Knowledge Distillation via Instance Relationship Graph

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

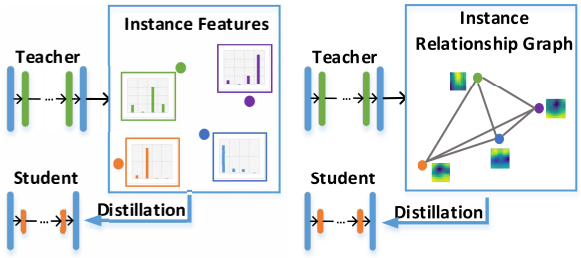
[a]*NLPR, Institute of Automation, Chinese Academy of Sciences*
[b]*Ant Financial*
[c]*National Computer Network Emergency Response Technical Team/Coordination Center of China*

Abstract

The key challenge of knowledge distillation is to extract general, moderate and sufficient knowledge from a teacher network to guide a student network. In this paper, a novel Instance Relationship Graph (IRG) is proposed for knowledge distillation. It models three kinds of knowledge, including instance features, instance relationships and feature space transformation, while the latter two kinds of knowledge are neglected by previous methods. Firstly, the IRG is constructed to model the distilled knowledge of one network layer, by considering instance features and instance relationships as vertexes and edges respectively. Secondly, an IRG transformation is proposed to models the feature space transformation across layers. It is more moderate than directly mimicking the features at intermediate layers. Finally, hint loss functions are designed to force a student's IRGs to mimic the structures of a teacher's IRGs. The proposed method effectively captures the knowledge along the whole network via IRGs, and thus shows stable convergence and strong robustness to different network architectures. In addition, the proposed method shows superior performance over existing methods on datasets of various scales.

1. Introduction

In pursuit of high performance of Deep Neural Networks (DNNs), deeper and wider architectures have been proposed at the expense of larger model size and longer inference time. Examples include from AlexNet [13, 11, 2] to ResNet [7, 27, 21] and DenseNet [10]. However, in various practical applications, these networks can not satisfy the requirements of real-time response and low memory cost. Therefore, more and more efforts have been putting into model compression.



(a) Conventional method (b) Our method

Figure 1: Comparison between the conventional and proposed methods. (a) The conventional method uses instance features to guide the student. Each instance is an independent point in the feature space. (b) The proposed method defines Instance Relationship Graph (containing instance features, instance relationships and feature space transformation) as the distilled knowledge to guide the student.

Knowledge distillation [9, 26, 16, 25] is one of the most popular solutions for model compression. It utilizes a teacher-student framework to distill knowledge, such as predicted probabilities, from a teacher network to guide a student network. For example, Hinton *et al.* [9] leveraged the final predicted probabilities of the teacher network to supervise the student network. Zagoruyko *et al.* [26] transferred attention maps distilled from some mid-level layers to teach the student. We refer to the softened outputs or the intermediate-layer features of the network as *instance features*, since they are obtained from samples (also called instances) independently.

Nevertheless, there exist two limitations in conventional knowledge distillation methods. Firstly, the existing methods independently extract instance features from the teacher network as the distilled knowledge (as shown in Figure 1(a)). The instance relationships are never considered, but they help reduce the intra-class variations and enlarge the inter-class differences in the feature space. Moreover, instance features based methods usually suffer from signifi-

*Both authors contributed equally to this research.
†Corresponding authors: Bing Li (bli@nlpr.ia.ac.cn), Chunfeng Yuan (cfyuan@nlpr.ia.ac.cn)

cant performance drop when the teacher and student have different network architectures. On the contrary, the instance relationships are more robust to network changes. For example, the instance features of the same sample from two teacher networks can be totally different, while for both teachers, samples from the same class are often closer than those from the different classes in the feature space. Secondly, these methods only distill some specific layers' outputs of the teacher, without considering the inference procedure. It is a hard constraint for the student to directly fit all these layers' outputs of the teacher. Thus extracting moderate knowledge from the overall inference procedure is necessary.

In order to resolve the above limitations, a novel graph-based knowledge distillation method is proposed. It distills three kinds of knowledge along the whole network. Aside from the widely used instance features, two kinds of new knowledge including instance relationships and feature space transformation are defined. An Instance Relation Graph (IRG) is proposed to model the knowledge. Specifically, for a DNN layer, an IRG is constructed, in which the vertex of the IRG represents the instance features, and the edge denotes the instance relationships (as shown in Figure 1(b)). The instance relationships provide sufficient and general information of the feature distribution and make the distilled knowledge be able to guide a student network with different architectures from its teacher. In order to avoid forcing too tight constraints, the feature space transformation across layers is introduced as the third type of the knowledge and an IRG transformation is proposed to model this knowledge. The feature space transformation is a more relaxed description than the densely fitting on teacher's instance features at intermediate layers. By combining IRG and IRG transformation, the proposed method models more general, moderate and sufficient knowledge than the existing methods. Finally, two loss functions are designed for IRG and IRG transformation respectively. The hint losses are optimized together to help boost the performance of the student model.

Experiments on 4 different datasets are conducted, under different teacher-student architectures. The experimental results demonstrate that the proposed method shows stable improvement on different teacher-network pairs, and outperforms the state-of-the-art by more than **1x**. In summary, the main contributions of our work are three-fold:

- To the best of our knowledge, we at the first time exploit three kinds of knowledge for knowledge distillation including instance features, instance relationships, and feature space transformation across layers.
- An IRG and its transformation are proposed to model all the types of the knowledge. The instance features and instance relationships are considered as the vertexes and edges of the IRG respectively. The feature space transformation is naturally expressed as the IRG transformation from one layer to another. Therefore, all three kinds of the knowledge of a network can be well represented via IRGs.
- Different hint losses are introduced to supervise the training of the student network. They help the student learn different kinds of knowledge preserved in IRGs. The experimental results have shown the superior of the proposed method.

2. Related Work

There are mainly two types of methods on model compression. The first type is to remove redundant information from complex trained models, such as network pruning and model quantization. Specifically, network pruning [14, 15, 8, 17, 22] aims to delete unimportant connections of the trained network, while model quantization methods [5, 18, 3, 23] represent the float weights with fewer bits. Though pruning and quantization methods have achieved high compression ratio with low performance loss, they can not change the network architectures.

Different from the first type, a new concept called knowledge distillation is introduced by Hinton *et al.* [9] based on a teacher-student framework. Knowledge distillation method transfers knowledge from the trained teacher to the student network. Recently, it has been applied in many areas, such as image classification [13], scene recognition [29] and face verification [20].

Existing knowledge distillation methods focus on transferring instance features from the teacher to the student. For example, Ba *et al.* [1] trained the student network to mimic the teacher via regressing logits before the *Softmax* layer. Zhou *et al.* [30] made the student share some lower layers with the teacher and train them simultaneously, but they also used logits as the distilled knowledge. For transferring the instance features of intermediate layers, Romero *et al.* [19] proposed FitNet, which extracted the feature maps of the intermediate layer as well as the final output to teach the student network. After that, Zagoruyko *et al.* [26] defined Attention Transfer (AT) based on attention maps to improve the performance of the student network. However, these methods independently extract the instance features from the teacher while the instance relationships in the feature space is barely considered. Moreover, the instance features of the intermediate layers are closely related to the network design, which is not general for different teacher-student pairs.

Further, most methods directly teach the student to fit the instance features of the teacher, ignoring the feature space transformation process. To address this issue, Yim *et al.* [24] presented Flow of Solution Procedure (FSP) to transfer the inference procedure of the teacher rather than the intermediate layer results. The FSP matrix is actually the inner

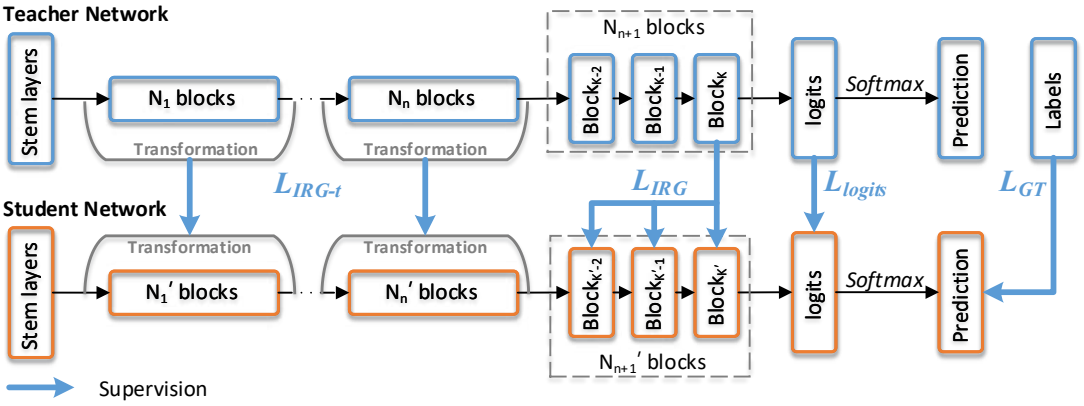


Figure 2: Overall framework of the proposed method.

production of the feature channels from two layers, which is regarded as the flow for solving a problem. However, the FSP matrix can only be computed between two layers with the same output resolution. Besides, the computational cost of FSP is rather high.

3. Proposed Method

In this section, the overall framework of the proposed method is firstly introduced. Then a knowledge graph called IRG and its transformation are constructed for representing **general, moderate and sufficient** knowledge. Subsequently, the hint losses about IRG and its transformation are formulated to utilize the mined knowledge. Finally, the overall loss is formulated based on the previous loss functions to supervise the training of the student network.

The overall framework of the proposed method is illustrated in Figure 2. The upper blue network is the teacher network, while the lower orange network is the student network. Except the *SoftmaxLoss* L_{GT} from the ground truth, three supervision signals are added to transfer the distilled knowledge, including L_{IRG}, L_{logits} and $L_{IRG\text{-}t}$. All of the three signals are derived from IRG, which represents the feature space of a certain layer. Specifically, L_{IRG} is used to transfer the instance features and the instance relationships. L_{logits} represents the instance features and is a special case of L_{IRG}. It can be absorbed into L_{IRG}. And $L_{IRG\text{-}t}$ distills the feature space transformation knowledge. Eventually, the three loss functions make up Multi-Type Knowledge (MTK) loss (L_{MTK}), which transfers all the three types of knowledge from the teacher to the student.

3.1. Instance Relationship Graph

Given I training instances $\boldsymbol{x} = \{x_i\}_{i=1}^{I}$, let $f_l(x_i)$ be the instance features of x_i at l-th layer, which can be the final softened outputs [9] or the feature maps [26]. The instance relationships are formulated as an adjacent matrix of the instance features, referring to as $\mathbf{A}_l$. An exmaple of IRG is shown in Figure 3(a). Then an IRG denoted as IRG_l is constructed to represent the feature space of the l-th layer, expressed as

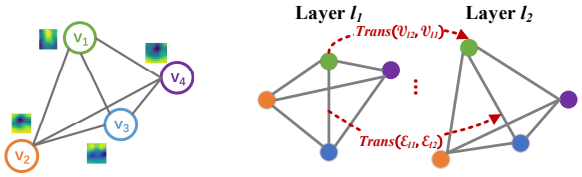


Figure 3: Structure of IRG. (a) An example of IRG. (b) An example of IRG transformation.

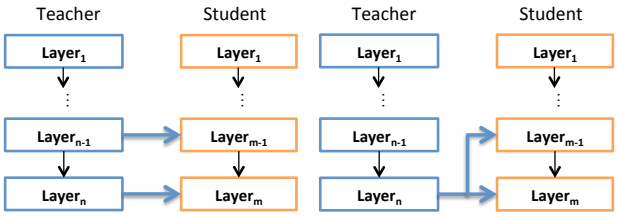
$$\begin{aligned} IRG_l &= (\mathcal{V}_l, \mathcal{E}_l) = (\{f_l(x_i)\}_{i=1}^{I}, \mathbf{A}_l), \\ \mathbf{A}_l(i,j) &= ||f_l(x_i) - f_l(x_j)||_2^2, \ i,j = 1,...,I, \end{aligned} \tag{1}$$

where $\mathcal{V}_l$ is the vertex set of IRG representing the instance features at the l-th layer, $\mathcal{E}_l$ is the edge set of IRG representing the instance relationship. Each element of the feature relationship matrix, $\mathbf{A}_l$, represents an edge. And each edge is defined as the Euclidean distance between the instance features of two linked instances as shown in Equation 1.

Based on the formulation of IRG, its transformation is defined. Let $IRG\text{-}t_{l_1 l_2}$ be the IRG transformation from the l_1-th layer to the l_2-th layer. As shown in Figure 3(b), it is natural to decompose $IRG\text{-}t_{l_1 l_2}$ into the vertex transformation (or called the instance feature transformation) $Trans(\mathcal{V}_{l_1}, \mathcal{V}_{l_2})$ and the edge transformation (or called the instance relationship transformation) $Trans(\mathcal{E}_{l_1}, \mathcal{E}_{l_2})$, namely

$$\begin{aligned} IRG\text{-}t_{l_1 l_2} &= Trans(IRG_{l_1}, IRG_{l_2}) \\ &= (Trans(\mathcal{V}_{l_1}, \mathcal{V}_{l_2}), Trans(\mathcal{E}_{l_1}, \mathcal{E}_{l_2})) \\ &= (\mathbf{\Lambda}_{l_1,l_2}, \mathbf{\Theta}_{l_1,l_2}), \\ \mathbf{\Lambda}_{l_1,l_2}(i,i) &= ||f_{l_1}(x_i) - f_{l_2}(x_i)||_2^2, \quad i = 1,...,I, \\ \mathbf{\Theta}_{l_1,l_2} &= ||\mathbf{A}_{l_1} - \mathbf{A}_{l_2}||_2^2, \end{aligned} \tag{2}$$

where $Trans(\cdot)$ is the transformation function, $\mathbf{\Lambda}_{l_1,l_2}$ and $\mathbf{\Theta}_{l_1,l_2}$ are the vertex transformation matrix and edge transformation matrix, respectively. As shown in Equation 2, each element of $\mathbf{\Lambda}_{l_1,l_2}$ represents the instance feature transformation of the same instance x_i from one layer to another. Similarly, $\mathbf{\Theta}_{l_1,l_2}$ is defined as the Euclidean distance between the two relationship matrixes $\mathbf{A}_{l_1}$ and $\mathbf{A}_{l_2}$.



(a) One-to-one mode for L_{IRG} (b) One-to-many mode for L_{IRG}

Figure 4: Two possible deploy modes for L_{IRG}.

Then $IRG\text{-}t_{l_1 l_2}$ contains the knowledge of the feature space transformation from the l_1-th layer to the l_2-th layer.

3.2. Loss for IRG

The loss L_{IRG} is defined as the difference between the teacher's IRG and the student's. Let IRG_L^T be the IRG of the teacher network at the L-th layer. Similarly, $IRG_{l_M}^S$ is the l_M-th layer's IRG in the student network. The formulations of the two IRGs follow Equation 1. Then, the difference of the two IRGs is divided into the difference of the vertexes $Dist(\mathcal{V}_L^T, \mathcal{V}_{l_M}^S)$ and the difference of the edges $Dist(\mathcal{E}_L^T, \mathcal{E}_{l_M}^S)$. Both parts are evaluated by Euclidean distance as follows:

$$\begin{aligned} &L_{IRG}(\boldsymbol{x}) = Dist(IRG_L^T, IRG_{l_M}^S) \\ &= \lambda_1 \cdot Dist(\mathcal{V}_L^T, \mathcal{V}_{l_M}^S) + \lambda_2 \cdot Dist(\mathcal{E}_L^T, \mathcal{E}_{l_M}^S) \\ &= \lambda_1 \cdot \sum_{i=1}^{I} ||f_L^T(x_i) - f_{l_M}^S(x_i)||_2^2 \\ &+ \lambda_2 \cdot ||\mathbf{A}_L^T - \mathbf{A}_{l_M}^S||_2^2. \end{aligned} \tag{3}$$

Note that λ_1 and λ_2 are the penalty coefficients balanced the two terms. **Most previous works only considering the instance features, and they can be regarded as a special case of the IRG-based method by setting λ_2 to be zero.**

To fully utilize the effectiveness of L_{IRG} when applying it to a specific task, there are two factors that may influence the performance.

First, there are two possible deploy modes for L_{IRG} as shown in Figure 4. In particular, under **one-to-one mode**, the selected layers of the student is supervised by the corresponding layers of the teacher network. It is obvious that one-to-one mode performs the best when the teacher and student shares the network structure. On the other hand, **one-to-many mode** utilizes the last layer of the teacher (L) to guide the selected layers (L_M) of the student. Since the last layer usually learns the general distribution of the dataset, IRG of the last layer is less correlated with the network design. Since one-to-many mode extracts more **general** knowledge, the formulation of L_{IRG} in Equation 4 follows this mode.

Second, the vertex difference $Dist(\mathcal{V}_L^T, \mathcal{V}_{l_M}^S)$ in Equation 3 can be computed only if $f_L^T(x_i)$ and $f_{l_M}^S(x_i)$ have the same feature resolution and feature channel number. However, this can not be satisfied under most (L, l_M) combinations, which indicates the edge difference is not a **general** type of knowledge. Further, adopting the knowledge distillation densely for intermediate layers is not a **moderate** constraint for the student. Therefore, the vertex difference is only deployed for the logits layers. Consequently, L_{IRG} in this work is obtained as follows:

$$L_{IRG}(\boldsymbol{x}) = \lambda_1 \cdot L_{logits}(\boldsymbol{x}) + \lambda_2 \cdot \sum_{l_M \in \mathbf{L_M}} ||\mathbf{A}_L^T - \mathbf{A}_{l_M}^S||_2^2. \tag{4}$$

3.3. Loss for IRG Transformation.

IRG transformation is the representation of the instance feature space transformation, consisting of vertex transformation and edge transformation. Therefore, the loss $L_{IRG\text{-}t}$ also contains two parts as shown as follows:

$$\begin{aligned} &L_{IRG\text{-}t}(\boldsymbol{x}) \\ &= Dist(IRG\text{-}t_{l_1 l_2}^T, IRG\text{-}t_{l_3 l_4}^S) \\ &= Dist(Trans(\mathcal{V}_{l_1}^T, \mathcal{V}_{l_2}^T), Trans(\mathcal{V}_{l_3}^S, \mathcal{V}_{l_4}^S)) \\ &+ Dist(Trans(\mathcal{E}_{l_1}^T, \mathcal{E}_{l_2}^T), Trans(\mathcal{E}_{l_3}^S, \mathcal{E}_{l_4}^S)) \\ &= ||\mathbf{\Lambda}_{l_1,l_2}^T - \mathbf{\Lambda}_{l_3,l_4}^S||_2^2 + ||\mathbf{\Theta}_{l_1,l_2}^T - \mathbf{\Theta}_{l_3,l_4}^S||_2^2, \end{aligned} \tag{5}$$

where $\mathbf{\Lambda}_{l_1,l_2}^T$ and $\mathbf{\Theta}_{l_1,l_2}^T$ are the vertex and edge transformation of the teacher from the l_1-th layer to the l_2-th layer, while $\mathbf{\Lambda}_{l_3,l_4}^S$ and $\mathbf{\Theta}_{l_3,l_4}^S$ together represent the feature space transformation of the student. Then $||\mathbf{\Lambda}_{l_1,l_2}^T - \mathbf{\Lambda}_{l_3,l_4}^S||_2^2$ and $||\mathbf{\Theta}_{l_1,l_2}^T - \mathbf{\Theta}_{l_3,l_4}^S||_2^2$ are adopted to evaluate the vertex transformation difference and the edge transformation difference between the teacher and the student. Similar to L_{IRG}, there is also an important factor influencing the performance of $L_{IRG\text{-}t}$.

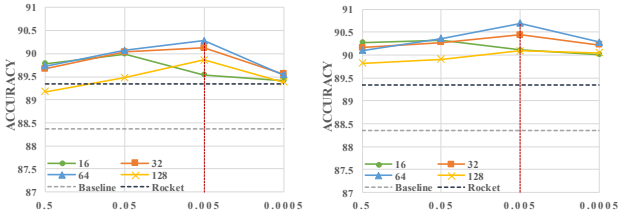
The edge transformation part consumes much more computation resources compared with the vertex part. To be specific, for an IRG with I vertices, time complexity of the vertex part is $O(I)$, while that of the edge part is $O(I^2)$. In addition, the distilled knowledge of the vertex transformation and the edge transformation is redundant. Therefore, the edge part of IRG transformation loss is omitted for the sake of effectiveness. Finally, the resulting IRG transformation loss function is formulated as follows:

$$L_{IRG\text{-}t}(\boldsymbol{x}) = ||\mathbf{\Lambda}_{l_1,l_2}^T - \mathbf{\Lambda}_{l_3,l_4}^S||_2^2. \tag{6}$$

3.4. Multi-Type Knowledge Loss

We define a MTK loss (L_{MTK}) to train the student network.It is formulated based on the *SoftmaxLoss* for Ground Truth (GT) (L_{GT}), loss for IRG (L_{IRG}) and loss for IRG transformation ($L_{IRG\text{-}t}$) as follows:

$$\begin{aligned} &L_{MTK}(\boldsymbol{x}) \\ &= L_{GT}(\boldsymbol{x}) + L_{IRG}(\boldsymbol{x}) + \lambda_3 \cdot L_{IRG\text{-}t}(\boldsymbol{x}) \\ &= L_{GT}(\boldsymbol{x}) + \lambda_1 \cdot L_{logits}(\boldsymbol{x}) \\ &+ \lambda_2 \cdot \sum_{l_M \in \mathbf{L_M}} ||\mathbf{A}_L^T - \mathbf{A}_{l_M}^S||_2^2 \\ &+ \lambda_3 \cdot \sum_{l_1 l_2 l_3 l_4 \in \mathbf{L_{Tran}}} ||\mathbf{\Lambda}_{l_1,l_2}^T - \mathbf{\Lambda}_{l_3,l_4}^S||_2^2, \end{aligned} \tag{7}$$



(a) (b)

Figure 5: (a) Performance of L_{IRG}. (b) Performance of L_{MTK}.

in which λ_1, λ_2 and λ_3 are the three penalty coefficients, while $\mathbf{L_M}$ and $\mathbf{L_{Tran}}$ represent the layer set for IRG and its transformation, respectively. Using the MTK loss, the student network can be optimized to acquire all the three types of knowledge from the teacher network.

4. Experiments

4.1. Ablation Analysis

In this section, experiments are conducted to verify the effectiveness of L_{IRG} and $L_{IRG\text{-}t}$. The detailed experimental settings are as below.

4.1.1 Experiment Settings

CIFAR10 [12] is adopted as our training and test dataset for ablation analysis. Images are first padded to 36×36 and then cropped to 32×32 for training. ResNet20 [6] or ShuffleNet-x0.5 [28] is adopted as the teacher network, while we reduce the channels of ResNet20 by half to obtain the student network named ResNet20-x0.5. Note that the "Baseline" is the ResNet20-x0.5 directly trained by L_{GT}.

4.1.2 The Effectiveness of $\mathbf{L_{IRG}}$

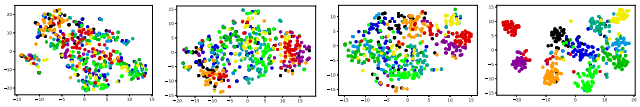
The hyper-parameters of L_{IRG} are first decided according to the experiments. After this, two deploy modes including one-to-one mode and one-to-many mode are compared and analyzed.

(1) Hyper-parameter Tuning. Besides the coefficient λ_2, batch size is also a crucial hyper-parameter, since the instance relationship matrix, namely $\mathbf{A}_l$, is computed by a batch of instances (see Section 3.1). $\mathbf{A}_l$ with a larger batch size contains more instance relationships as well as more comprehensive knowledge. In the meantime, it may be a harder regularization for the student. In order to achieve a trade-off between extracting **moderate** knowledge and extracting **sufficient** knowledge, experiments are conducted under different settings as shown in Figure 5(a). It can be seen that L_{IRG} outperforms the baseline and *Rocket* for most of the cases. According to the results, we choose batch size as 64 and λ_2 as 0.005 for L_{IRG} for the rest of the paper.

(2) Performance Analysis of One-to-one Mode and One-to-many Mode. Under one-to-one mode, one layer's IRG from the teacher is used to guide a corresponding layer's IRG of the student (as shown in Figure 4(a)). For one-to-many mode, last layer's IRG is selected as the supervision for the student's last several layers (as shown in Figure 4(b)).

Table 1: Student performance under different modes of L_{IRG}. O2O refers to one-to-one mode, while O2M refers to one-to-many mode.

Student/Teacher: ResNet20-x0.5(88.36) / ResNet20(91.45)					
O2O 1layer	89.87	O2O 3layers	**90.03**	O2O 5layers	89.65
O2M 1layer	89.87	O2M 3layers	**90.28**	O2M 5layers	90.02
Student/Teacher: ResNet20-x0.5(88.36) / ShuffleNet-x0.5(91.47)					
O2O 1layer	89.83	O2O 3layers	**89.89**	O2O 5layers	89.50
O2M 1layer	89.83	O2M 3layers	**90.21**	O2M 5layers	89.93



(a) Eltwise1 (b) Eltwise5 (c) Eltwise8 (d) Eltwise9

Figure 6: Feature visualizations at different layers of the teacher network.

Experiments are conducted under different configurations for both modes. The results are shown in Table 1. For example,"O2M 3layer" (one-to-many mode with 3 layers) refers to the situation that teahcer's last Eltwise layer (Eltwise9) is selected to supervise the last 3 Eltwise layers (Eltwise7-9) of the student, while"O2O 3layer" (one-to-one mode with 3 layers) refers to the situation that teacher's last 3 Eltwise layers supervise the corresponding 3 layers of the student, respectively.

According to the results, both one-to-one model and one-to-many mode outperform baseline by a significant margin while one-to-many mode continuously outperforms one-to-one mode. Figure 6 visualizes the feature maps of the teacher at different layers. It can be observed that deeper layers learn more discriminative and general features, and the last Eltwise layer with the best discrimination is the most suitable supervision for the student network. Therefore, one-to-many mode, which always extracts knowledge from teacher's Eltwise9 layer, forces all the supervised layers of the student to benefit from the discriminative feature space and thus outperforms one-to-one mode.

Furthermore, one-to-many mode is more robust to the teacher-student pair changes. As shown in Table 1, when the teacher network changes from ResNet20 to ShuffleNet-x0.5, one-to-one mode suffers from performance drop while the performance of one-to-many mode is rather stable. It is because Eltwise9 learns the general distribution of the dataset, which is less related to the network architecture. On the contrary, the feature space of the shallower layers such as Eltwise7 are closely related to the network architecture. Thus when the teacher has a totally different design from that of the student, one-to-one mode performs much worse

Table 2: Model performance of different methods. Performance gain over the best competing method is marked in the brackets.

	CIFAR10	CIFAR100 coarse	CIFAR100 fine
Baseline	88.36	72.51	59.88
KD	89.09	73.03	60.21
FSP	89.21	73.18	60.46
AT	89.15	73.15	60.58
$Rocket^{\dagger}$	89.35	73.39	60.88
L_{IRG}	90.28 **(0.93)**	74.32 **(0.93)**	61.93 **(1.05)**
L_{MTK}	**90.69** **(1.34)**	**74.64** **(1.25)**	**62.25** **(1.37)**
Teacher	91.45	78.40	68.42

than one-to-many mode. For the rest of the paper, "O2M 3layer" mode is always adopted for L_{IRG}, which exceeds the baseline by $\mathbf{1.92}\%$.

4.1.3 The Effectiveness of $\mathbf{L_{IRG\text{-}t}}$

Besides the instance features and instance relationships stored in IRG, the transformation of IRGs is also an important type of knowledge. Therefore, $L_{IRG\text{-}t}$ and L_{IRG} are combined to obtain L_{MTK}. By comparing the performance of L_{IRG} and L_{MTK}, the effectiveness of $L_{IRG\text{-}t}$ can be verified.

(1) Hyper-parameter Tuning. Just as L_{IRG}, the regularization strength of $L_{IRG\text{-}t}$ is controlled by batch size and λ_3. The relationship between the two factors and accuracy is shown in Figure 5(b). $L_{IRG\text{-}t}$ is less sensitive to batch size as well as the penalty factor, compared with L_{IRG}. **Therefore, it takes limited time to find an appropriate λ_3 for L_{MTK}.** Consequently, based on the results in Figure 5(b), we choose batch size as 64 and λ_3 as 0.005 for $L_{IRG\text{-}t}$ for the rest of the paper.

(2) Performance Analysis of $\mathbf{L_{IRG\text{-}t}}$. $L_{IRG\text{-}t}$ considers the transformations of multiple pairs of layers. In particular, for ResNet20, three pairs of layers are utilized as supervisions, each of which represents the feature space transformation under a certain feature map resolution. In this way, the overall $L_{IRG\text{-}t}$ boosts the feature learning process from the beginning of the network to its end, so as to reinforce the model performance.

As shown in Figure 5, L_{MTK} continuously outperforms L_{IRG}, which indicates the effectiveness of $L_{IRG\text{-}t}$. In particular, L_{MTK} achieves an accuracy of $\mathbf{90.69}\%$ on CIFAR10, obtaining a performance gain of $\mathbf{0.41}\%$ ($\mathbf{2.33}\%$) over L_{IRG} (baseline). Furthermore, as shown in Figure 8, with the help of $L_{IRG\text{-}t}$, L_{MTK} shows more stable convergence on the test loss and accuracy. It is because $L_{IRG\text{-}t}$ considers the global information flow of the network and is a more moderate constraint.

4.2. Performance Comparisons

In this section, we compare the proposed method with 4 state-of-the-arts, including **KD** [9], **FSP** [24], **AT** [26] and **Rocket** [30] (using logits as the distilled knowledge). First, the performance of different methods is evaluated on CIFAR10, CIFAR100-coarse and CIFAR100-fine. Secondly, different teacher-student pairs are implemented to evaluate the methods' generalization ability on network architectures. Finally, we particularly conduct experiments on ImageNet and a subset of CIFAR10, called CIFAR10-small to show the superiority of the proposed method on datasets with different scales. The detailed experiment settings are as below.

4.2.1 Experiment Settings

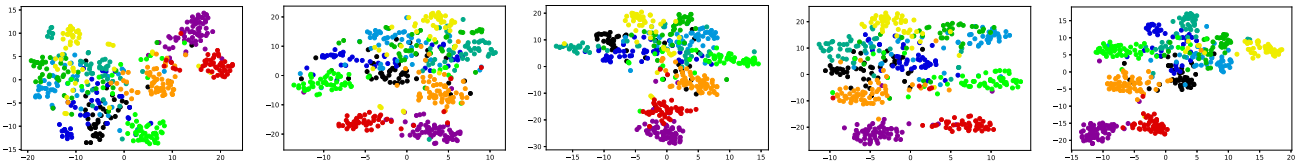
CIFAR10, CIFAR100 [12], ImageNet [4] and CIFAR10-small are used for performance evaluation. Note that 10% of CIFAR10 are randomly sampled to obtain CIFAR10-small. Two types of teacher networks and three types of student networks are used for performance evaluation. Specifically, ResNet20 and ShuffleNet-x0.5 [28] are the two teachers. Besides ResNet20-x0.5, ResNet20-x0.375 and ResNet14-x0.5 are the possible student networks. ResNet14-x0.5 is obtained by reducing 3 residual blocks from ResNet20-x0.5, while ResNet20-x0.375 has 0.375 time of channels of ResNet20.

4.2.2 Evaluation on CIFAR10 and CIFAR100

CIFAR10 and CIFAR100 are two typical datasets for knowledge distillation evaluation. In this section, ResNet20 and ResNet20-x0.5 are used as the teacher network and the student network respectively. When training the networks, images of CIFAR10 (CIFAR100) are first padded to 36×36 and then cropped to 32×32. Furthermore, the training-test division strictly follows the official protocol.

According to the results in Table 2, the proposed method significantly outperforms all the competing methods. To be specific, L_{IRG} outperforms $Rocket$, the best competing method, by $\mathbf{0.93}\%$ to $\mathbf{1.05}\%$ on different datasets. By taking both IRG and IRG transformations into consideration, L_{MTK} outperforms $Rocket$ by a larger margin from $\mathbf{1.25}\%$ to $\mathbf{1.37}\%$. Since the performance gain of $Rocket$ over the baseline is 0.77% to 1.0%, the proposed method ($\mathbf{1.60}\%$ to $\mathbf{2.43}\%$) **doubles** this performance gain.

We attribute the significant performance improvement to L_{MTK}'s extraction of all the three types of knowledge. The previous methods only consider a subset of knowledge. For example, KD, AT and $Rocket$ only extract instance features from the teacher, while FSP only takes feature transformation into consideration. Thus, these competing methods are all special cases of our method. Further, none of them uses instance relationships as the distilled knowledge. According to the experiments, the instance relationships not only extract sufficient knowledge from the teacher, but also make the knowledge distillation process more robust to the network designs. In addition, the loss functions making up



(a) KD (b) AT (c) FSP (d) *Rocket* (e) L_{MTK}

Figure 7: Feature visualizations of Eltwise9 layer for different methods. Each color represents a class, best viewed in color.

Table 3: Model performance of different teacher-student pairs. Note that $Rocket^{\dagger}$ adopts logits as the distilled knowledge and shares lower layers of the teacher and the student. Therefore, the model size of $Rocket^{\dagger}$ is a little larger than the reported one. The number in the brackets shows the performance increase over the best competing method.

Dataset	Teacher Net.	Student Net.	Baseline	KD	FSP	AT	$Rocket^{\dagger}$	L_{IRG}	L_{MTK}	Teacher
CIFAR10	ResNet20 (1.06M)	ResNet20-x0.5 (0.28M)	88.36	89.09	89.17	89.29	89.45	90.28 (0.93)	**90.69** (1.34)	91.45
	ResNet20 (1.06M)	ResNet14-x0.5 (0.18M)	86.65	87.01	87.23	87.12	87.53	88.55 (1.02)	**89.08** (1.55)	91.45
	ResNet20 (1.06M)	ResNet20-x0.375 (0.16M)	86.54	87.23	87.11	87.39	87.67	88.52 (0.85)	**89.01** (1.34)	91.45
	ShuffleNet-x0.5 (0.94M)	ResNet20-x0.5 (0.28M)	88.36	89.12	89.07	89.05	89.22	90.29 (1.07)	**90.65** (1.43)	91.47
CIFAR100-coarse	ResNet20 (1.06M)	ResNet20-x0.5 (0.28M)	72.51	73.03	73.18	73.15	73.39	74.32 (0.93)	**74.64** (1.25)	78.40
	ResNet20 (1.06M)	ResNet14-x0.5 (0.18M)	68.55	68.76	68.73	68.69	69.07	69.94 (0.87)	**70.18** (1.11)	78.40
	ResNet20 (1.06M)	ResNet20-x0.375 (0.16M)	66.72	66.98	67.07	67.22	67.45	68.26 (0.81)	**68.57** (1.12)	78.40
	ShuffleNet-x0.5 (0.94M)	ResNet20-x0.5 (0.28M)	72.51	72.96	72.87	72.99	73.27	74.22 (0.95)	**74.56** (1.29)	78.69
CIFAR10-fine	ResNet20 (1.06M)	ResNet20-x0.5 (0.28M)	59.88	60.21	60.46	60.58	60.88	61.93 (1.05)	**62.25** (1.37)	68.42
	ResNet20 (1.06M)	ResNet14-x0.5 (0.18M)	56.23	56.44	56.34	56.26	56.55	57.44 (0.89)	**57.68** (1.13)	68.42
	ResNet20 (1.06M)	ResNet20-x0.375 (0.16M)	53.87	54.09	54.24	54.38	54.52	55.37 (0.85)	**55.66** (1.14)	68.42
	ShuffleNet-x0.5 (0.94M)	ResNet20-x0.5 (0.28M)	59.88	60.15	60.23	60.31	60.97	61.83(0.86)	**62.06** (1.09)	68.67

L_{MTK} are carefully designed, so as to achieve a good trade-off among generalization, sufficiency and moderation of the distilled knowledge. Therefore, L_{IRG} and $L_{IRG\text{-}t}$ are complementary to each other, boosting the performance in harmony.

Figure 7 visualizes the distribution of Eltwise9 layer of the student networks from different methods. The feature space of L_{MTK} is significantly more separable than those of the other methods, especially on class boundaries. Moreover, different classes are well clustered with smaller inner-class variation and larger inter-class variation. L_{MTK} makes usage of three kinds of knowledge from the teacher network, which enables it to learn more compact and discriminative representations. On the contrary, previous methods only utilizes single type of knowledge, thus the expression ability of the student network is limited.

4.2.3 Evaluation on Different Networks

In this subsection, different teacher-student pairs are explored. The experimental results are reported in Table 3. Under different network settings, the proposed method continuously outperforms the other methods. And we find that the robustness to the network designs is related to the type of the distilled knowledge.

First, some of the methods are more sensitive to the changes of the teacher-student pairs. For instance, FSP and AT perform worse when the teacher and student have different network architectures. It is because FSP and AT extract network-related knowledge. FSP extracts feature space transformation knowledge while AT extracts attention maps of middle layers. The network-related knowledge is hard to transform from the teacher to a student with a different network design.

Second, KD, Rocket, L_{IRG} and L_{MTK} are relatively more robust to the teacher-student pair changes. For example, though ResNet20 and ShuffleNet-x0.5 have totally different architectures, all the four methods perform stably when the teacher network changes. It is because these methods utilize the knowledge that is not closely related to the network architecture. Specifically, KD and Rocket distill the predicted class probabilities of the teacher, while L_{IRG} learns instance relationship. Since the class probabilities and the learned instance relationships are usually stable, KD, Rocket and L_{IRG} are able to work robust to the network changes. As for L_{MTK}, which consists of three types of distilled knowledge, is also robust to different networks. When one of the type, for example, feature transformation type, works a little worse, the other two types still perform well. Thus the overall performance does not significantly decrease.

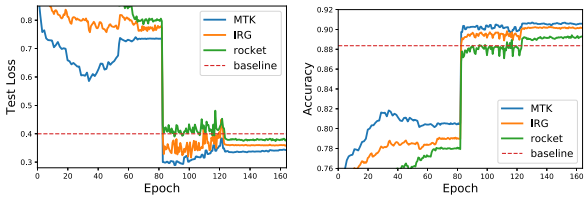
4.2.4 Evaluation on CIFAR10-small and ImageNet

To explore the effectiveness of the proposed method on datasets with different scales, experiments are conducted on CIFAR10-small and ImageNet. According to the results in Table 4, both L_{IRG} and L_{MTK} continuously outperform the competing methods, especially on CIFAR10-small.

CIFAR10-small. For real world applications, there are usually limited labeled images in hand. It is necessary to evaluate the model performance on small-scale dataset. Therefore, CIFAR10-small is constructed by randomly selecting 10% samples from the training set CIFAR10. Then

Table 4: Model performance on CIFAR10-small and ImageNet. We randomly select 10% of the training instances for 10 times and the average performance is reported.

Dataset	Teacher Net.	Student Net.	Baseline	KD	FSP	AT	$Rocket^{\dagger}$	L_{IRG}	L_{MTK}	Teacher
CIFAR10-small	ResNet20	ResNet14-x0.5	55.53	59.29	60.11	59.98	62.23	64.87 (2.64)	**66.04** (3.81)	91.45
	ResNet20	ResNet20-x0.375	57.32	62.83	63.21	63.52	64.14	66.96 (2.82)	**68.16** (4.02)	91.45
ImageNet	ResNet101-v2	ResNet18	70.83	71.43	71.28	71.58	71.93	72.68 (0.75)	**73.06** (1.13)	78.05
	ResNet101-v2	ResNext26	74.89	75.60	75.62	75.73	76.16	76.87 (0.71)	**77.18** (1.02)	78.05



(a) (b)

Figure 8: Test loss and accuracy comparisons.

all the student networks are trained on CIFAR10-small and the accuracy on the test set of CIFAR10 is reported in Table 4. Note that the teacher network is still trained on the full training set of CIFAR10.

According to the results, the performance gain (the numbers in the brackets) is tripled compared with the original CIFAR10 settings. We attribute it to the ability to extract sufficient knowledge of the proposed method. Previous works extract knowledge from independent instances from the teacher network. Thus, the amount of their knowledge is proportional to the number of training samples N. The knowledge is very limited when there are only a few training samples. L_{IRG} and L_{MTK}, by digging N instance features and N^2 instance relationships stored in IRG, extract much more knowledge from the teacher.

ImageNet. Experiments are conducted on ImageNet to show the efficienctiveness of the proposed method on large-scale dataset. Since ImageNet consists millions of high-resolution images, ResNet101-v2 is used as the teacher network while ResNet18 and ResNext-26 are introduced as the students. For training, images are first resized to 299×299 and randomly cropped to 224×224. As shown in Figure 3, L_{IRG} and L_{MTK} outperform competing methods by a significant margin. It indicates the proposed method is efficient on large-scale dataset.

4.2.5 Complexity Analysis

Since the proposed method computes IRGs and IRG transformations, it takes extra training time and GPU memory. In this section, the complexity of the algorithm is analyzed. According to the experiments, the additional resource cost is limited under different experimental settings. In particular, the extra time and memory are proportional to batch size and number of feature channels. In other words, once the batch size and feature channel are fixed, the additional training time and GPU memory is a constant. To be specific, for CIFAR10, it takes 3-4 hours to train a student with L_{MTK}, while the typical time of $Rocket$ is 1.2 hours and the typical time of baseline is around an hour. For ImageNet, L_{MTK} just takes around 4 hours more compared with the one-week baseline process. On the other hand, the extra GPU memory cost is around 100M for both CIFAR10 and ImageNet. Therefore, the proposed method can be easily deployed for real world applications with a little extra training resource cost but significant performance gain.

Though additional loss functions are introduced, the proposed method takes similar or less epochs to converge, compared with the best competing method. As shown in Figure 8, both L_{IRG} and L_{MTK} achieve lower test loss and higher accuracy compared with $Rocket$, under the same training configuration. Furthermore, L_{MTK} shows more stable convergence since $L_{IRG\text{-}t}$ introduces the moderate knowledge, namely feature space transformation.

5. Conclusion

We find that knowledge can be divided into three types: instance features, instance relationships and feature space transformation. However, most recent works only concentrate on the instance features. In this paper, an Instance Relationship Graph (IRG) is defined to preserve all the types of knowledge. IRG-based knowledge distillation method is then proposed and hint loss functions corresponding to different types of knowledge are presented to optimize the student network. The experiments verify that the proposed method shows strong robustness to teacher-student architecture changes. In addition, it shows superior performance on both big-scale and small-scale datasets over existing methods.

Acknowledgement. This work is partly supported by the National Key R&D Plan (Nos. 2017YFB1002801 and 2016QY01W0106), the Natural Science Foundation of China (Nos. U1803119, U1736106, 61751212, 61721004, 61772225 and 61876100), the NSFC-General Technology Collaborative Fund for Basic Research (Grant No. U1636218), the Key Research Program of Frontier Sciences, CAS (Grant No. QYZDJ-SSW-JSC040), Beijing Natural Science Foundation (Nos. JQ18018, L172051 and L182058) and the CAS External Cooperation Key Project. Bing Li is also supported by Youth Innovation Promotion Association, CAS.

References

[1] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[2] P. Ballester and R. M. de Araújo. On the performance of googlenet and alexnet applied to sketches. In *AAAI*, pages 1124–1128, 2016.

[3] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[5] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

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

[8] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *International Conference on Computer Vision (ICCV)*, volume 2, 2017.

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

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[12] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html (vi sited on Mar. 1, 2016)*, 2009.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[15] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2755–2763. IEEE, 2017.

[16] L. Lu, M. Guo, and S. Renals. Knowledge distillation for small-footprint highway networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4820–4824. IEEE, 2017.

[17] J.-H. Luo and J. Wu. An entropy-based pruning method for cnn compression. *arXiv preprint arXiv:1706.05791*, 2017.

[18] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

[19] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[20] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[22] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[23] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016.

[24] 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.

[25] R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[26] 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.

[27] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[28] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. 2017.

[29] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[30] 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.