RELATIONAL REPRESENTATION DISTILLATION –SUPPLEMENTARY MATERIAL–

1. ALGORITHM

Algorithm 1 provides the pseudo-code of RRD.

2. IMPLEMENTATION DETAILS

We implement RRD in PyTorch following the implementation of CRD[1].

2.1. Baseline Methods

We compare our approach to the following state-of-the-art methods from the literature: (1) Knowledge Distillation (KD) [1]; (2) FitNets: Hints for Thin Deep Nets [2]; (3) Attention Transfer (AT) [3]; (4) Similarity-Preserving Knowledge Distillation (SP) [4]; (5) Correlation Congruence (CC) [5]; (6) Variational Information Distillation for Knowledge Transfer (VID) [6]; (7) Relational Knowledge Distillation (RKD) [7]; (8) Learning Deep Representations with Probabilistic Knowledge Transfer (PKT) [8]; (9) Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons (AB) [9]; (10) Paraphrasing Complex Network: Network Compression via Factor Transfer (FT) [10]; (11) A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning (FSP) [11]; (12) Like What You Like: Knowledge Distill via Neuron Selectivity Transfer (NST) [12]; (13) Contrastive Representation Distillation (CRD) [13]; (14) A Comprehensive Overhaul of Feature Distillation (OFD); (15) Rethinking Soft Labels for Knowledge Distillation: A Bias-Variance Tradeoff Perspective (WSLD) [14]; (16) Respecting Transfer Gap in Knowledge Distillation (IPWD) [15]; (17) Knowledge Distillation via Softmax Regression Representation Learning (SRRL) [16]; (18) Cross-Layer Distillation with Semantic Calibration (SemCKD) [17]; (19) Distilling Knowledge via Knowledge Review (ReviewKD) [18]; (20) Knowledge Distillation with the Reused Teacher Classifier (SimKD) [19]; (21) Searching A Fast Knowledge Distillation Process via Meta Optimization (DistPro) [20]; (22) Knowledge Distillation via N-to-One Representation Matching (NORM) [21]; (23) Information Theoretic Representation (ITRD) [22]; (24) Feature Kernel Distillation (FKD) [23]; (25) Complementary Relation Contrastive Distillation (CRCD) [24]; (26) Distilling Knowledge from Self-Supervised Teacher by Embedding Graph Alignment (EGA) [25]; (27) Wasserstein Contrastive Representation Distillation (WCoRD) [26].

[1]Available at: `https://github.com/HobbitLong/RepDistiller`.

2.2. Network Architectures

We use the following network architectures as described in [13]: (1) Wide Residual Network (WRN) [27], where WRN-d-w represents a wide ResNet with depth d and width factor w; (2) ResNet [28], where resnet-d represents a CIFAR-style ResNet with 3 groups of basic blocks having 16, 32, and 64 channels, respectively, and resnet-8 $\times 4$ and resnet-32 $\times 4$ indicate a 4-times wider network with 64, 128, and 256 channels; (3) ResNet [28], where ResNet-d represents an ImageNet-style ResNet with Bottleneck blocks and more channels; (4) MobileNet-v2 [29], using a width multiplier of 0.5 in our experiments; (5) VGG [30], where the VGG network used is adapted from its original ImageNet counterpart; and (6) ShuffleNet-v1 [31] and ShuffleNet-v2 [32], which are adapted for efficient training with input sizes of 32×32.

2.3. Optimization

All methods evaluated in our experiments use SGD with 0.9 Nesterov momentum. For CIFAR-100, we initialize the learning rate as 0.05, and decay it by 0.1 every 30 epochs after the first 150 epochs until the last 240 epoch. For MobileNet-v2, ShuffleNet-v1, and ShuffleNet-v2, we use a learning rate of 0.01 as this learning rate is optimal for these models in a grid search, while 0.05 is optimal for other models. The batch size is set to 64 for CIFAR-100, and the weight decay is set to 5×10^{-4}. For ImageNet, the initial learning rate is set to 0.1 and then divided by 10 at the 30th, 60th, and 90th epochs of the total 120 training epochs. The mini-batch size is set to 256, and the weight decay is set to 1×10^{-4}. All results are reported as means over three trials, except for the results on ImageNet, which are reported in a single trial.

3. RESULTS

3.1. Results on CIFAR-100

Table 1 and Table 2 provide a comprehensive overview of the top-1 accuracies of student networks trained with various

Algorithm 1 Pseudocode of RRD in a PyTorch-like style.

```
# f_t, f_s: outputs at the penultimate layer of teacher and student networks
# t_dim: The input feature dimension for the teacher
# s_dim: The input feature dimension for the student
# feat_dim: The projection feature space dimension
# nce_k: number of instances in queue
# nce_t_s, nce_t_t: the temperature paramters for student and teacher networks
# N: batch size

class RRDLoss(nn.Module):
   def __init__(self, s_dim, t_dim, feat_dim, nce_k=16384, nce_t_t=0.07, nce_t_s=0.04):
      super(RRDLoss, self).__init__()

      # embedding layer
      self.embed_s = nn.Linear(s_dim, feat_dim)
      self.embed_t = nn.Linear(t_dim, feat_dim)

      # memory buffer
      self.register_buffer("queue", torch.randn(nce_k, feat_dim))
      self.queue = F.normalize(self.queue, dim=0)
      self.register_buffer("queue_ptr", torch.zeros(1, dtype=torch.long))

   def forward(self, f_s, f_t):
      f_s = self.embed_s(f_s)
      f_t = self.embed_t(f_t)

      f_s = F.normalize(f_s, dim=1)
      f_t = F.normalize(f_t, dim=1)

      l_s = torch.einsum("nc,kc->nk", [f_s, self.queue])
      l_t = torch.einsum("nc,kc->nk", [f_t, self.queue])

      loss = -torch.sum(
         F.softmax(l_t / self.nce_t_t, dim=1) *
         F.log_softmax(l_s / self.nce_t_s, dim=1), dim=1).mean()

      self._dequeue_and_enqueue(f_t)

      return loss
```

state-of-the-art distillation techniques across a wide range of teacher-student architectural combinations. Unlike the main text, which summarizes a subset of results, these tables offer an extended comparison involving more models and training configurations. Our proposed method, RRD, shows strong performance across diverse network architectures and teacher-student pairs. RRD performs nearly as well as the top methods in knowledge distillation, achieving accuracy rates very close to the best-performing techniques, indicating an effective balance between simplicity and performance.

3.2. Results on ImageNet

Table 3 presents the top-1 accuracies of student networks trained with various distillation techniques across different teacher-student architectural pairings. These findings affirm the scalability of our RRD method on large datasets like ImageNet, highlighting its ability to effectively distill complex models. Our approach achieves competitive results, surpassing KD across all tested architectures. Furthermore, RRD shows improvement across different architectures, demonstrating its effectiveness in various distillation scenarios. The combination of RRD with KD further improves results among the compared techniques in most cases.

3.3. Capturing Inter-class Correlations

Cross-entropy loss overlooks the relationships among class logits in a teacher network, often resulting in less effective knowledge transfer. Distillation techniques that use ”soft targets”, such as those described by [1], have successfully captured these relationships, improving student model performance. Figure 1 assesses the effectiveness of different distillation methods on the CIFAR-100 KD task using WRN-40-2 as the teacher and WRN-40-1 as the student. We compare students trained without distillation, with attention transfer [3], with KL divergence [1], and with our proposed RRD method. Our findings show that RRD achieves close alignment between teacher and student logits, as evidenced by reduced differences in their correlation matrices. While RRD does not match CRD [13] in terms of exact correlation alignment, it significantly enhances learning efficiency and reduces error rates. The smaller discrepancies between teacher and student logits indicate that the RRD objective captures a substantial portion of the correlation structure in the logits, resulting in lower error rates, though CRD achieves a slightly closer match.

Table 1: Test top-1 accuracy (%) of student networks on CIFAR-100, comparing students and teachers of the same architecture using various distillation methods. The values in bold indicate the maximum of each column. ↑ denotes outperformance over KD and ↓ denotes underperformance.

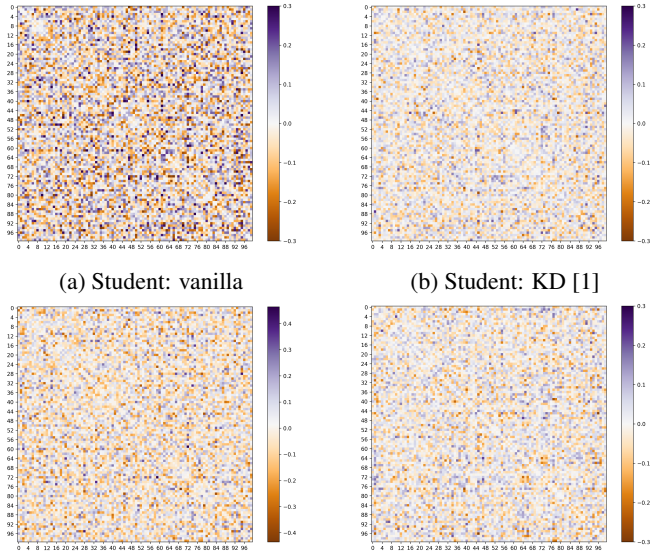
Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	resnet-56 resnet-20	resnet-110 resnet-20	resnet-110 resnet-32	resnet-32x4 resnet-8x4	VGG-13 VGG-8
Teacher	75.61	75.61	72.34	74.31	74.31	79.42	74.64
Student	73.26	71.98	69.06	69.06	71.14	72.50	70.36
KD [1]	74.92	73.54	70.66	70.67	73.08	73.33	72.98
FitNet [2]	73.58 (↓)	72.24 (↓)	69.21 (↓)	68.99 (↓)	71.06 (↓)	73.50 (↑)	71.02 (↓)
AT [3]	74.08 (↓)	72.77 (↓)	70.55 (↓)	70.22 (↓)	72.31 (↓)	73.44 (↑)	71.43 (↓)
SP [4]	73.83 (↓)	72.43 (↓)	69.67 (↓)	70.04 (↓)	72.69 (↓)	72.94 (↓)	72.68 (↓)
CC [5]	73.56 (↓)	72.21 (↓)	69.63 (↓)	69.48 (↓)	71.48 (↓)	72.97 (↓)	70.81 (↓)
VID [6]	74.11 (↓)	73.30 (↓)	70.38 (↓)	70.16 (↓)	72.61 (↓)	73.09 (↓)	71.23 (↓)
RKD [7]	73.35 (↓)	72.22 (↓)	69.61 (↓)	69.25 (↓)	71.82 (↓)	71.90 (↓)	71.48 (↓)
PKT [8]	74.54 (↓)	73.45 (↓)	70.34 (↓)	70.25 (↓)	72.61 (↓)	73.64 (↑)	72.88 (↓)
AB [9]	72.50 (↓)	72.38 (↓)	69.47 (↓)	69.53 (↓)	70.98 (↓)	73.17 (↓)	70.94 (↓)
FT [10]	73.25 (↓)	71.59 (↓)	69.84 (↓)	70.22 (↓)	72.37 (↓)	72.86 (↓)	70.58 (↓)
FSP [11]	72.91 (↓)	n/a	69.95 (↓)	70.11 (↓)	71.89 (↓)	72.62 (↓)	70.33 (↓)
NST [12]	73.68 (↓)	72.24 (↓)	69.60 (↓)	69.53 (↓)	71.96 (↓)	73.30 (↓)	71.53 (↓)
CRD [13]	75.48 (↑)	74.14 (↑)	71.16 (↑)	71.46 (↑)	73.48 (↑)	75.51 (↑)	73.94 (↑)
CRD+KD [13]	75.64 (↑)	74.38 (↑)	71.63 (↑)	71.56 (↑)	73.75 (↑)	75.46 (↑)	74.29 (↑)
OFD [33]	75.24 (↑)	74.33 (↑)	70.38 (↓)	n/a	73.23 (↑)	74.95 (↑)	73.95 (↑)
WSLD [14]	n/a	73.74 (↑)	71.53 (↑)	n/a	73.36 (↑)	74.79 (↑)	n/a
IPWD [15]	n/a	74.64 (↑)	71.32 (↑)	n/a	73.91 (↑)	76.03 (↑)	n/a
SRRL [16]	n/a	74.64 (↑)	n/a	n/a	n/a	75.39 (↑)	n/a
SemCKD [17]	n/a	74.41 (↑)	n/a	n/a	n/a	76.23 (↑)	n/a
ReviewKD [18]	76.12 (↑)	75.09 (↑)	71.89 (↑)	n/a	73.89 (↑)	75.63 (↑)	74.84 (↑)
SimKD [19]	n/a	75.56 (↑)	n/a	n/a	n/a	**78.08** (↑)	n/a
DistPro [20]	76.36 (↑)	n/a	72.03 (↑)	n/a	73.74 (↑)	n/a	n/a
NORM [21]	75.65 (↑)	74.82 (↑)	71.35 (↑)	71.55 (↑)	73.67 (↑)	76.49 (↑)	73.95 (↑)
NORM+KD [21]	76.26 (↑)	75.42 (↑)	71.61 (↑)	72.00 (↑)	74.95 (↑)	76.98 (↑)	74.46 (↑)
NORM+CRD [21]	76.02 (↑)	75.37 (↑)	71.51 (↑)	71.90 (↑)	73.81 (↑)	76.49 (↑)	73.58 (↑)
WCoRD [26]	75.88 (↑)	74.73 (↑)	71.56 (↑)	71.57 (↑)	73.81 (↑)	75.95 (↑)	74.55 (↑)
WCoRD+KD [26]	76.11 (↑)	74.72 (↑)	71.92 (↑)	71.88 (↑)	74.20 (↑)	76.15 (↑)	74.72 (↑)
CRCD [24]	**76.67** (↑)	**75.95** (↑)	**73.21** (↑)	**72.33** (↑)	**74.98** (↑)	76.42 (↑)	**74.97** (↑)
FKD [23]	n/a	n/a	n/a	n/a	n/a	75.57 (↑)	73.78 (↑)
ITRD (corr) [22]	75.85 (↑)	74.90 (↑)	71.45 (↑)	71.77 (↑)	74.02 (↑)	75.63 (↑)	74.70 (↑)
ITRD (corr+mi) [22]	76.12 (↑)	75.18 (↑)	71.47 (↑)	71.99 (↑)	74.26 (↑)	76.19 (↑)	74.93 (↑)
RRD (ours)	75.01 (↑)	73.55 (↑)	70.71 (↑)	70.72 (↑)	73.10 (↑)	74.48 (↑)	73.99 (↑)
RRD+KD (ours)	75.66 (↑)	74.39 (↑)	72.19 (↑)	71.74 (↑)	73.54 (↑)	75.08 (↑)	74.32 (↑)

Table 2: Test top-1 accuracy (%) of student networks on CIFAR-100 involving students and teachers from different architectures, using various distillation methods. The values in bold indicate the maximum of each column. ↑ denotes outperformance over KD and ↓ denotes underperformance.

Teacher Student	VGG-13 MobileNet-v2	ResNet-50 MobileNet-v2	ResNet-50 VGG-8	ResNet-32x4 ShuffleNet-v1	ResNet-32x4 ShuffleNet-v2	WRN-40-2 ShuffleNet-v1
Teacher	74.64	79.34	79.34	79.42	79.42	75.61
Student	64.60	64.60	70.36	70.5	71.82	70.5
KD [1]	67.37	67.35	73.81	74.07	74.45	74.83
FitNet [2]	64.14 (↓)	63.16 (↓)	70.69 (↓)	73.59 (↓)	73.54 (↓)	73.73 (↓)
AT [3]	59.40 (↓)	58.58 (↓)	71.84 (↓)	71.73 (↓)	72.73 (↓)	73.32 (↓)
SP [4]	66.30 (↓)	68.08 (↑)	73.34 (↓)	73.48 (↓)	74.56 (↑)	74.52 (↓)
CC [5]	64.86 (↓)	65.43 (↓)	70.25 (↓)	71.14 (↓)	71.29 (↓)	71.38 (↓)
VID [6]	65.56 (↓)	67.57 (↑)	70.30 (↓)	73.38 (↓)	73.40 (↓)	73.61 (↓)
RKD [7]	64.52 (↓)	64.43 (↓)	71.50 (↓)	72.28 (↓)	73.21 (↓)	72.21 (↓)
PKT [8]	67.13 (↓)	66.52 (↓)	73.01 (↓)	74.10 (↑)	74.69 (↑)	73.89 (↓)
AB [9]	66.06 (↓)	67.20 (↓)	70.65 (↓)	73.55 (↓)	74.31 (↓)	73.34 (↓)
FT [10]	61.78 (↓)	60.99 (↓)	70.29 (↓)	71.75 (↓)	72.50 (↓)	72.03 (↓)
NST [12]	58.16 (↓)	64.96 (↓)	71.28 (↓)	74.12 (↑)	74.68 (↑)	76.09 (↑)
CRD [13]	69.73 (↑)	69.11 (↑)	74.3 (↑)	75.11 (↑)	75.65 (↑)	76.05 (↑)
CRD+KD [13]	69.94 (↑)	69.54 (↑)	74.58 (↑)	75.12 (↑)	76.05 (↑)	76.27 (↑)
OFD [33]	69.48 (↑)	69.04 (↑)	n/a	75.98 (↑)	76.82 (↑)	75.85 (↑)
WSLD [14]	n/a	68.79 (↑)	73.80 (↓)	75.09 (↑)	n/a	75.23 (↑)
IPWD [15]	n/a	70.25 (↑)	74.97 (↑)	76.03 (↑)	n/a	76.44 (↑)
SRRL [16]	n/a	n/a	n/a	75.18 (↑)	n/a	n/a
SemCKD [17]	n/a	n/a	n/a	n/a	77.62 (↑)	n/a
ReviewKD [18]	70.37 (↑)	69.89 (↑)	n/a	77.45 (↑)	77.78 (↑)	77.14 (↑)
SimKD [19]	n/a	n/a	n/a	77.18 (↑)	n/a	n/a
DistPro [20]	n/a	n/a	n/a	77.18 (↑)	77.54 (↑)	77.24 (↑)
NORM [21]	68.94 (↑)	70.56 (↑)	75.17 (↑)	77.42 (↑)	78.07 (↑)	77.06 (↑)
NORM+KD [21]	69.38 (↑)	71.17 (↑)	75.67 (↑)	**77.79** (↑)	**78.32** (↑)	**77.63** (↑)
NORM+CRD [21]	69.17 (↑)	71.08 (↑)	75.51 (↑)	77.50 (↑)	77.96 (↑)	77.09 (↑)
WCoRD [26]	69.47 (↑)	70.45 (↑)	74.86 (↑)	75.40 (↑)	75.96 (↑)	76.32 (↑)
WCoRD+KD [26]	70.02 (↑)	70.12 (↑)	74.68 (↑)	75.77 (↑)	76.48 (↑)	76.68 (↑)
CRCD [24]	n/a	n/a	n/a	n/a	n/a	n/a
FKD [23]	n/a	n/a	74.61 (↑)	75 (↑)	n/a	n/a
ITRD (corr) [22]	69.97 (↑)	**71.41** (↑)	**75.71** (↑)	76.8 (↑)	77.27 (↑)	77.35 (↑)
ITRD (corr+mi) [22]	**70.39** (↑)	71.34 (↑)	75.49 (↑)	76.91 (↑)	77.40 (↑)	77.09 (↑)
RRD (ours)	67.93 (↑)	68.84 (↑)	74.01 (↑)	74.11 (↑)	74.80 (↑)	74.98 (↑)
RRD+KD (ours)	69.98 (↑)	69.13 (↑)	74.26 (↑)	75.18 (↑)	76.29 (↑)	76.31 (↑)

Table 3: Test top-1 (%) on ImageNet validation set using various distillation methods. The table compares students and teachers of the same and different architecture. The values in bold indicate the maximum of each column while underlined values mark the second best.

Teacher Student	ResNet-34 ResNet-18	ResNet-50 ResNet-18	ResNet-50 MobileNet
Teacher	73.31	76.16	76.16
Student	69.75	69.75	69.63
KD [1]	70.67	71.29	70.49
AT [3]	71.03	71.18	70.18
SP [4]	70.62	71.08	n/a
CC [5]	69.96	n/a	n/a
VID [6]	n/a	71.11	n/a
RKD [7]	70.40	n/a	68.50
AB [9]	n/a	n/a	68.89
FT [10]	n/a	n/a	69.88
FSP [11]	70.58	n/a	n/a
NST [12]	70.29	n/a	n/a
CRD [13]	71.17	71.25	69.07
OFD [33]	71.03	n/a	71.33
WSLD [14]	**72.04**	n/a	71.52
IPWD [15]	<u>71.88</u>	n/a	**72.65**
RRD (ours)	71.22	<u>71.33</u>	70.66
RRD+KD (ours)	71.40	**71.51**	<u>71.83</u>



(a) Student: vanilla (b) Student: KD [1]

(c) Student: AT [3] (d) Student: RRD (ours)

Fig. 1: Comparison of correlation matrix differences between teacher and student logits across various distillation methods on the CIFAR-100 task. Subfigures show results for (a) students trained without distillation, (b) with KL divergence [1], (c) with attention transfer (AT) [3], and (d) with our RRD method, highlighting better matching between student's and teacher's correlations. Results have been re-implemented according to [13].

4. REFERENCES

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," 2015.

[2] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio, "Fitnets: Hints for thin deep nets," in *Proceedings of the 4th International Conference on Learning Representations*, 2014.

[3] Sergey Zagoruyko and Nikos Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proceedings of the 5th International Conference on Learning Representations*, 2016.

[4] Frederick Tung and Greg Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1365–1374.

[5] Baoyun Peng, Xi Li, Yifan Wu, Yizhou Fan, Bo Wang, Qi Tian, and Jun Liang, "Correlation congruence for knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5007–5016.

[6] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9163–9171.

[7] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho, "Relational knowledge distillation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3967–3976.

[8] Nikolaos Passalis and Anastasios Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 268–284.

[9] Byeongho Heo, Minsik Lee, Seong Joon Yun, Jin Young Choi, and In So Kweon, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 3779–3787.

[10] Jangho Kim, Seongwon Park, and Nojun Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Advances in Neural Information Processing Systems*, 2018, pp. 2760–2769.

[11] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133–4141.

[12] Zehao Huang and Naiyan Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," in *Advances in Neural Information Processing Systems*, 2017, pp. 185–195.

[13] Yonglong Tian, Dilip Krishnan, and Phillip Isola, "Contrastive representation distillation," 2022.

[14] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang, "Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective," 2021.

[15] Yulei Niu, Long Chen, Chang Zhou, and Hanwang Zhang, "Respecting transfer gap in knowledge distillation," 2022.

[16] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos, "Knowledge distillation via softmax regression representation learning," in *International Conference on Learning Representations*, 2021.

[17] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Yan Feng, and Chun Chen, "Cross-layer distillation with semantic calibration," 2021.

[18] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia, "Distilling knowledge via knowledge review," 2021.

[19] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen, "Knowledge distillation with the reused teacher classifier," 2022.

[20] Xueqing Deng, Dawei Sun, Shawn Newsam, and Peng Wang, "Distpro: Searching a fast knowledge distillation process via meta optimization," 2022.

[21] Xiaolong Liu, Lujun Li, Chao Li, and Anbang Yao, "Norm: Knowledge distillation via n-to-one representation matching," 2023.

[22] Roy Miles, Adrian Lopez Rodriguez, and Krystian Mikolajczyk, "Information theoretic representation distillation," 2022.

[23] Bobby He and Mete Ozay, "Feature kernel distillation," in *International Conference on Learning Representations*, 2022.

[24] Jinguo Zhu, Shixiang Tang, Dapeng Chen, Shijie Yu, Yakun Liu, Aijun Yang, Mingzhe Rong, and Xiaohua Wang, "Complementary relation contrastive distillation," 2021.

[25] Yuchen Ma, Yanbei Chen, and Zeynep Akata, "Distilling knowledge from self-supervised teacher by embedding graph alignment," 2022.

[26] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin, "Wasserstein contrastive representation distillation," 2021.

[27] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," 2017.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015.

[29] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[30] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[31] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[32] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 116–131.

[33] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi, "A comprehensive overhaul of feature distillation," 2019.