Knowledge Distillation via Instance Relationship Graph
Supplementary Document

Yufan Liu*[a], Jiajiong Cao*[b], Bing Li†[a], Chunfeng Yuan†[a], Weiming Hu[a], Yangxi Li[c] and Yunqiang Duan[c]

[a]*NLPR, Institute of Automation, Chinese Academy of Sciences*
[b]*Ant Financial*
[c]*National Computer Network Emergency Response Technical Team/Coordination Center of China*

In this supplementary document, we give more details about the experimental settings and results.

1. Further Discussion about $L_{IRG\text{-}t}$

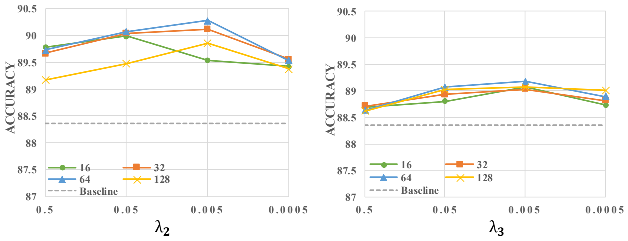
Though we have verified the effectiveness of $L_{IRG\text{-}t}$ by comparing the performance of L_{IRG} and L_{MTK}, the performance of single $L_{IRG\text{-}t}$ is not shown due to the limitation of space. Therefore, in this section, we train the student network only with $L_{IRG\text{-}t}$, and analyze the performance.

In the experiment, ResNet20 and ResNet20-x0.5 are adopted as the teacher network and the student network, respectively. CIFAR10 is used for training and validation. In addition, besides L_{IRG}, **FSP** [2] is selected as a competing method, since **FSP** also distills knowledge from the overall inference procedure.

1.1. Comparison with L_{IRG}

As shown in Figure 1, both L_{IRG} and $L_{IRG\text{-}t}$ help improve the performance significantly. In addition, L_{IRG} and $L_{IRG\text{-}t}$ are complementary to each other.

For L_{IRG}, though the performance gain over the baseline is more significant than that of $L_{IRG\text{-}t}$, it is much more sensitive to the value of λ_2. In particular, when λ_2 varies from 0.0005 to 0.5, the performance fluctuation is $0.6 - 0.8\%$. Under the same condition, the performance fluctuation of $L_{IRG\text{-}t}$ is $0.2 - 0.4\%$. It is because L_{IRG} concentrates on extracting sufficient knowledge (instance relationships and instance features). In this way, it leads to strong regularization to the student and cause large fluctuation. On the contrary, $L_{IRG\text{-}t}$ considers the feature space transformation, which is a more moderate constraint compared with L_{IRG}. It helps the student converge smoothly and stably. Therefore, L_{IRG} and $L_{IRG\text{-}t}$ are complementary to each other. By combining them, L_{MTK} extracts sufficient and moderate knowledge, and thus shows stable convergence as well as large performance improvement.



(a) L_{IRG}

(b) $L_{IRG\text{-}t}$

Figure 1: (a) Performance under different batch sizes and λ_2 settings of L_{IRG}. (b) Performance under different batch sizes and λ_3 settings of $L_{IRG\text{-}t}$. Note that “Baseline” is trained with L_{GT} only.

1.2. Comparison with FSP

Both **FSP** and the proposed $L_{IRG\text{-}t}$ take the feature space transformation as the distilled knowledge. **FSP** defines a Flow of Solution Procedure (FSP) matrix to transfer information. The inner product of the i-th channel of the l_1-th layer and the j-th channel of the l_2-th layer is computed as an element of the matrix. While $L_{IRG\text{-}t}$ first performs global average pooling for the feature maps of the l_1-th layer and the l_2-th layer. Then, Euclidean distance of the two pooled features is computed as the element of the feature transformation matrix.

However, there are two limitations of **FSP**. First, the computational cost of **FSP** is rather high. For N input instances, **FSP** needs to compute inner production for $N \times C_{l_1} \times C_{l_2}$ times, where C_{l_1} and C_{l_1} are the channel

*Both authors contributed equally to this research.

†Corresponding authors: Bing Li (bli@nlpr.ia.ac.cn), Chunfeng Yuan (cfyuan@nlpr.ia.ac.cn)

Table 1: Training time of one iteration when batch size is 64.

	KD [1]	FSP [2]	AT [3]	Rocket [4]	L_{IRG}	L_{MTK}
CIFAR10/100 (ms)	50	302	74	51	190	205
ImageNet (s)	3.85	4.23	3.97	3.86	3.92	4.04

numbers of the l_1-th layer and the l_2-th layer, respectively. While the proposed $L_{IRG\text{-}t}$ only requires N times of computation. Further, the inner production is computed on the whole feature map for **FSP**, which requires much computational resources. On the contrary, $L_{IRG\text{-}t}$ uses the average pooled feature vectors for computation. Therefore, the training time of **FSP** is $4-5$ times of that of $L_{IRG\text{-}t}$.

Second, **FSP** is a relative hard constraint. Since it utilizes very detailed information, for example, every pixel of the intermediate-layer features, it is hard for the student to converge to a good solution. Therefore, it takes longer time to select an appropriate weight for the **FSP** loss. In addition, using pixel level feature maps makes **FSP** easier to be influenced by the noise compared with the proposed $L_{IRG\text{-}t}$.

In conclusion, though both **FSP** and the proposed $L_{IRG\text{-}t}$ extract knowledge from the feature space transformation, $L_{IRG\text{-}t}$ not only achieves competitive performance, but also takes much less time to train (see Table 1).

2. More Details on Training Complexity

Due to the proposed method introduces three types of knowledge and the corresponding hint loss functions to the knowledge distillation framework, the significant performance gain may be obtained at the expense of higher training complexity. However, according to our experiments in Table 1, the additional training time and GPU memory cost are limited, especially for larger dataset scenarios.

Concretely, on CIFAR10/100, the proposed method takes around 4 times of training time of the standard method (KD). On ImageNet, all the proposed methods have similar training time and the proposed time only takes 5% more time than KD. For the overall training process, the proposed method takes 3-4 hours longer than that of KD. In addition, for both scenarios, the additional GPU memory is less than 100M. Therefore, the extra training time and GPU memory cost are acceptable for small-scale dataset while they are negligible for large dataset.

3. More Details on Hyper-parameter Tuning

The additional loss functions bring more hyper-parameters to be tuned. There are three penalty coefficients including λ_1 and λ_2 for L_{IRG}, and λ_3 for $L_{IRG\text{-}t}$. During hyper-parameter tuning, λ_2 is the main parameter to be tuned. The other two λ's are easier to be tuned because the performance is not very sensitive to them.

In order to reduce the search space, we take a greedy strategy for hyper-parameter search. To be specific, we first conduct experiments to select the best λ_1. Second, we fix λ_1 and search the best λ_2. Finally, λ_3 is selected based on the best λ_1 and λ_2. This strategy significantly reduces the search space and helps find appropriate hyper-parameters in short time. Therefore, the additional hyper-parameters do not increase the difficulty of tuning.

References

[1] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[2] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[3] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[4] G. Zhou, Y. Fan, R. Cui, W. Bian, X. Zhu, and K. Gai. Rocket launching: A universal and efficient framework for training well-performing light net. In *AAAI Conference on Artificial Intelligence*, volume 1050, page 8, 2018.