Dropout Connects Transformers and CNNs: Transfer General Knowledge for Knowledge Distillation – Supplementary Materials –

Bokyeung Lee, Jonghwan Hong, Hyunuk Shin, Bonhwa Ku, and Hanseok Ko
Department of Electrical and Computer Engineering, Korea University
{bksain, jhong2661, hushin, hush999, and hsko}@korea.ac.kr

1. Understanding of General and Specific Knowledge

Specifically, we distinguish between 'general' and 'specific' knowledge in knowledge distillation training. General knowledge refers to the overall representation that both teacher and student models can generate through training. On the other hand, specific knowledge pertains to information that is uniquely represented by each model, influenced by inductive biases such as structure and model size—critical information for task achievement. Figure 1 in the main manuscript illustrates the problem setting of previous KD methods in Transformer-to-CNN KD. During the KD training stage, where representations of the teacher and student include both general and specific knowledge, KD methods aim to minimize the disparity between their representations, covering logits and feature maps. If specific knowledge isn't removed, the student network might imitate the teacher too closely, sacrificing its specific knowledge. This happens because the teacher and student don't have each other's specific knowledge, leading to a decline in the student's overall performance. Ideally, if the student absorbs only the teacher's general knowledge, it won't merely mimic but become a more generalized model. In CNN-to-CNN KD, specific knowledge is a small part of the capacity gap issue. In contrast, in Transformer-to-CNN KD, specific knowledge makes up a significant portion of the capacity gap problem.

In the knowledge distillation training stage, the teacher and student representation (feature maps and logits) can be divided into general and specific knowledge. General and specific knowledge can depend on the teacher and student. If the structure and size of the teacher and student are similar, general knowledge occupies a large part of the representation of the teacher and student. In contrast, when the structure and size of the teacher and student are dissimilar like Transformer-to-CNN knowledge distillation scenario, general knowledge occupies a small part of the representation of teacher and student and specific knowledge occupies a large part of the representation of the teacher and student.

Although it cannot be clearly distinguished, the knowledge of a network may consist of general and specific knowledge implicitly. For easier understanding, we can express the representation of the teacher and student:

$$R_T = G_T + S_T, \tag{1}$$

$$R_S = G_S + S_S, \tag{2}$$

where R_T and R_S are the representation of the teacher and student, G_T and G_S are general knowledge of the teacher and student, and S_T and S_S are specific knowledge of the teacher and student. To transfer the knowledge of the teacher to the student, we generally minimize the distance between the representations of the teacher and student R_T and R_S. G_S can be easily closer to G_T because the student network can train general knowledge through knowledge distillation training. However, the student network does not learn the specific knowledge of the teacher because the specific knowledge of the teacher network denotes unique information that the student network can not learn. Since knowledge distillation makes R_S closer to R_T, S_T prevents G_S from learning to get closer to G_T and makes S_S disappear. In order for G_S to learn only G_T, we need to remove specific knowledge of the teacher and student and then minimize the distance between G_T and G_S. Therefore, our motivation is to remove the specific knowledge of the teacher and student and minimize the distance between the two representations of the teacher and student.

2. Encoder, Decoder and Classifier Architecture

Figure 1 illustrates the architecture of the encoder, decoder, and classifier. The encoders, decoders, and classifiers of the teacher and student have the same architecture, and they do not share the weight.

3. CKA Calculation

To calculate CKA similarity between the teacher and student, we follow the previous works [4], and we extract the

Teacher	Student	Teacher	Student	Vanilla KD	AT	RKD	ReviewKD	DKD	DropKD
ResNet56	ResNet20	72.34	69.06	70.66	70.55	69.61	71.89	**71.97**	**71.97**
ResNet101	ResNet20	74.31	69.06	70.67	68.99	69.25	-	-	**71.99**

Table 1. Results for homogeneous architectures (CNN-to-CNN KD) on CIFAR-100. We report the top-1 accuracy (%).

Teacher	Student	Teacher	Student	Vanilla KD	RKD	MLD	DropKD
ViT-L/14	MobileViTV2	90.42	85.07	85.92	86.90	87.22	**87.42**

Table 2. Results for homogeneous architectures (Transformer-to-Transformer KD) on RAF-DB. We report the top-1 accuracy (%).

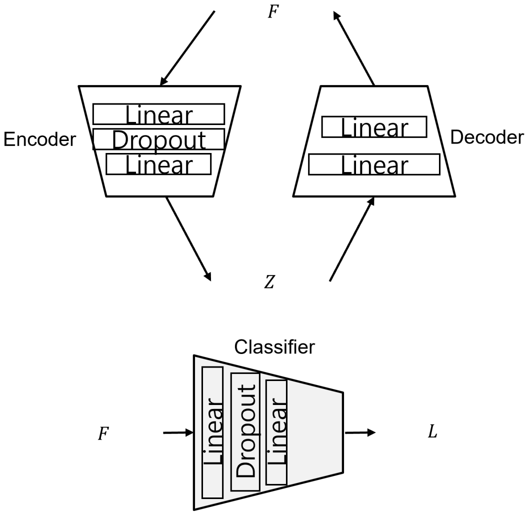


Figure 1. The architecture of encoder, decoder, and classifier.

feature maps from the teacher and student $F_T \in \mathbb{R}^{N \times d_t}$ and $F_S \in \mathbb{R}^{N \times d_s}$, where N indexes the batch size within a training dataset. To obtain CKA similarity, we compute normalized similarity in terms of the Hilbert-Schmidt Independence Criterion (HSIC) [1].

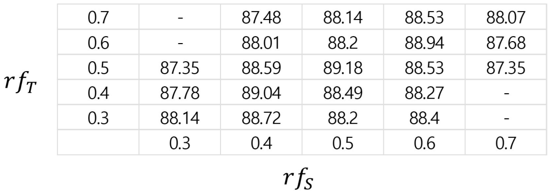
$$HSIC(\mathbf{K}, \mathbf{L}) = \frac{1}{n(n-3)}(tr(\tilde{\mathbf{K}}\tilde{\mathbf{L}}) + \frac{\mathbf{1}^\top \tilde{\mathbf{K}}\mathbf{1}\mathbf{1}^\top \tilde{\mathbf{L}}\mathbf{1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1}^\top \tilde{\mathbf{K}}\tilde{\mathbf{L}}\mathbf{1}), \tag{3}$$

where $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{L}}$ are acquired by setting the diagonal entries of similarity matrices $\mathbf{K}$ and $\mathbf{L}$ to zero. CKA can be calculated by averaging HSIC values over k samples:

$$CKA = \frac{\frac{1}{k}\sum_{i=1}^{k} HSIC(F_{Ti}F_{Ti}^\top, F_{Si}F_{Si}^\top)}{\sqrt{\frac{1}{k}\sum_{i=1}^{k} HSIC(F_{Ti}F_{Ti}^\top, F_{Ti}F_{Ti}^\top)}\sqrt{\frac{1}{k}\sum_{i=1}^{k} HSIC(F_{Si}F_{Si}^\top, F_{Si}F_{Si}^\top)}} \tag{4}$$

4. Training Details

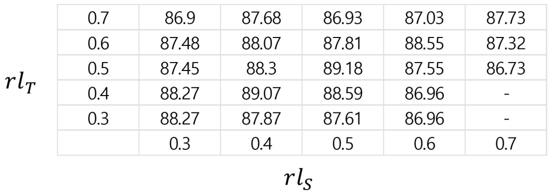
Our code is based on Pytorch with RTX TITANs. For each comparison knowledge distillation method, settings from the author's official code were used or hyperparameter settings that produced maximum performance were applied.



rf_T	0.7	-	87.48	88.14	88.53	88.07
	0.6	-	88.01	88.2	88.94	87.68
	0.5	87.35	88.59	89.18	88.53	87.35
	0.4	87.78	89.04	88.49	88.27	-
	0.3	88.14	88.72	88.2	88.4	-
		0.3	0.4	0.5	0.6	0.7

rf_S

Figure 2. The performance according to dropout rates rf_T and rf_S



rl_T	0.7	86.9	87.68	86.93	87.03	87.73
	0.6	87.48	88.07	87.81	88.55	87.32
	0.5	87.45	88.3	89.18	87.55	86.73
	0.4	88.27	89.07	88.59	86.96	-
	0.3	88.27	87.87	87.61	86.96	-
		0.3	0.4	0.5	0.6	0.7

rl_S

Figure 3. The performance according to dropout rates rl_T and rl_S

5. The Performance according to Dropout Rates

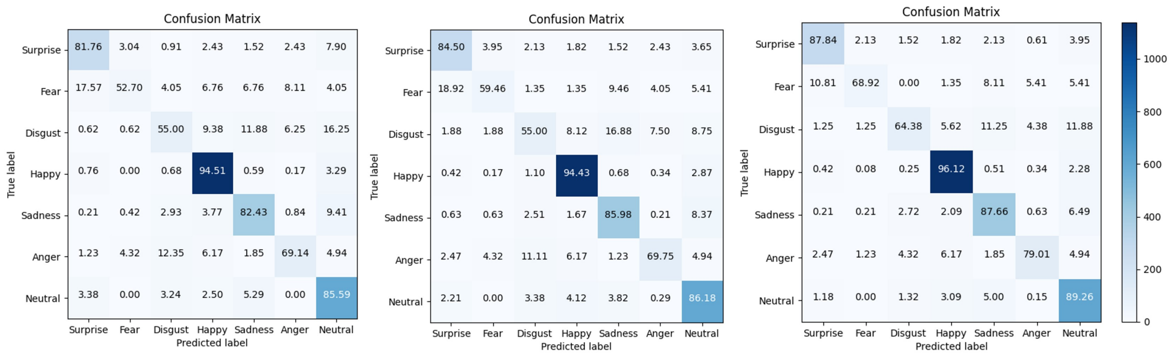
The experiment setting is the same as the Ablation Study in the main manuscript. Figure 2 shows evaluation results from DropKD according to different dropout rates of feature dropout distillation. Figure 2 shows evaluation results from DropKD according to different dropout rates of logit dropout distillation. DropKD showed the highest performance when the dropout rate of the student and teacher was the same. This can also be connected to domain knowledge of the projection space. It can be seen that the appropriate dropout rate at which feature maps become sparse has an important effect on generating general knowledge.

6. Evaluation on homogeneous architecture KD

Table 1 shows the results of homogeneous architecture scenarios (CNN-to-CNN KD). Our DropKD still achieves state-of-the-art results on CIFAR-100. Table 2 shows the results of homogeneous architecture scenarios (Transformer-to-Transformer KD). DropKD outperforms previous KD

Teacher	Student	Teacher	Student	Vanilla KD	RKD	MLD	DropKD
ResNet50	MobileViTV2	88.75	85.07	85.53	85.40	85.76	**86.27**

Table 3. Results for heterogeneous architectures (CNN-to-Transformer KD) on RAF-DB. We report the top-1 accuracy (%).



Teacher: ResNet50
Student: MobileNetV2
Distillation: MLD
UAR: 74.45%

Teacher: ViT-L/14
Student: MobileNetV2
Distillation: MLD
UAR: 76.47%

Teacher: ViT-L/14
Student: MobileNetV2
Distillation: DropKD (ours)
UAR: 81.88%

Figure 4. Confusion matrix and comparison of UAR

FER Model	FLOPs	Parameters	Acc (%)
ResNet50	3.89G	23.52M	88.75
ResNet18	1.8G	11.18M	86.53
SCN (ResNet18)	1.8G	11.18M	88.14
EAC (ResNet18)	1.8G	11.18M	89.20
MobileNetV2	320M	3.5M	84.49
ResNet18 + DropKD	1.8G	11.18M	**90.45**
MobileNetV2 + DropKD	320M	3.5M	**89.18**

Table 4. Comparison with state-of-the-art methods on RAF-DB.

methods on RAF-DB.

7. Evaluation on CNN-to-Transformer KD

Table 3 shows the results of homogeneous architecture scenarios (CNN-to-Transformer KD). Our DropKD still achieves state-of-the-art results on RAF-DB.

8. Comparison with state-of-the-art methods on RAF-DB

For comparison on RAF-DB, We employ state-of-the-art methods on RAF-DB such as SCN [2] and EAC [3]. Table 4 shows that ResNet18 + DropKD outperforms ResNet18 with EAC method and MobileNetV2 + DropKD achieves similar performance compared to ResNet18 with EAC.

KD Methods	Acc-both	Acc-single	Acc-single-top2
MobileNetV2	31.90	73.50	90.95
+KD	32.32	73.73	91.41
+DropKD	**36.49**	**76.01**	**91.54**

Table 5. Compound emotion dataset test

9. Confusion matrix and UAR

Figure 4 shows confusion matrices and unweight average recall (UAR). The larger UAR value the better the balance. Our proposed method outperforms the recent KD method by a large margin. So, these results mean that our proposed method transfers robust knowledge to the student.

10. Evaluation of generalization performance

To demonstrate the generalization performance of the proposed DropKD, we evaluate DropKD on a compound emotion dataset. We train MobileNetV2 with KD method on RAF-DB with single emotion and test the model on RAF-DB with compound emotion. Table 5 shows that our DropKD outperforms the previous KD method. Acc-both: The two highest predicted emotion classes match both labels. Acc-single: The highest predicted class is included in both labels. Acc-single-top2: At least one of the two highest predicted emotion classes is included in both labels. It means that our DropKD works better than previous methods in real-world situations.

References

[1] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(5), 2012. 2

[2] Kai Wang, Xiaojiang Peng, Jianfei Yang, Shijian Lu, and Yu Qiao. Suppressing uncertainties for large-scale facial expression recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6897–6906, 2020. 3

[3] Yuhang Zhang, Chengrui Wang, Xu Ling, and Weihong Deng. Learn from all: Erasing attention consistency for noisy label facial expression recognition. In *European Conference on Computer Vision*, pages 418–434. Springer, 2022. 3

[4] Martin Zong, Zengyu Qiu, Xinzhu Ma, Kunlin Yang, Chunya Liu, Jun Hou, Shuai Yi, and Wanli Ouyang. Better teacher better student: Dynamic prior knowledge for knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2022. 1