Information Theoretic Representation Distillation

Roy Miles[1⋆], Adrian Lopez-Rodriguez[1⋆], and Krystian Mikolajczyk[1]

Imperial College London, UK
{r.miles18,al4415,k.mikolajczyk}@imperial.ac.uk

Abstract. Despite the empirical success of knowledge distillation, current state-of-the-art methods are computationally expensive to train, which makes them difficult to adopt in practice. To address this problem, we introduce two distinct complementary losses inspired by a cheap entropy-like estimator. These losses aim to maximise the correlation and mutual information between the student and teacher representations. Our method incurs significantly less training overheads than other approaches and achieves competitive performance to state-of-the-art on the knowledge distillation and cross-model transfer tasks. We further demonstrate the effectiveness of our method on a binary distillation task, whereby it leads to a new state-of-the-art for binary quantisation and approaches the performance of a full precision model. The code, evaluation protocols, and trained models will be publicly available.

1 Introduction

Deep learning has significantly advanced state-of-the-art across a wide range of computer vision tasks. Despite this success, most models are too computationally expensive to deploy on resource constrained devices. Fortunately, the training of such models is coupled with significant parameter redundancy, which has been explicitly exploited in the pruning and quantisation literature [3,19,29,60]. Knowledge distillation proposes an alternative approach whereby a much larger pre-trained model can provide additional supervision for a smaller model during training. This paradigm removes the restriction of the two models to share the same underlying architecture, thus enabling hand-crafted designs of the target architecture to meet the imposed resource constraints. However, some of the recent state-of-the-art distillation methods, *e.g.* including the recent union of self-supervision and knowledge distillation [43, 47], have made it increasingly expensive to train these student models. To this end, we develop a distillation method with a low computational overhead.

Information theory provides a natural lens for quantifying the statistical relationship between these models, and so is a common framework for deriving distillation losses [5,38]. Hence, we propose **I**nformation **T**heoretic **R**epresentation

⋆ The authors contributed equally to this paper

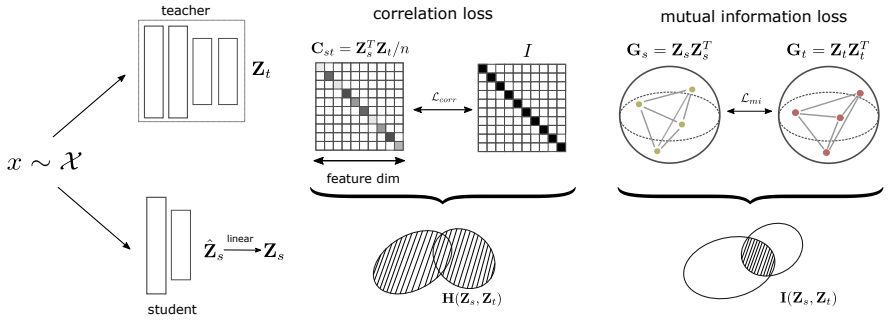

Fig. 1: Information theoretic representation distillation (ITRD) involves two distinct losses, namely a correlation loss and a mutual information loss. The former loss maximises the correlation between the student and teacher, while the latter maximises a quantity resembling the mutual information that aims to transfer the intra-batch sample similarity.

Distillation (ITRD) as a unified and computationally efficient framework that directly connects information theory with representation distillation. Specifically, this framework is inspired by the generalised Rényi's entropy and makes the training for specific applications more effective. Rényi's entropy is a generalisation of Shannon's entropy and has lead to improvements in other areas [24,35,51]. As figure 1 shows, we propose to model the distillation task with two distinct loss functions that correspond to maximising the correlation and mutual information between the student and teacher representations. The correlation loss aims to increase the similarity between teacher and student representations across the feature dimension. Conversely, the mutual information loss aims to match the intra-batch sample similarity between the teacher and the student. Our results show a strong accuracy v.s. training cost trade-off in comparison to state-of-the-art across two standard benchmarks, CIFAR100 and ImageNet, for a range of architecture pairings where we achieve up to 24.4% relative improvement. Our loss directly addresses the training efficiency problem, which we believe will encourage its adoption amongst machine learning researchers and practitioners. We further demonstrate the effectiveness of this framework on representation transfer and binary network transfer, whereby we are able to improve upon the state-of-the-art for both.

2 Related Work

Knowledge Distillation (KD) attempts to transfer the knowledge from a large pre-trained model (teacher) to a much smaller compressed model (student). This was originally introduced in the context of image classification [14], whereby the soft predictions of the teacher can act as pseudo ground truth labels for the student. The soft predictions then provide the student with supervision on the correlations between classes which are not explicitly available from one-hot encoded

ground truth labels. Spherical knowledge distillation [11] proposes to re-scale the logits before KD to address the capacity gap problem, while Prime-Aware Adaptive Distillation [58] introduces an adaptive sample weighting. Hinted losses provide a natural extension of KD using an L_2 distance between the student and teacher's intermediate representation [31]. Attention transfer [55] proposed to re-weight the spatial entries before the matching losses, while neuron selectivity transfer [15], similarity-preserving KD [40], and relational KD [25] attempt to transfer the structural similarity. Similarly, FSP matrices [48] attempt to capture the flow of information and Review KD [6] propose the use of attention based and hierarchical context modules. KD can also be modelled directly within a probabilistic framework [2, 26] through estimating and maximising the mutual information between the student and the teacher. ICKD [21] propose to transfer the correlation between channels of intermediate representations. A natural extension of supervised contrastive learning in the context of knowledge distillation was proposed in CRD [38]. WCoRD [5] also use a contrastive learning objective but through leveraging the dual and primal forms of the Wasserstein distance. CRCD [59] further develop this contrastive framework through the use of both feature and gradient information. Unfortunately, all of these contrastive methods require a large set of negative samples, which are sampled from a memory bank. The use of these memory banks incurs additional memory and computational costs, which we avoid altogether.

Additional self-supervision tasks have shown strong performance when coupled with representation distillation. Both SSKD [43] and HSAKD [47] introduce auxiliary tasks for classifying the rotation of images. However, these approaches incur a high training cost due to the added self-supervision task, which augments the training batches and adds additional classifiers. Weight sharing through jointly training sub-networks has also been shown to provide implicit knowledge distillation [23, 49, 50] and promising results. In this paper, we propose two distinct distillation losses applied to the features before the final fully-connected layer. Similarly to CRD [38], we posit that the logit representations lack relevant structural information that is necessary for effective distillation through the low dimensional embedding, while using the earlier intermediate representations can hinder the downstream task performance.

Information Theory (IT) provides a natural lens for interpreting and modelling the statistical relationships between intermediate representations of a neural network. This intersection of information theory and deep learning has subsequently led to a rigorous foundation in understanding the dynamics of training [1, 39], while also offering fruitful insights into other application domains, such as network pruning and knowledge distillation. In the context of representation distillation, most losses can be modelled as maximising some lower bound on the mutual information between the student and the teacher [5, 38]. In this work, we propose to forge an alternative connection between knowledge distillation and information theory using infinitely divisible kernels [4]. Specifically, we show that maximising both the correlation and mutual information yields two complimentary loss functions that can be related to these entropy-like quantities.

We achieve this using a matrix-based function that closely resembles Rényi's α-entropy [33, 34, 42], which is in turn a natural extension of the well-known Shannon's entropy used in IT. More recently, this work has been applied in the context of representation learning [53] for parameterising the information bottleneck principle.

3 Preliminaries

Representation Distillation describes the family of distillation methods which use the representation space that is given as the input to the final fully connected layer of a model. The generalised loss used for representation distillation can be concisely expressed in the following form:

$$\mathcal{L} = L_{XE}(\mathbf{y}, \text{softmax}(\mathbf{y}_s)) + \beta \cdot d(\mathbf{z}_s, \mathbf{z}_t) \tag{1}$$

where $\mathbf{z}_s \in \mathbb{R}^{d_s}$ and $\mathbf{z}_t \in \mathbb{R}^{d_t}$ are the student and teacher representations, β is a loss weighting, and d is the distillation loss function. The cross entropy between labels $\mathbf{y}$ and student logits $\mathbf{y}_s$, i.e., L_{XE} above, can be defined as the sum of an entropy and KL divergence term. Furthermore, standard KD [13] uses an additional KL divergence as the distillation loss between the student and teacher logits, with a temperature term that can soften or sharpen the two distributions.

Following [38], the motivation for using the feature representation space, as opposed to logits or any of the intermediate feature maps is two-fold. Firstly, this space preserves the structural information about the input, which may be lost through the low-dimensional embedding of the final layer. Secondly, intermediate feature matching losses may negatively impact the students' downstream performance in the cross-architecture tasks due to differing inductive biases [38], while also incurring significant computational and memory overheads due to the high-dimensionality of these feature maps.

In our work, to maximize the information transfer, we propose to express the distillation loss $d(.,.)$ as the weighted sum of a correlation and mutual information term. Below we link these two terms to a general formulation of entropy [34].

Information Theory Rényi's α-entropy [30] provides a natural extension of Shannon's entropy, which has been successfully applied in the context of differential privacy [24], understanding autoencoders [51], and face recognition [35]. For a random variable X with probability density function (PDF) $f(x)$ in a finite set $\mathcal{X}$, the α-entropy $\mathbf{H}_\alpha(X)$ is defined as:

$$\mathbf{H}_\alpha(f) = \frac{1}{1-\alpha} \log_2 \int_{\mathcal{X}} f^\alpha(x) dx \tag{2}$$

Where the limit as $\alpha \to 1$ is the well-known Shannon entropy. To avoid the need for evaluating the underlying probability distributions, a set of entropy-like quantities that closely resemble Renyi's entropy were proposed in [34, 42] and instead

estimate these information quantities directly from data. They are based on the theory of infinitely divisible matrices and leverage the representational power of reproducing kernel Hilbert spaces (RKHS), which have been widely studied and adopted in classical machine learning. Since its fruition, this framework has been applied in understanding convolutional neural networks (CNNs) [52], whereby they verify the important data processing inequality in information theory and further demonstrate a redundancy-synergy trade-off in layer representations. We propose to apply these estimators in the context of representation distillation, although they can also be applied in the context of network pruning.

In the following section, we provide definitions of the entropy based quantities and their connections with positive semidefinite matrices. This idea then naturally leads to a multi-variate extension using Hadamard products, from which conditional and mutual information can be defined. For brevity, we omit the proofs and connections with Rényi's axioms, which can be found in [34] and [42].

Definition 1: Let $X = \{x^{(1)}, \dots x^{(n)}\}$ be a set of n data points of dimension d and $\kappa : X \times X \to \mathbb{R}$ be a real valued positive definite kernel. The Gram matrix $\mathbf{K}$ is obtained from evaluating κ on all pairs of examples, that is $K_{ij} = \kappa(x^i, x^j)$. The matrix-based analogue to Rényi's α-entropy for a normalized positive definite (NPD) matrix $\mathbf{A}$ of size $n \times n$, such that $tr(\mathbf{A}) = 1$, can be given by the following functional:

$$\begin{aligned} \mathbf{S}_\alpha(\mathbf{A}) &= \frac{1}{1-\alpha} \log_2(tr(\mathbf{A}^\alpha)) \\ &= \frac{1}{1-\alpha} \log_2 \left[\sum_{i=1}^{n} \lambda_i(\mathbf{A}^\alpha) \right] \end{aligned} \tag{3}$$

where $\mathbf{A}$ is the kernel matrix $\mathbf{K}$ normalised to have a trace of 1 and $\lambda_i(\mathbf{A})$ denotes its i-th eigenvalue. This estimator can be seen as a statistic on the space computed by the kernel κ, while also satisfying useful properties attributed to entropy. In practice, the choice of both κ and α can be governed by domain specific knowledge, which we exploit for the task of knowledge distillation. The *log* in these definitions, which is conventionally taken as base 2, can be interpreted as a data-dependant transformation, and its argument is called the *information potential* [33]. In the context of optimisation, the information potential and entropy definitions can be used interchangeably since they are related by a strictly monotonic function.

We are interested in the statistical relationship between two sets of variables, namely the student and teacher representations. To measure this relationship, we introduce the notion of joint entropy, which naturally arises using the product kernel.

Definition 2: Let X and Y be two sets of data points. After computing the corresponding Gram matrices $\mathbf{A}$ and $\mathbf{B}$, the joint entropy is then given by:

$$\mathbf{S}_\alpha(\mathbf{A}, \mathbf{B}) = \mathbf{S}_\alpha \left(\frac{\mathbf{A} \circ \mathbf{B}}{tr(\mathbf{A} \circ \mathbf{B})} \right) \tag{4}$$

where $\circ$ denotes the Hadamard product between two matrices. Using these two definitions, the notion of conditional entropy and mutual information can be derived. We focus on the mutual information, which is given by:

$$\mathbf{I}_\alpha(\mathbf{A};\mathbf{B}) = \mathbf{S}_\alpha(\mathbf{A}) + \mathbf{S}_\alpha(\mathbf{B}) - \mathbf{S}_\alpha(\mathbf{A},\mathbf{B}) \tag{5}$$

Both equation 4 and 5 form a foundation for the correlation and mutual information losses respectively, which are proposed in the following section.

4 Information Theoretic Loss Functions

In this section we introduce two distillation losses that use two distinct and complimentary similarity measures between the student and teacher representations. The first loss uses a correlation measure which captures the similarity across the feature dimension, while the second loss is derived from a measure of mutual information and captures the similarity between examples within the mini-batch.

4.1 Maximising correlation

This first loss attempts to correlate the student and teacher representations. The intuition is that if the two sets of representations are perfectly correlated then the student is at least as discriminative as the teacher. Let $\mathbf{Z}_s \in \mathbb{R}^{n\times d}$ and $\mathbf{Z}_t \in \mathbb{R}^{n\times d}$ [1] denote a batch of representations from the student and teacher respectively. These matrices are computed before the final fully-connected layer to preserve the structural information of the data, thus enabling a strong distillation signal for the student. We first normalise these representations to zero mean and unit variance across the batch dimension and then propose to construct a cross-correlation matrix, $\mathbf{C}_{st} = \mathbf{Z}_s^T\mathbf{Z}_t/n \in \mathbb{R}^{d\times d}$. Perfect correlation between the two sets of representations is achieved if all of the diagonal entries $v_i = (\mathbf{C}_{st})_{ii}$ are equal to one. To formulate this as a minimization problem, we propose the following loss:

$$\mathcal{L}_{corr} = \log_2 \sum_{i=1}^{d} |v_i - 1|^{2\alpha} \tag{6}$$

This general objective is motivated by the recent work on Barlow Twins [56] for self-supervised learning, however, there are several distinct differences. Firstly, we drop the redundancy reduction term, which minimizes the off-diagonal entries in the cross correlation matrix, since we are not jointly learning both representations, *i.e.*, the teacher is fixed. In fact we observed that this objective significantly hurts the performance of the student. This performance degradation was similarly observed when decorrelating the off-diagonal entries in the self-correlation

[1] For clarity, we omit a linear embedding layer used on the student representations to match its dimensionality with the teacher.

matrix $\mathbf{C}_{ss}$, and is likely a consequence of the limited model capacity. Secondly, we introduce an α parameter, which provides a natural generalisation to emphasise low or highly correlated features. Finally, the $\log_2$ transformation was empirically shown to improve the performance by reducing spurious variations within a batch. These modifications were not only empirically justified, but also provide a closer relationship with the matrix-based entropy function in equation 3, which is discussed next.

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 $(\mathbf{C}_{ss} \circ \mathbf{C}_{tt})^2$ [2] 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 [9], 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.

4.2 Maximising mutual information

The correlation loss aims to match the information present in each feature dimension between the teacher and student representations. The mutual information loss provides an additional complimentary objective whereby we transfer the intra-batch similarity (*i.e.*, the relationship between samples) from the teacher representations to the student representations. The natural choice for achieving this through the lens of information theory is to maximise the mutual information between the two representations. Maximising the mutual information has been successfully applied in past distillation methods [2], following the idea that a high mutual information indicates a high dependence between the two models

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

and thus resulting in a strong student representation. Most others work relate their distillation losses to some lower bound on mutual information [38], however, using an alternative cheap entropy-like estimator, we propose to maximise this quantity directly.

$$\begin{aligned}\mathcal{L}_{mi} &= -\mathbf{I}_\alpha(\mathbf{G}_s;\mathbf{G}_t) \\ &= \mathbf{S}_\alpha(\mathbf{G}_s,\mathbf{G}_t) - \mathbf{S}_\alpha(\mathbf{G}_s) - \cancel{\mathbf{S}_\alpha(\mathbf{G}_t)}\end{aligned} \tag{7}$$

where $\mathbf{G}_s \in \mathbb{R}^{n\times n}$ and $\mathbf{G}_t \in \mathbb{R}^{n\times n}$ are the student and teacher Gram matrices (*i.e.*, $\mathbf{A}$ and $\mathbf{B}$ in equation 5). These matrices are constructed using a batch of normalised features $\mathbf{Z}_s$ and $\mathbf{Z}_t$ with a polynomial kernel of degree 1. The resulting matrix is subsequently normalised to have a trace of one. The teacher entropy term in this loss is omitted since the teacher weights are fixed during training. Substituting the marginal and joint entropy definitions from equations 3 and 4, with $\mathbf{G}_{st} = \mathbf{G}_s \circ \mathbf{G}_t$ (normalised to have a trace of one), leads to

$$\mathcal{L}_{mi} = \frac{1}{1-\alpha}\log_2 \sum_{i=1}^{n} \lambda_i \left(\mathbf{G}_{st}^{\alpha}\right) - \frac{1}{1-\alpha}\log_2 \sum_{i=1}^{n} \lambda_i \left(\mathbf{G}_{s}^{\alpha}\right) \tag{8}$$

Where $\mathbf{G}_{st}$ is also normalised to have unit trace. Since computing the eigenvalues for lots of large matrices can be computationally expensive during training [16], we restrict our attention to $\alpha = 2$. This allows us to use the Frobenius norm as a proxy objective and one of which has a connection with the eigenspectrum - $\|\mathbf{A}_F\|^2 \geq \sum_{i=1}^{n} \lambda_i^2(\mathbf{A})$.

$$\mathcal{L}_{mi} = \log_2 \|\mathbf{G}_s\|_F^2 - \log_2 \|\mathbf{G}_{st}\|_F^2 \tag{9}$$

In practice, we observed that removing the *log* transformations improved the performance, thus resulting in a slight departure from the connection to mutual information. Specifically, the loss instead minimises the distance between the marginal and joint *information potential*, rather than the mutual information.

4.3 Combining correlation and mutual information

Both the proposed losses provide two different learning objectives. Maximising the correlation is applied across the feature dimension, thus ensuring that the students average representation across the batch is perfectly correlated with the teacher. On the other hand, maximising the mutual information encourages the same similarity between samples as from the teacher. These two losses effectively operate distinctly over the two dimensions of the representations, namely the *feature*-dim and the *batch*-dim. The final loss for which we aim to minimise is given as follows:

$$\mathcal{L}_{ITRD} = \mathcal{L}_{CE} + \beta_{corr}\mathcal{L}_{corr} + \beta_{mi}\mathcal{L}_{mi} \tag{10}$$

where $\mathcal{L}_{CE}$ is a standard cross-entropy loss, while β_{corr} and β_{mi} are hyperparameters to weight the losses. To demonstrate the simplicity of our proposed method, and similarly to past works [56], we provide the PyTorch-based pseudocode in algorithm 1.

Algorithm 1 PyTorch-style pseudocode for ITRD

```
# f_s: Student network
# f_t: Teacher network
# y: Ground-truth labels
# y_s, y_t: Student and teacher logits
# z_s, z_t: Student and teacher representations (n x d)
for x in loader:
  # Forward pass
  z_s, y_s = f_s(x)
  z_t, y_t = f_t(x)
  z_s = embed(z_s)
  # Cross entropy loss
  loss = cross_entropy(y_s, y)

  # Normalise representations
  z_s_norm = (z_s - z_s.mean(0)) / z_s.std(0)
  z_t_norm = (z_t - z_t.mean(0)) / z_t.std(0)
  # Compute cross-correlation vector
  v = einsum('bx,bx→x', z_s, z_t) / n
  # Compute correlation loss
  dist = torch.pow(v - torch.ones_like(v), 2)
  h_st = torch.log2(torch.pow(dist, alpha).sum())
  loss += h_st.mul(beta_corr)

  # Compute Gram matrices
  z_s_norm = normalize(z_s, p=2)
  z_t_norm = normalize(z_t, p=2)
  g_s = einsum('bx,dx→bd', z_s_norm, z_s_norm)
  g_t = einsum('bx,dx→bd', z_t_norm, z_t_norm)
  g_st = g_s * g_t
  # Normalize Gram matrices
  g_s = g_s / torch.trace(g_s)
  g_st = g_st / torch.trace(g_st)
  # Compute the mutual information loss
  p = g_s.pow(2) - g_st.pow(2)
  loss += p.sum().mul(beta_mi)

  # Optimisation step
  loss.backward()
  optimizer.step()
```

5 Experiments

We evaluate our proposed distillation across two standard benchmarks, namely the CIFAR-100 and ImageNet datasets. To further demonstrate the effectiveness of our loss, we perform additional experiments on the transferability of the students representations and on distilling from a full-precision model to a binary network. For all of these experiments, we jointly train the student model with an additional linear embedding for the student representation. This embedding is used for the correlation loss and is shared by the mutual information loss when there is a mismatch in dimensions between the student and the teacher.

5.1 Model compression

Experiments on CIFAR-100 classification [18] consist of 60K 32×32 RGB images across 100 classes and with a 5:1 training/testing split. The results are shown in

Table 1: CIFAR-100 test *accuracy* (%) of student networks trained with a number of distillation methods. The best results are highlighted in **bold**, while the second best results are underlined. The mean and standard deviation was estimated over 3 runs. Same-architecture transfer experiments are highlighted in blue, whereas cross-architectural transfer is shown in red.

Teacher **Student**	**W40-2** **W16-2**	**W40-2** **W40-1**	**R56** **R20**	**R110** **R20**	**R110** **R32**	**R32x4** **R8x4**	**V13** **V8**	**V13** **MN2**	**R50** **MN2**	**R50** **V8**	**R32x4** **SN1**	**R32x4** **SN2**	**W40-2** **SN1**
Teacher	75.61	75.61	72.32	74.31	74.31	79.42	74.64	74.64	79.34	79.34	79.42	79.42	75.61
Student	73.26	71.98	69.06	69.06	71.14	72.50	70.36	64.60	64.60	70.36	70.50	71.82	70.50
KD [14]	74.92	73.54	70.66	70.67	73.08	73.33	72.98	67.37	67.35	73.81	74.07	74.45	74.83
FitNet [31]	73.58	72.24	69.21	68.99	71.06	73.50	71.02	64.14	63.16	70.69	73.59	73.54	73.73
AT [55]	74.08	72.77	70.55	70.22	72.31	73.44	71.43	59.40	58.58	71.84	71.73	72.73	73.32
SP [40]	73.83	72.43	69.67	70.04	72.69	72.94	72.68	66.30	68.08	73.34	73.48	74.56	74.52
CC [27]	73.56	72.21	69.63	69.48	71.48	72.97	70.71	64.86	65.43	70.25	71.14	71.29	71.38
RKD [25]	73.35	72.22	69.61	69.25	71.82	71.90	71.48	64.52	64.43	71.50	72.28	73.21	72.21
PKT [26]	74.54	73.45	70.34	70.25	72.61	73.64	72.88	67.13	66.52	73.01	74.10	74.69	73.89
FT [17]	73.25	71.59	69.84	70.22	72.37	72.86	70.58	61.78	60.99	70.29	71.75	72.50	72.03
NST [15]	73.68	72.24	69.60	69.53	71.96	73.30	71.53	58.16	64.96	71.28	74.12	74.68	74.89
CRD [38]	75.64	74.38	71.63	71.56	73.75	75.46	74.29	69.94	69.54	74.58	75.12	76.05	76.27
WCoRD [5]	76.11	74.72	**71.92**	71.88	74.20	76.15	74.72	70.02	70.12	74.68	75.77	76.48	76.68
ReviewKD [6]	**76.12**	75.09	71.89	-	73.89	75.63	74.84	70.37	69.89	-	**77.45**	**77.78**	77.14
$\mathcal{L}_{corr}$	75.85 ±0.12	74.90 ±0.29	71.45 ±0.21	71.77 ±0.34	74.02 ±0.27	75.63 ±0.09	74.70 ±0.27	69.97 ±0.33	**71.41** ±0.41	**75.71** ±0.02	76.80 ±0.28	77.27 ±0.25	**77.35** ±0.25
$\mathcal{L}_{corr} + \mathcal{L}_{mi}$	**76.12** ±0.04	**75.18** ±0.22	71.47 ±0.07	**71.99** ±0.46	**74.26** ±0.05	**76.19** ±0.22	**74.93** ±0.12	**70.39** ±0.39	71.34 ±0.33	75.49 ±0.32	76.91 ±0.19	77.40 ±0.06	77.09 ±0.08

table 1 for a range of student-teacher architecture pairs, where all of the reported methods use the same teacher weights. For a fair comparison, we only compare our results to methods that use the standard CRD [38] teacher weights.

The model abbreviations in the results table are given as follows: Wide residual networks (WRNd-w) [54], MobileNetV2 [10] (MN2), ShuffleNetV1 [57] / ShuffleNetV2 [37] (SN1 / SN2), and VGG13 / VGG8 [36] (V13 / V8). R32x4, R8x4, R110, R56 and R20 denote **CIFAR**-style residual networks, while R50 denotes an **ImageNet**-style ResNet50 [12].

CRCD [59] is not shown in this table since it uses different teacher weights, which are not released. Additionally, using the unofficial code that is released by the authors, we were unable to replicate their reported results. Although both SSKD and HSAKD do provide official implementations and corresponding teacher weights, their use of self-supervision and additional auxiliary tasks is much more computationally expensive and orthogonal to our work. However, we do include these methods in the experiment on ImageNet since the same teacher weights are used.

Table 2: Relative performance improvement (averaged over all architecture pairs in table 1) of the correlation and mutual information based losses against ReviewKD, WCoRD and $\mathcal{L}_{corr}$ only.

v.s.	ReviewKD	WCoRD	$\mathcal{L}_{corr}$
$\mathcal{L}_{corr}$	-3.7%	+16.2%	-
$\mathcal{L}_{corr} + \mathcal{L}_{mi}$	+6.8%	+24.4%	+10.5%

For all experiments in table 1, we set $\beta_{corr} = 2.0$ and $\beta_{mi} = 1.0$ (or $\beta_{mi} = 0.0$ when only using $\mathcal{L}_{corr}$). For the correlation loss α, we use a value of 1.01 for the same architectures and 1.50 for the cross-architectures. ITRD achieves the best performance for 10 out of 13 of the architecture pairs, with a 6.8% and 24.4% relative improvement[3] over ReviewKD and WCoRD respectively. The addition of $\mathcal{L}_{mi}$ is also shown to complement the $\mathcal{L}_{corr}$ loss through a 10.5% average relative improvement over all pairs, as shown in table 2.

Experiments on ImageNet classification [32] involve 1.3 million images from 1000 different classes. In this experiment, we set the input size to 224×224, and follow a standard augmentation pipeline of cropping, random aspect ratio and horizontal flipping. We use the *torchdistill* library with standard settings, *i.e.*, 100 epochs of training using SGD with an initial learning rate of 0.1 that is divided by 10 at epochs 30, 60 and 90. The results are shown against the total training efficiency in figure 2. The training efficiency is measured in *img/s*, which is inversely proportional to the total training time. For evaluating this metric, we used the official *torchdistill* implementations where possible. In the case of HSAKD, we used their official implementation and for CRCD we used the unofficial implementation provided by the authors. For a fair comparison, the batch sizes were scaled to ensure the training would fit within a pre-determined memory constraint of 8GB, and we used for training an RTX 2080Ti GPU.

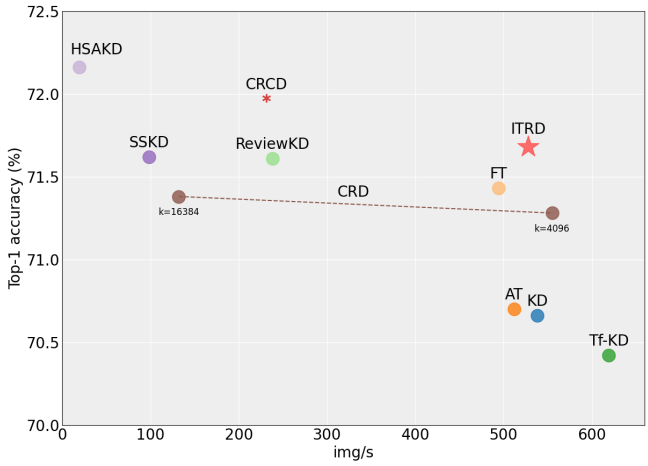

Fig. 2: Top-1 Accuracy on ImageNet v.s. training efficiency with a ResNet-18 as the student and a pre-trained ResNet-34 as the teacher. For CRCD, the training efficiency was evaluated using the authors unofficial implementation, while this accuracy is reported in their paper.

[3] For clarity, we use the same definition for relative improvement as provided in WCoRD [5]. This is given by $\frac{X-Y}{X-KD}$, where the X method is compared to Y relative to standard KD with KL divergence.

In terms of accuracy, ITRD achieves an error of 28.32%, being only behind CRCD and HSAKD, which are much more computationally intensive through the use of either negative contrastive sampling and a gradient-based loss, or additional augmented training data. Conversely, our method is computationally efficient. We only introduce a small complexity overhead that comes from a single linear layer that embeds the student and teacher representations in the same space, and also from computing the gram and cross-correlation matrices. The results show the applicability of our method to large-scale datasets, while also being significantly more efficient and simple to adopt.

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 3 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 3. 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.

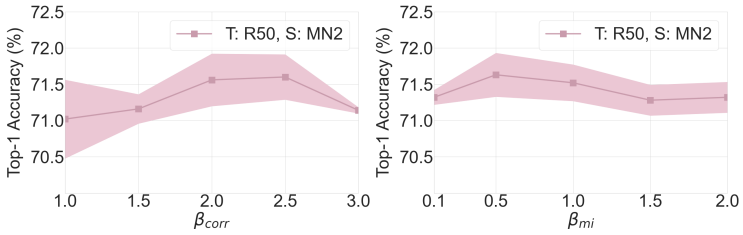

Fig. 3: Accuracy (%) when varying both the correlation loss (left) and mutual information loss (right) weightings.

Table 3: Accuracy (%) when varying α in the correlation loss for CIFAR-100 ResNet50→ MobileNetV2 distillation.

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

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 [46], and STL-10 [7]. 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 4 and show that ITRD outperforms CRD+KD.

Table 4: 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 (%).

	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

5.2 Binary distillation

Quantisation is often described as an orthogonal approach for network compression against other methods such knowledge distillation, pruning, and low-rank decomposition. Binary neural networks (BNNs) [8,20,28,45] are an extreme case of quantisation, where the weights can only represent two values. BNNs can obtain a steep increase of inference speed on CPUs [29] and FPGAs [41], while achieving significant model size reduction at the cost of only a small drop in accuracy compared to their full-precision counterparts.

In this section, we show that combining ITRD with binary quantisation can begin to bridge the gap between the binary and full-precision networks. For our experiments we use the state-of-the-art method ReCU [45] as our base model, and we distill the information from a full precision teacher to our binary student. In this experiment, both the full precision teacher and the binary student share the same architecture, the only difference being the quantisation modules in the student. Table 5 shows the full set of results, and for all distillation methods we employed the same hyperparameters used in the previous experiments. Both CRD and ReviewKD were shown to degrade the students performance. In contrast, ITRD improves upon the baseline accuracy by 1.3%, which is only 0.7% shy of the full-precision model. Since both the student and teacher networks adopt different non-linearities, the two networks interpolate and extrapolate between data very differently [44], subsequently making effective distillation very difficult. Although we expect a thorough search of hyperparameters for CRD and ReviewKD could improve their performance, the results demonstrate the robustness of ITRD to the training parameters as they were not modified for this experiment.

Table 5: Performance comparison with the state-of-the-arts on CIFAR-10. W/A denotes the bit length of the weights and activations. FP is short for full precision.

Network	Method	W/A	Top-1 (%)
ResNet-18	FP	32/32	94.8
	IR-Net [28]	1/1	91.5
	RBNN [20]	1/1	92.2
	ReCU [45]	1/1	92.8
	ReCU + CRD	1/1	92.1
	ReCU + ReviewKD	1/1	92.6
	ReCU + KD	1/1	93.3
	ReCU $+\mathcal{L}_{corr}$	1/1	93.9
	ReCU $+\mathcal{L}_{corr} + \mathcal{L}_{mi}$	1/1	**94.1**

5.3 Discussion

Reproducibility To aid the reproducibility of this work, we implemented ITRD in both the CRD evaluation framework [38] and the *torchdistill* [22] KD reproducibility framework, which will both be released. Furthermore, the pseudo-code in algorithm 1 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.

Limitations and future work. A large amount of the spatial information is lost by the final representation, which is the space in which the distillation losses are defined. This loss of spatial information may be regarded as a limitation for more structured tasks (*e.g.*, semantic segmentation or object detection), however, we expect that research in this direction would be a natural extension of this work. Another promising direction for this work is in the context of deep mutual learning. Jointly training both the teacher and student models can provide effective collaboration at the cost of increased training time.

6 Conclusion

In this work, we proposed an information-theoretic setting for representation distillation. Using this framework, we introduce novel distillation losses that are very simple and computationally inexpensive to adopt into most deep learning pipelines. Each of the proposed losses aim to extract complementary information from the teacher network. The correlation loss aims to guide the student to match the teacher representation on a feature-level. Conversely, the mutual information loss aims to transfer the intra-batch similarity between samples from the teacher to the student. We have shown the superiority of our approach compared to methods of similar computational costs on standard classification benchmarks. Furthermore, we have shown the applicability of our method to binary networks, whereby we begin to bridge the performance gap between full-precision and binary networks.

References

1. Advani, M., Kolchinsky, A., Tracey, B.D.: On the information bottleneck theory of deep learning (2019) 3
2. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. CVPR (2019) 3, 7
3. Bethge, J., Bartz, C., Yang, H., Chen, Y., Meinel, C.: MeliusNet : Can Binary Neural Networks Achieve MobileNet-level Accuracy ? 1
4. Bhati, R.: Infinitely Divisible Matrices. Transactions of the American Mathematical Society (1969) 3
5. Chen, L., Wang, D., Gan, Z., Liu, J., Henao, R., Carin, L.: Wasserstein Contrastive Representation Distillation. CVPR (2020) 1, 3, 10, 11
6. Chen, P., Liu, S., Zhao, H., Jia, J.: Distilling Knowledge via Knowledge Review 3, 10
7. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. JMLR (2011) 13
8. Ding, R., Chin, T.W., Liu, Z., Marculescu, D.: Regularizing activation distribution for training binarized deep networks. CVPR (2019) 13
9. El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., Jegou, H.: XCiT: Cross-Covariance Image Transformers. NeurIPS (2021) 7
10. Fox, M.H., Kim, K., Ehrenkrantz, D.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. CVPR (2018) 10
11. Guo, J., Chen, M., Hu, Y., Zhu, C., He, X., Cai, D.: Reducing the Teacher-Student Gap via Spherical Knowledge Distillation. arXiv preprint (2020) 3
12. He, K., Zhang, X., Ren, S., Sun, J.: ResNet - Deep Residual Learning for Image Recognition. CVPR (2015) 10
13. Hinton, G., Ilya, S., Martens, J., Dahl, G.: On the importance of initialization and momentum in deep learning. ICML (2013) 4
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network. NeurIPS (2015) 2, 10
15. Huang, Z., Wang, N.: Like What You Like: Knowledge Distill via Neuron Selectivity Transfer (2017) 3, 10
16. Kerr, A., Campbell, D., Richards, M.: QR decomposition on GPUs. Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units, GPGPU-2 (2009) 8
17. Kim, J., Park, S.U., Kwak, N.: Paraphrasing complex network: Network compression via factor transfer. NeurIPS (2018) 10
18. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images (2009) 9
19. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning Filters For Efficient Convnets. ICLR (2017) 1
20. Lin, M., Ji, R., Xu, Z., Zhang, B., Wang, Y., Wu, Y., Huang, F., Lin, C.W.: Rotated binary neural network. NeurIPS (2020) 13, 14
21. Liu, L., Huang, Q., Lin, S., Xie, H., Wang, B., Chang, X., Liang, X.: Exploring Inter-Channel Correlation for Diversity-preserved Knowledge Distillation. ICCV (2021) 3
22. Matsubara, Y.: torchdistill : A Modular, Configuration-Driven Framework for Knowledge Distillation (2020) 14
23. Miles, R., Mikolajczyk, K.: Cascaded channel pruning using hierarchical self-distillation. BMVC (2020) 3

24. Mironov, I.: Rényi Differential Privacy. Proceedings - IEEE Computer Security Foundations Symposium (2017) 2, 4
25. Park, W., Corp, K., Kim, D., Lu, Y.: Relational Knowledge Distillation. CVPR (2019) 3, 10
26. Passalis, N., Tefas, A.: Learning Deep Representations with Probabilistic Knowledge Transfer. ECCV (2018) 3, 10
27. Peng, B., Jin, X., Li, D., Zhou, S., Wu, Y., Liu, J., Zhang, Z., Liu, Y.: Correlation congruence for knowledge distillation. CVPR (2019) 10
28. Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., Song, J.: Forward and Backward Information Retention for Accurate Binary Neural Networks. CVPR (2020) 13, 14
29. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. ECCV (2016) 1, 13
30. Rényi, A.: On Measures of Entropy and Information. Proceedings of the fourth Berkeley Symposium on Mathematics, Statistics and Probability (1960) 4
31. Romero, A., Ballas, N., Ebrahimi Kahou, S., Chassang, A., Gatta, C., Bengio, Y.: FitNets: Hints For Thin Deep Nets. ICLR (2015) 3, 10
32. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (2014) 11
33. Sanchez Giraldo, L.G., Principe, J.C.: Information theoretic learning with infinitely divisible kernels. ICLR (2013) 4, 5
34. Sanchez Giraldo, L.G., Rao, M., Principe, J.C.: Measures of entropy from data using infinitely divisible Kernels. IEEE Transactions on Information Theory (2015) 4, 5
35. Shekar, B.H., Sharmila Kumari, M., Mestetskiy, L.M., Dyshkant, N.F.: Face recognition using kernel entropy component analysis. Neurocomputing (2011) 2, 4
36. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks For Large-scale Image Recognition. ICLR (2015) 10
37. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: MnasNet: Platform-Aware Neural Architecture Search for Mobile. CVPR (2018) 10
38. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. ICLR (2019) 1, 3, 4, 8, 10, 14
39. Tishby, N.: Deep Learning and the Information Bottleneck Principle 3
40. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. ICCV (2019) 3, 10
41. Umuroglu, Y., Fraser, N.J., Gambardella, G., Blott, M., Leong, P., Jahre, M., Vissers, K.: FINN: A framework for fast, scalable binarized neural network inference. In: FPGA 2017 (2017) 13
42. Williams, P.L., Beer, R.D.: Nonnegative Decomposition of Multivariate Information (2010) 4, 5
43. Xu, G., Liu, Z., Li, X., Loy, C.C.: Knowledge Distillation Meets Self-supervision. ECCV (2020) 1, 3
44. Xu, K., Zhang, M., Li, J., Du, S.S., Kawarabayashi, K.i., Jegelka, S.: How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. ICLR (2020) 13
45. Xu, Z., Lin, M., Liu, J., Chen, J., Shao, L., Gao, Y., Tian, Y., Ji, R.: ReCU: Reviving the Dead Weights in Binary Neural Networks. ICCV (2021) 13, 14
46. Ya, L., Xuan, Y.: Tiny imagenet visual recognition challenge (2015) 13

47. Yang, C., An, Z., Cai, L., Xu, Y.: Hierarchical Self-supervised Augmented Knowledge Distillation. IJCAI (2021) 1, 3
48. Yim, J.: A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. CVPR (2017) 3
49. Yu, J., Huang, T.: Universally Slimmable Networks and Improved Training Techniques. ICCV (2019) 3
50. Yu, J., Yang, L., Xu, N., Yang, J., Huang, T.: Slimmable Neural Networks. ICLR (2018) 3
51. Yu, S., Príncipe, J.C.: Understanding autoencoders with information theoretic concepts. Neural Networks (2019) 2, 4
52. Yu, S., Wickstrom, K., Jenssen, R., Principe, J.C.: Understanding Convolutional Neural Networks With Information Theory: An Initial Exploration. IEEE Transactions on Neural Networks and Learning Systems (2020) 5
53. Yu, X., Yu, S., Príncipe, J.C.: Deep deterministic information bottleneck with matrix-based entropy functional. ICASSP (2021) 4
54. Zagoruyko, S., Komodakis, N.: Wide Residual Networks. BMVC (2016) 10
55. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: ICLR (2019) 3, 10
56. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. ICML (2021) 6, 8
57. Zhang, X., Zhou, X., Lin, M.: ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. CVPR (2018) 10
58. Zhang, Y., Lan, Z., Dai, Y., Zeng, F., Bai, Y.: Prime-Aware Adaptive Distillation pp. 1–17 3
59. Zhu, J., Tang, S., Chen, D., Yu, S.: Complementary Relation Contrastive Distillation 3, 10
60. Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Guo, Y., Wu, Q., Huang, J., Zhu, J.: Discrimination-aware Channel Pruning for Deep Neural Networks. NeurIPS (2018) 1