{K}_t^I \tag{2}$$

where $:=$ represents updating and α is the EMA coefficient to balance the intensity of updating. If it is the first time to update $\mathcal{K}_t^I$, we directly initialize it with $p(\theta_{\mathcal{N}_{s,t}}, x)$. It is worth noting that because we can store $\mathcal{K}$ into the disk instead of GPU memory, subnet logits EMA only introduces negligible storage and computation cost.

Hierarchical Group Knowledge Transfer. After group knowledge aggregation, there is a group knowledge pool containing different levels of group knowledge. To reduce the capacity gap and obtain abundant knowledge, we transfer the neighboring larger group knowledge to the subnet. Formally, for a given subnet $\mathcal{N}_{s,t}$ in t-th group, the supervision is $\mathcal{K}_{t-1}^I$, the loss is Kullback-Leible divergence

$$\mathcal{L}_{GK} = KL(\mathcal{K}_{t-1}^I, p(\theta_{\mathcal{N}_{s,t}}, x)). \tag{3}$$

And the total loss of GKT is the weighted sum of group knowledge supervision and standard cross-entropy loss

$$\mathcal{L}_{GKT} = CE(p(\theta_{\mathcal{N}_{s,t}}, x), y) + \beta KL(\mathcal{K}_{t-1}^I, p(\theta_{\mathcal{N}_{s,t}}, x)). \tag{4}$$

β is a loss balanced coefficient. For subnets in the largest group, they are supervised by their own group knowledge.

Inheriting Exponential Decay Rule. Under the unraveled view, there are 2^L subnets in the given residual network, where L is the number of residual blocks, which is a huge sampling space. It's almost impossible to train every subnet sufficiently. Thus, (Ye et al. 2022c) proposes to keep the ordered residual structure of subnets for sampling space reduction and easier subnet training. (Huang et al. 2016) follows the intuition that the earlier blocks extract more important low-level features and are more reliably present, thus adopting linear decay sampling to drop the deeper blocks with higher probability. Furthermore, we experimentally verify that training relatively large subnet benefits the main net more, which will be discussed detailedly in Section .

Inspired by these, we propose an inheriting exponential decay subnet sampling strategy. To ensure larger subnets are sampled with a higher probability, we propose to change the sampling distribution of global block-wise linear decay (Huang et al. 2016) to stage-wise exponential decay. Since we utilize the subnet-in-subnet sampling, our sampling distribution is dynamic w.r.t. the parent subnet, thus called "inheriting". Besides, we also keep the ordered residual structure when sampling. The effectiveness of inheriting exponential decay sampling is verified in **Appendix D4**.

Formally, for a parent subnet $\mathcal{N}_{s,t}$ belonging to t-th group, it is usually made up of several stages, and each stage contains several residual blocks. We suppose $\mathcal{N}_{s,t}$ has D stages and the block number of each stage is $[l_1, l_2, ...l_D]$. For each stage, we utilize a sampling distribution corresponding to the number of retained ordered residual blocks to control the sampling process. To reduce hyper-parameters, we give a total base number $q \in (0, 1)$, and the sampling distribution of d-th stage, i.e. χ_d, corresponding to block number $[1, 2, ...l_d]$, is computed as

$$u = q^{D-d+1} \tag{5}$$

$$v = [u^{l_d}, u^{l_d-1}, ...u] \tag{6}$$

$$\chi_d = [\frac{u^{l_d}}{\sum v}, \frac{u^{l_d-1}}{\sum v}, ...\frac{u}{\sum v}] \tag{7}$$

where u, v are temporary values. Superscript represents the exponent. From Equation 6 and 7, we can observe that the probability of sampling larger subnets from any parent network has been remarkably increased.

Experiments

We first verify the effectiveness and efficiency of GKT on image classification with CNNs and transformers. To further demonstrate the generality of GKT, we conduct experiments on downstream tasks. Then, ablation experiments show the indispensability of each component. Finally, investigation experiments reveal the mechanism of GKT. The details of experiment settings are shown in **Appendix A**.

Image Classification

We demonstrate the effectiveness and efficiency of GKT on typical residual convolutional networks including ResNet-34, ResNet-50 (He et al. 2016), WRN16-8, WRN28-10 (Zagoruyko and Komodakis 2016) and MobileNetV3 (Howard et al. 2019), and mainstream datasets including CIFAR-100, Tiny ImageNet and ImageNet-1K. Observing that residual connections popularly exist in visual transformers, we conduct experiments on transformers including ViT (Dosovitskiy et al. 2020), Swin (Liu et al. 2021) and CaiT (Touvron et al. 2021) to prove the generalization ability further. To verify the superiority of GKT, we compare the test accuracy with standard training, subnet training methods, i.e., ST (Ye et al. 2022c) and Stodepth (Huang et al. 2016), prevailing SD methods, i.e., CS-KD (Yun et al. 2020), PS-KD (Kim et al. 2021), DLB (Shen et al. 2022) and LWR (Deng and Zhang 2021), and online distillation method. i.e., ONE (Zhu, Gong et al. 2018).

The results on **CIFAR-10/100** and **Tiny ImageNet** are shown in Table 1 (on various CNNs) and Table 3 (on various transformers). For **CNNs**, GKT universally and remarkably boosts the performance of different residual networks on different datasets. To be more specific, compared with standard training, the average Top-1 test accuracy improvements of GKT on different networks are up to 1.67% on CIFAR-100 and 1.22% on Tiny ImageNet, respectively. Besides, compared with other SD and subnet training methods, GKT consistently achieves a new state-of-the-art performance, which demonstrates the superiority of GKT over other approaches. Specifically, the Top-1 test accuracy gains of GKT compared with the second-best method are up to 1.54% on CIFAR-100 and 2.02% on Tiny ImageNet respectively. For **transformers**, it is observed that GKT remarkably improves the accuracy e.g., +4.57% on Tiny ImageNet. This verifies the potential of GKT to boost different residual architectures.

We also verify the generalization ability and efficiency of GKT on **ImageNet-1K**, i.e. a mainstream large scale

Dataset	Method	WRN16-8	WRN28-10	MobileNetV3	ResNet-34	ResNet-50
CIFAR-100	Baseline	79.95	82.17	78.09	77.78	78.14
	StoDepth (Huang et al. 2016)	80.64	82.75	78.77	80.62	78.43
	ONE (Zhu, Gong et al. 2018)	80.49	83.02	80.85	79.96	80.44
	CS-KD (Yun et al. 2020)	80.77	81.25	78.60	79.35	80.34
	PS-KD (Kim et al. 2021)	81.17	82.53	79.36	79.30	80.07
	LWR (Deng and Zhang 2021)	81.05	82.00	80.23	79.81	80.70
	DLB (Shen et al. 2022)	80.87	81.35	78.55	79.26	80.36
	ST (Ye et al. 2022c)	80.75	82.84	81.07	78.62	81.06
	GKT	**81.53**	**84.38**	**81.70**	**81.40**	**81.73**
TinyImageNet	Baseline	59.23	61.72	63.91	63.67	64.28
	StoDepth (Huang et al. 2016)	60.29	62.02	64.97	65.75	65.80
	ONE (Zhu, Gong et al. 2018)	62.13	64.15	65.39	66.98	66.58
	CS-KD (Yun et al. 2020)	60.23	62.24	64.72	66.11	66.74
	PS-KD (Kim et al. 2021)	60.93	63.23	66.41	65.77	65.96
	LWR (Deng and Zhang 2021)	61.62	63.91	66.43	63.51	65.03
	DLB (Shen et al. 2022)	61.48	64.29	65.05	65.86	65.78
	ST (Ye et al. 2022c)	60.58	63.27	66.38	66.06	66.43
	GKT	**62.58**	**65.49**	**67.49**	**68.13**	**67.96**

Table 1: Main experimental results of the proposed GKT and other methods on the CIFAR-100 and TinyImageNet datasets. The best performance is highlighted in bold, and the second-best performance is highlighted in underline.

Method	ResNet-34		ResNet-50		Swin-T		Swin-S		ViT-S	
	Top-1 Acc(%)	Cost (hours)	Top-1 Acc(%)	Cost (hours)	Top-1 Acc(%)	Cost (hours)	Top-1 Acc(%)	Cost (hours)	Top-1 Acc(%)	Cost (hours)
Baseline	74.70	98.04	76.98	202.16	77.50	301.47	79.36	460.82	75.55	301.91
StoDepth	74.96	**94.76**	77.43	196.77	79.62	297.74	81.23	452.48	77.03	296.41
ST	75.25	166.98	77.60	345.45	79.87	437.23	81.43	685.28	77.27	447.77
GKT	75.50	95.48	78.10	**194.32**	80.40	**294.38**	82.00	**451.18**	78.51	**295.62**
GTK (+epoch)	**75.83**	141.13	**78.40**	290.36	**80.72**	442.55	**82.26**	678.24	**78.83**	445.78

Table 2: Verification of typical CNNs (ResNet) and Transformers (Swin Transformer and ViT) on the ImageNet-1K dataset.

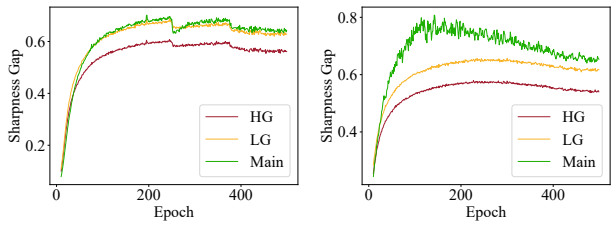
dataset. As shown in Table 2, both in CNNs and transformers, GKT can achieve significant performance gains over the baseline of different networks, and perform better than Stodepth and ST. Meanwhile, the time cost of GKT is almost the smallest among these methods. Besides, the performance of different networks can be further boosted when increasing the training epoch of GKT.

Ablation Experiments

To measure the effect of each component, we remove the components of GKT one by one on WRN28-10. The results are shown in Table 5. The first line is the performance of GKT, which is the best in the table. And it is observed that with the removal of each component, the performance is inferior step-by-step, which can prove the separate effects of each component. More experiments comparing our components with other naïve methods are shown in **Appendix D**.

Downstream Tasks

To verify the generalization of GKT, we finetune the ImageNet pretrained ResNet-50 of GKT and baseline on downstream tasks with three well-known frameworks including Faster R-CNN (Ren et al. 2015), Mask R-CNN (He et al. 2017), Panoptic FPN (Kirillov et al. 2019) on COCO2017 (Lin et al. 2014). The results are shown in Table 4, and GKT consistently obtains improvement on object detection (e.g., +0.7% Det mAP on Faster R-CNN), instance segmentation (+0.8% Seg mAP on Mask R-CNN), and panoptic segmentation (+0.62% SQ on Panoptic FPN). It demonstrates that GKT can facilitate networks to learn more general representations and benefit different tasks.



(a) ResNet-50 (b) MobileNetV3

Figure 5: The capacity gap (measured by sharpness gap (Guo 2022; Rao et al. 2022)) of different supervision strategies (i.e., obtaining knowledge from Largest Group (LG), Main Net (Main), and Hierarchical Group (HG)) during the training process of (a) ResNet-50, (b) MobileNetV3.

	ViT_C10	Swin_C10	CaiT_C10	ViT_C100	Swin_C100	CaiT_C100	ViT_TinyImg
Baseline	93.23	94.05	94.71	72.15	76.02	76.52	54.14
StoDepth	93.58	94.46	94.91	73.81	76.87	76.89	57.07
GKT	**93.8**	**95.21**	**95.24**	**75.31**	**77.32**	**78.79**	**58.71**

Table 3: Verification of various transformers on CIFAR-10/100 (C10/100) and Tiny ImageNet (TinyImg).

	ResNet-50	Faster R-CNN(Det)	Mask R-CNN(Det&Seg)		Panoptic FPN(Panotic Seg)		
	ImageNet Acc	Det mAP@0.5	Det mAP@0.5	Seg mAP@0.5	PQ	SQ	RQ
Baseline	76.98	59.4	59.6	56.5	41.76	78.21	51.38
GKT	**78.10**	**60.1**	**60.1**	**57.3**	**42.27**	**78.83**	**51.62**

Table 4: Verification on object detection, instance segmentation and panoptic segmentation with schedule 1× on COCO2017.

SIS Sampling	SL-EMA	HGKT	Top-1 Acc(%)
✓	✓	✓	**84.38**
✓	✓	×	83.51
✓	×	×	82.40
×	×	×	82.17

Table 5: Influence of different components including Subnet-in-Subnet (SIS) Sampling, Subnet Logits EMA (SL-EMA), Hierarchical Group Knowledge Transfer (HGKT).

Sampling Strategy	ST	GKT
UR(1∼12)	82.84	81.06
UR(5∼6)	81.51	80.58
UR(7∼8)	**82.91**	82.32
UR(9∼10)	82.76	82.93
UR(11∼12)	82.58	83.51
EDR(1∼12)	82.9	**84.38**

Table 6: Influence of different sampling strategies on the final performance. To explore the influence of sampling space, we implement ST and GKT with different sampling rules and spaces, including uniform rule (UR) on several different spaces and Exponential Decay Rule (EDR) on the full space.

Investigation Experiments

Investigation 1: Do we need to train every subnet well? ST reveals the loafing issue and proposes to train each subnet equally. However, according to (Barzilai et al. 2022), the weights of subnets contributing to the main net performance are not the same. Inspired by this, we explore the influence of different subnets on the main net. Specifically, we first divide the sampling space into several mini-spaces by the layer number of the subnet. Then we conduct ST or GKT following the uniform rule (UR) (Ye et al. 2022c) on these mini-spaces and the full space, compared with following the exponential decay rule (EDR) on the full space. As shown in Table 6, GKT performs better on larger mini-spaces, and both GKT and ST perform well on relatively large mini-spaces. Besides, following EDR, GKT and ST both achieve the best performance. These prove that we should pay more attention to training relatively large subnets.

Investigation 2: Does our method reduce the capacity gap? An important purpose of GKT is to reduce the capacity gap during subnet training. To verify these, we record the parameters of subnets in different groups in GKT on WRN28-10. We equally and linearly divide the parameters range into three parts including tiny part (7.4∼17.17M), medium part (17.17∼26.85M) and large part (26.85∼36.54M), and use the recorded parameters of each group to compute the sampling ratio and parameter expectation. The results are shown in Table 7. We can observe that GKT is more inclined to sample larger subnets for larger groups, and the parameter expectation of different groups is hierarchical. Further, we quantify the capacity gap by measuring the sharpness gap, which is a commonly used metric to represent the capacity gap (Guo 2022; Rao et al. 2022). As shown in Figure 5, compared with directly transferring knowledge from the main net or largest group, GKT can significantly and consistently obtain the lowest sharpness gap.

Group Rank	Tiny	Medium	Large	Expectation
Group#1	0.35	0.34	0.31	21.54M
Group#2	0.19	0.29	0.52	25.50M
Group#3	0.06	0.15	0.79	30.43M

Table 7: Properties of different subnet groups obtained by subnet-in-subnet sampling on WRN28-10 (36.54M). We show the sampling ratio of tiny (7.4∼17.17M), medium (17.17∼26.85M), and large(26.85∼36.54M) subnets for different subnet groups and give the parameter expectation.

Conclusion

In this work, from the unraveled view of residual networks, we observe that all the subnets with different capacities are provided with the same supervision in previous methods, leading to a serious capacity gap and lack of knowledge. To solve these issues, we identify the hierarchical subnet group knowledge inspired by the sociology field, and propose a novel group knowledge based training (GKT) framework to boost residual networks effectively and efficiently. Besides, we find training relatively large subnets benefit the main net more, which guides our design of the subnet sampling strategy. Comprehensive experiments on multiple tasks and networks verify GKT's generalization and superiority.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. U1909207 and 62071127), National Key Research and Development Program of China (No. 2022ZD0160100), Shanghai Natural Science Foundation (No. 23ZR1402900), and Zhejiang Lab Project (No. 2021KH0AB05).

References

Bai, T.; Chen, J.; Zhao, J.; Wen, B.; Jiang, X.; and Kot, A. 2020. Feature distillation with guided adversarial contrastive learning. *arXiv preprint arXiv:2009.09922*.

Barzilai, D.; Geifman, A.; Galun, M.; and Basri, R. 2022. A Kernel Perspective of Skip Connections in Convolutional Networks. *arXiv preprint arXiv:2211.14810*.

Baum, J. A.; and Ingram, P. 1998. Survival-enhancing learning in the Manhattan hotel industry, 1898–1980. *Management science*, 44(7): 996–1016.

Chen, D.; Mei, J.-P.; Wang, C.; Feng, Y.; and Chen, C. 2020. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 3430–3437.

Cho, J. H.; and Hariharan, B. 2019. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4794–4802.

Chung, I.; Park, S.; Kim, J.; and Kwak, N. 2020. Feature-map-level online adversarial knowledge distillation. In *International Conference on Machine Learning*, 2006–2015. PMLR.

Deng, X.; and Zhang, Z. 2021. Learning with retrospection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7201–7209.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*, 1–15. Springer.

Ding, Z.; Chen, S.; Li, Q.; and Wright, S. J. 2022. Overparameterization of deep ResNet: zero loss and mean-field analysis. *Journal of machine learning research*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Du, S.; You, S.; Li, X.; Wu, J.; Wang, F.; Qian, C.; and Zhang, C. 2020. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *advances in neural information processing systems*, 33: 12345–12355.

Galshetwar, V. M.; Patil, P. W.; and Chaudhary, S. 2022. Lrnet: lightweight recurrent network for video dehazing. *Signal, Image and Video Processing*, 1–9.

Guo, J. 2022. Reducing the teacher-student gap via adaptive temperatures.

He, F.; Liu, T.; and Tao, D. 2020. Why resnet works? residuals generalize. *IEEE transactions on neural networks and learning systems*, 31(12): 5349–5362.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *stat*, 1050: 9.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1314–1324.

Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; and Weinberger, K. Q. 2016. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 646–661. Springer.

Huang, Y.; Ye, P.; Huang, X.; Li, S.; Chen, T.; and Ouyang, W. 2023. Experts weights averaging: A new general training scheme for vision transformers. *arXiv preprint arXiv:2308.06093*.

Jung, S.; Lee, D.; Park, T.; and Moon, T. 2021. Fair feature distillation for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12115–12124.

Kane, A. A.; Argote, L.; and Levine, J. M. 2005. Knowledge transfer between groups via personnel rotation: Effects of social identity and knowledge quality. *Organizational behavior and human decision processes*, 96(1): 56–71.

Kim, J.; Park, S.; and Kwak, N. 2018. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31.

Kim, K.; Ji, B.; Yoon, D.; and Hwang, S. 2021. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6567–6576.

Kirillov, A.; Girshick, R.; He, K.; and Dollár, P. 2019. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6399–6408.

Li, X.-C.; Fan, W.-s.; Song, S.; Li, Y.; Zhan, D.-C.; et al. 2022. Asymmetric Temperature Scaling Makes Larger Networks Teach Well Again. In *Advances in Neural Information Processing Systems*.

Liang, C.; Bai, W.; Qiao, L.; Ren, Y.; Sun, J.; Ye, P.; Yan, H.; Ma, X.; Zuo, W.; and Ouyang, W. 2023. Rethinking the BERT-like Pretraining for DNA Sequences. *arXiv preprint arXiv:2310.07644*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft

coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Mei, Z.; Ye, P.; Ye, H.; Li, B.; Guo, J.; Chen, T.; and Ouyang, W. 2023. Automatic Loss Function Search for Adversarial Unsupervised Domain Adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*.

Mirzadeh, S. I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; and Ghasemzadeh, H. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5191–5198.

Rao, J.; Meng, X.; Ding, L.; Qi, S.; and Tao, D. 2022. Parameter-efficient and student-friendly knowledge distillation. *arXiv preprint arXiv:2205.15308*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Shen, Y.; Xu, L.; Yang, Y.; Li, Y.; and Guo, Y. 2022. Self-distillation from the last mini-batch for consistency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11943–11952.

Shen, Z.; and Xing, E. 2022. A fast knowledge distillation framework for visual recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, 673–690. Springer.

Sun, T.; Ding, S.; and Guo, L. 2022. Low-degree term first in ResNet, its variants and the whole neural network family. *Neural Networks*, 148: 155–165.

Tan, M.; and Le, Q. 2021. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, 10096–10106. PMLR.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272.

Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 32–42.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veit, A.; Wilber, M. J.; and Belongie, S. 2016. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29.

Wu, G.; and Gong, S. 2021. Peer collaborative learning for online knowledge distillation. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 35, 10302–10310.

Ye, P.; He, T.; Li, B.; Chen, T.; Bai, L.; and Ouyang, W. 2023a. β-DARTS++: Bi-level Regularization for Proxy-robust Differentiable Architecture Search. *arXiv preprint arXiv:2301.06393*.

Ye, P.; He, T.; Tang, S.; Li, B.; Chen, T.; Bai, L.; and Ouyang, W. 2023b. Stimulative Training++: Go Beyond The Performance Limits of Residual Networks. *arXiv preprint arXiv:2305.02507*.

Ye, P.; Li, B.; Chen, T.; Fan, J.; Mei, Z.; Lin, C.; Zuo, C.; Chi, Q.; and Ouyang, W. 2022a. Efficient joint-dimensional search with solution space regularization for real-time semantic segmentation. *International Journal of Computer Vision*, 130(11): 2674–2694.

Ye, P.; Li, B.; Li, Y.; Chen, T.; Fan, J.; and Ouyang, W. 2022b. β-DARTS: Beta-Decay Regularization for Differentiable Architecture Search. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10864–10873. IEEE.

Ye, P.; Tang, S.; Li, B.; Chen, T.; and Ouyang, W. 2022c. Stimulative Training of Residual Networks: A Social Psychology Perspective of Loafing. In *Thirty-Sixth Conference on Neural Information Processing Systems*.

Yun, S.; Park, J.; Lee, K.; and Shin, J. 2020. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13876–13885.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *British Machine Vision Conference 2016*. British Machine Vision Association.

Zhu, X.; Gong, S.; et al. 2018. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31.