Supplementary Material: Information Theoretic Representation Distillation

Roy Miles*
r.miles18@imperial.ac.uk
Adrian Lopez-Rodriguez*
al4415@imperial.ac.uk
Krystian Mikolajczyk
k.mikolajczyk@imperial.ac.uk

MatchLab
Imperial College London
Department of Electrical and Electronic Engineering
London, UK

1 Mutual Information Loss

Information potential The *log* transformation must be used in both the marginal and joint entropy terms to be a valid measure of mutual information. Without it, the loss resembles a distance between these quantities, rather than a ratio. However, empirically, we observed a small degradation in accuracy when using the log in both these terms, which can be seen in table 1.

$\mathcal{L}_{mi}$	$G_s - G_{st}$	$log_2(G_s) - log_2(G_{st})$
Mean	71.52	71.36
Std	0.25	0.35

Table 1: Accuracy (%) with and without the $\log_2$ data transformation. The experiments were performed for CIFAR-100 ResNet50→ MobileNetV2 distillation.

Exploring different kernels κ The kernel used for the $\mathcal{L}_{mi}$ loss was a polynomial kernel of degree 2, however, we also considered the use of a radial basis function (RBF) kernel $\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$. To select the values of σ, we then used Silverman's rule of thumb [19] $\sigma = h \times n^{-1/(4+d)}$, where n is the size of the mini-batch, d is the dimensionality of the representations, while h is an empirical value. The results can be seen in table 2 for both $h = 1.0$, $h = 5.0$, and the polynomial kernel. Although the RBF kernel did show promising results, the value of h is very dependant on both the dataset and the architectures used. To promote reproducibility of our results, we thus chose to use the polynomial kernel throughout.

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.
* The authors contributed equally to this paper

κ	Polynomial	RBF ($\sigma \approx 1.0$)	RBF ($\sigma \approx 5.0$)
Mean	71.52	71.14	70.99
Std	0.25	0.18	0.08

Table 2: Accuracy (%) with the RBF kernel for the $\mathcal{L}_{mi}$ with different kernel sizes σ. The experiments were performed for CIFAR-100 ResNet50$\rightarrow$ MobileNetV2 distillation.

2 Correlation Loss

Relationship to joint entropy. The objective from equation 6 closely resembles maximising $\log_2 \sum_i v_i$. However, although these two objectives share the same optimum solution, the flexibility in tuning the sharpness of the loss with α proved very effective. If we consider the self correlation matrices $\mathbf{C}_{ss}$ and $\mathbf{C}_{tt}$, the diagonal entries in $(\mathbf{C}_{ss} \circ \mathbf{C}_{tt})^2$ [1] will be populated with products of pairs of cross-correlation terms between $\mathbf{Z}_s$ and $\mathbf{Z}_t$. This matrix construction can then be used in equation 4 to compute the joint α-order entropy between the student and the teacher, where $\alpha = 2$. In the case where the features are strictly independent, *i.e.*, $(\mathbf{C}_{ss})_{ij} = (\mathbf{C}_{tt})_{ij} = 0 \quad \forall\, i \neq j$, the objective of the proposed loss in equation 6 and maximising this joint entropy are equivalent. In the more general setting, the joint entropy formulation maximises the correlation between all pairs of exemplars, while our proposed loss only maximises the correlation along the leading diagonal of $\mathbf{C}_{st}$.

Correlation v.s. Gram matrices. The connection to joint entropy is limited in that the matrices used are correlation matrices as opposed to Gram matrices in equation 4. This is an important distinction since in this loss we wish to capture the similarity across the feature-dimension as opposed to the batch-dimension. However, despite this distinction, there is still an intimate connection between these two matrices. As discussed in the recent work on cross-covariance attention [5], the non-zero part of the eigenspectrum of the Gram and covariance matrices are equivalent. Since the entropy-like formulation described in equation 3 is a spectral function of $\mathbf{A}$, the two resulting quantities are in turn closely related.

3 Additional experiments

Transferability of representations The main task of representation distillation is to train a smaller model to learn general and discriminative representations of the data. To confirm this result, we explore the task of transferring these models to two different datasets, namely Tiny ImageNet [24], and STL-10 [4]. Tiny ImageNet is a subset of ImageNet that contains 200 classes, with 500 training and 50 validation images per class each of size 64×64. On the other hand, STL-10 contains 10 classes, with 500 training and 800 testing images per class each of size 32×32. A WRN-16-2 student is first trained using ITRD from a WRN-40-2 teacher on the CIFAR100 dataset, after which the representation extractor is frozen and a new linear classifier is fine-tuned on the target data. The results are shown in table 3 and show that ITRD outperforms CRD+KD.

[1] This exponent denotes the square of a matrix, rather than an element-wise operation.

	Student	KD	AT	FitNet	CRD	CRD+KD	ITRD	Teacher
CIFAR100 → STL-10	69.7	70.9	70.7	70.3	71.6	72.2	**72.7**	68.6
CIFAR100 → TinyImageNet	33.7	33.9	34.2	33.5	35.6	35.5	**36.0**	31.5

Table 3: Transferability of the representations from CIFAR-100 to STL-10 and TinyImageNet. Only the linear classifier heads of each model are fine-tuned on the target datasets. The top-1 classification accuracies are reported (%).

Ablation study is performed for the impact of the weightings in the loss, namely β_{corr} and β_{mi}. The experiments were performed on CIFAR100 with a ResNet50 for the teacher and a MobileNetV2 for the student. The results are given in figure 1 and show that the student's performance is relatively robust to a wide range of values. For the β_{mi} weighting, the average loss maintains within 0.5% and a similar level of variation is achieved for $\beta_{corr} \in [1.5, 2.5]$. We further provide some insight into the choice of α for the correlation loss. Specifically, we evaluate the students performance when trained using a range of values for α, of which the results can be seen in table 4. The same dataset and student-teacher architecture are used from the previous ablation experiments. The best results are achieved with $\alpha = 2.0$, which demonstrates the benefit of incorporating Rényi's generalisation for entropy into the proposed losses.

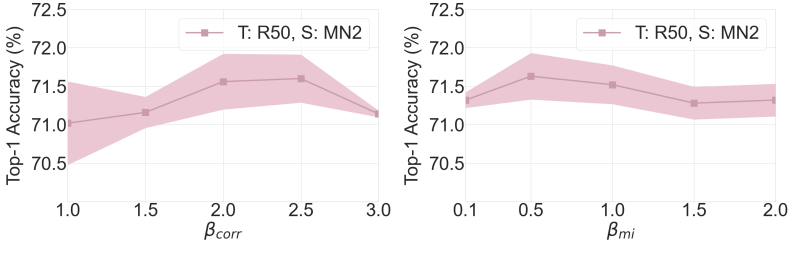

Figure 1: Accuracy (%) when varying both the correlation loss (left) and mutual information loss (right) weightings.

α	1.01	1.5	2.0	3.0	4.0	5.0	10.0
Mean	71.15	71.34	71.42	71.32	71.22	70.41	62.91
Std	0.21	0.33	0.39	0.16	0.06	0.43	1.21

Table 4: Accuracy (%) when varying α in the correlation loss for CIFAR-100 ResNet50→ MobileNetV2 distillation.

4 Discussion

Reproducibility To aid the reproducibility of this work, we implemented ITRD in both the CRD evaluation framework [21] and the *torchdistill* [14] KD reproducibility framework, which will both be released. Furthermore, the pseudo-code in algorithm **??** encapsulates both losses, showing the simplicity of using the proposed losses in current KD settings. We hope that the release of the code, along with the computational simplicity of our approach will encourage further development of this work.

5 Baseline Methods and Model Architectures

5.1 Baseline Methods

- Fitnets: Hints for thin deep nets [18]
- Knowledge Distillation (KD) [10]
- Attention Transfer (AT) [27]
- Like what you like: Knowledge distillation via neuron selectivity transfer (NST) [11]
- A gift from knowledge distillation: fast optimization, network minimization and transfer learning (FSP) [25]
- Learning deep representations with probabilistic knowledge transfer (PKT) [16]
- Paraphrasing complex network: network compression via factor transfer (FT) [12]
- Similarity-preserving knowledge distillation (SP) [22]
- Correlation congruence (CC) [17]
- Variational information distillation for knowledge transfer (VID) [1]
- Relational knowledge distillation (RKD) [15]
- Knowledge transfer via distillation of activation bound- aries formed by hidden neurons (AB) [9]
- Contrastive representation distillation (CRD) [21] via NCE [7].
- Wasserstein Contrastive Representation Distillation (WCoRD) [4]
- Distilling Knowledge via Knowledge Review (ReviewKD) [3]
- Complementary Relation Contrastive Distillation (CRCD) [29]

Note that the hyper-parameter setup for these baseline methods follows the setup in CRD [21].

5.2 Model Architectures

In experiments, we utilize the following model architectures.

- Wide Residual Network (WRN) [26]: WRN-d-w represents wide ResNet with depth d and width factor w.
- resnet [8]: We use ResNet-d to represent CIFAR-style resnet with 3 groups of basic blocks, each with 16, 32, and 64 channels, respectively. In our experiments, resnet8x4 and resnet32x4 indicate a 4 times wider net- work (namely, with 64, 128, and 256 channels for each of the blocks).
- ResNet [8]: ResNet-d represents ImageNet-style ResNet with bottleneck blocks and more channels.

- MobileNetV2 [6]: In our experiments, we use a width multiplier of 0.5.
- vgg [20]: The vgg networks used in our experiments are adapted from their original ImageNet counterpart.
- ShuffleNetV1 [28], ShuffleNetV2 [13]: ShuffleNets are proposed for efficient training and we adapt them to input of size 32x32.

5.3 Implementation Details

The CIFAR100 experimental evaluation and architectures used for comparisons are provided by Tian *et al*. in their work on contrastive representation distillation [21]. For the ImageNet experiments, we use the *torchdistill* [14] reproducibility framework, and for the binary distillation experiments we use the code provided by ReCU [23]. For completeness, we include the detail of the CRD provided architectures and training schedules here:

All methods evaluated in our experiments use SGD. 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 MobileNetV2, ShuffleNetV1 and ShuffleNetV2, 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.

6 Training costs

Figure 2 in the main paper provides the ImageNet training costs with the same hardware and a fixed memory constraint, where ITRD is relatively on par with the cheap standard KD. Table 5 shows additional results on the memory overhead for a fixed batch-size. In addition to these metrics, for the R34 $\rightarrow$ R18 ImageNet comparison, ITRD adds only **0.26M** trainable parameters by using a linear embedding layer, whereas e.g., ReviewKD introduces **1.8M** and CRD/WCoRD use a memory bank that stores tens of millions of parameters.

v.s.	ReviewKD	CRCD	HSAKD
Memory	**4.08**$\times$	over GPU (24GB) limit	**5.61**$\times$
Train time	**2.22**$\times$	**4.00**$\times$	**31.06**$\times$

Table 5: Relative overhead in terms of memory and training time against main competing distillation methods on ImageNet. Training time used a variable batch size to fit a pre-defined memory limit, while the memory experiments were using a fixed batch size.

Training convergence. Our method achieves a fast convergence, thus training with fewer epochs may lead to similar results to using the full training schedule. As shown in figure 2, ITRD achieves a higher final accuracy and also converges much faster after each learning rate drop than two of the main competitors (*i.e.*, CRD and KD).

7 Hyper-parameters fine-tuning

The proposed ITRD framework does not use any KL divergence for the logits. This would need additional tuning of α and τ, where τ is often very dataset dependent. Lots of other

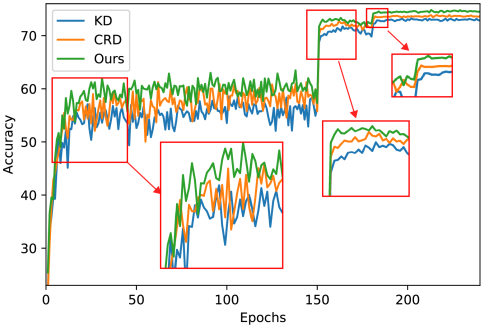

Figure 2: Training epochs *vs* validation accuracy for VGG13→VGG8 CIFAR 100 distillation. Zoomed-in regions show that our method converges faster to a higher accuracy.

methods report their results in conjunction with the KL loss [2, 21]. In addition, these methods also contain extra hyperparameters that need to be carefully tuned. For example, WCoRD[2] shows a robustness over a very small range $\mathbf{0} < \lambda_2 < \mathbf{0.2}$, while we provide a much larger range of values tested for our hyperparameters (β_{corr}, β_{mi}, α) whilst providing a comparable, or strictly lower, variation of performance. Similarly, CRD performance is also highly dependent on the number of negative samples used or the temperature chosen. Furthermore, we also use the same hyper-parameters used in the paper for the NLP experiments, thus demonstrating the robustness to the choice of hyperparameters.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. *CVPR*, 2019.

[2] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein Contrastive Representation Distillation. *CVPR*, 2020.

[3] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling Knowledge via Knowledge Review.

[4] Adam Coates, Honglak Lee, and Andrew Y. Ng. An analysis of single-layer networks in unsupervised feature learning. *JMLR*, 2011. ISSN 15324435.

[5] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jegou. XCiT: Cross-Covariance Image Transformers. *NeurIPS*, 2021.

[6] Michael H. Fox, Kyungmee Kim, and David Ehrenkrantz. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *CVPR*, 2018.

[7] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *JMLR*.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ResNet - Deep Residual Learning for Image Recognition. *CVPR*, 2015.

[9] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. *AAAI*, 2019.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *NeurIPS*, 2015.

[11] Zehao Huang and Naiyan Wang. Like What You Like: Knowledge Distill via Neuron Selectivity Transfer. 2017.

[12] Jangho Kim, Seong Uk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *NeurIPS*, 2018.

[13] Ningning Ma, Xiangyu Zhang, Hai Tao Zheng, and Jian Sun. Shufflenet V2: Practical guidelines for efficient cnn architecture design. In *Lecture Notes in Computer Science*, 2018.

[14] Yoshitomo Matsubara. torchdistill : A Modular, Configuration-Driven Framework for Knowledge Distillation. 2020.

[15] Wonpyo Park, Kakao Corp, Dongju Kim, and Yan Lu. Relational Knowledge Distillation. *CVPR*, 2019.

[16] Nikolaos Passalis and Anastasios Tefas. Learning Deep Representations with Probabilistic Knowledge Transfer. *ECCV*, 2018.

[17] Baoyun Peng, Xiao Jin, Dongsheng Li, Shunfeng Zhou, Yichao Wu, Jiaheng Liu, Zhaoning Zhang, and Yu Liu. Correlation congruence for knowledge distillation. *CVPR*, 2019.

[18] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints For Thin Deep Nets. *ICLR*, 2015.

[19] B.W. Silverman. Density estimation for statistics and data analysis. *Monographs on Statistics and Applied Probability*, 1986.

[20] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks For Large-scale Image Recognition. *ICLR*, 2015.

[21] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *ICLR*, 2019.

[22] Fred Tung and Greg Mori. Similarity-preserving knowledge distillation. *ICCV*, 2019.

[23] Zihan Xu, Mingbao Lin, Jianzhuang Liu, Jie Chen, Ling Shao, Yue Gao, Yonghong Tian, and Rongrong Ji. ReCU: Reviving the Dead Weights in Binary Neural Networks. *ICCV*, 2021.

[24] Le Ya and Yang Xuan. Tiny imagenet visual recognition challenge. 2015.

[25] Junho Yim. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. *CVPR*, 2017.

[26] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. *BMVC*, 2016.

[27] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2019.

[28] Xiangyu Zhang, Xinyu Zhou, and Mengxiao Lin. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *CVPR*, 2018.

[29] Jinguo Zhu, Shixiang Tang, Dapeng Chen, and Shijie Yu. Complementary Relation Contrastive Distillation.