A. Appendix

A.1. Limitations and Broader Impacts

In this paper, we presented a novel Mutual Information Maximization based knowledge distillation framework (MIMKD). Our method uses the JSD based lower-bound on mutual information which is optimized using only one negative sample. However, despite its favorable properties, our lower-bound may be less tight on the mutual information than the infoNCE bound as it approximates the mutual information by being monotonically related with it. Additionally, as we use only one negative sample, the performance of the method may be hindered by the presence of false negatives. The performance of the method is also effected by the architecture of the discriminator functions which can be explored further. We presented three information maximization formulations and demonstrated the value of region-consistent information maximization on distillation performance. We observe that the performance is slightly-sensitive to the hyper-parameters that control the relative value of our global, local, and feature information maximization formulations. This has been explored in great detail in our ablation sections and further demonstrated in figures 4, 5, and 6. Our method transfers representations from the teacher to the student. As such, harmful biases that the teacher has learnt are transferred to the student as well. And further exploration is required to alleviate the transfer of such biases during distillation.

A.2. Hyper-parameters for other methods

The student is trained with the following loss function which is a combination of the distillation loss and the cross-entropy loss for classification:

$$\mathcal{L} = \alpha\mathcal{L}_{cls} + (1-\alpha)\mathcal{L}_{KD} + \beta\mathcal{L}_{dis} \tag{7}$$

Note that we set $\alpha = 1$ for all methods except KD ß [12] and the value of β is set to the value recommended in the original work as follows:

1. KD [12]: $\alpha = 0.9, \beta = 0$
2. Fitnet [19]: $\beta = 100$
3. AT [28]: $\beta = 1000$
4. VID [1]: $\beta = 1$
5. CRD [24]: $\beta = 0.8$, for CRD evaluation, we use a original work inspired self-implementation with 4096 negative samples and $i \neq j$ negative sampling methodology as described in the original work.

A.3. Pairing Intermediate Representations

A.3.1 Similar CNN Architectures.

Consider the case of distillation when the teacher network is a pre-trained WRN-40-2 and the student network is a WRN-16-1. We use 4 same-sized representations extracted from intermediate layers of the networks. Therefore, the set $\mathcal{R} = \{(f_t^{(k)}(x), f_s^{(k)}(x))\}_{k=1}^{K}$ contains k pairs of same-sized 2-dimensional representations. Table 4 describes the sizes of the intermediate representations used for feature-based mutual information maximization. It can be seen that for this combination we use $k = 4$ in our formulation.

Table 4. Dimensions of intermediate representation in the form $channels \times height \times width$ used for feature-level mutual information maximization between a teacher WRN-40-2 and a student WRN-16-1 network. Alternatively, each value of k represents a pair of elements in the set $\mathcal{R}$.

	WRN-40-2	WRN-16-1
k	$f_t^{(k)}(x)$	$f_s^{(k)}(x)$
1	$16 \times 32 \times 32$	$16 \times 32 \times 32$
2	$32 \times 32 \times 32$	$16 \times 32 \times 32$
3	$64 \times 16 \times 16$	$32 \times 16 \times 16$
4	$128 \times 8 \times 8$	$64 \times 8 \times 8$

A.3.2 Dissimilar CNN Architectures.

Similar approach of defining the set $\mathcal{R}$ is followed in cases where the teacher and student networks have significantly different architectures. For instance, Table 5 shows the dimensions of intermediate representations used when the teacher network is a ResNet34 while the student is a ShuffleNetV2. Here $k = 4$ is used, however, for some combinations of different standard architectures we use $k = 3$ if only 3 pairs intermediate representations from the teacher and the student have the same size. Note that our method is invariant to the number of channels in the representations. Therefore, mismatch in the number of channels in pairs of representations in $\mathcal{R}$ is inconsequential for the formulation of our losses.

A.4. Mutual Information Discriminators

The parameterized mutual information discriminator functions (T_{ω_g}, T_{ω_l}, and T_{ω_f}) can be modeled as neural networks. In our experiments, we use two distinct discriminator architectures inspired from the functions presented in Deep InfoMax [13].

Table 5. Dimensions of intermediate representation in the form $channels \times height \times width$ used for feature-level mutual information maximization between a teacher WRN-40-2 and a student WRN-16-1 network. Alternatively, each value of k represents a pair of elements in the set $\mathcal{R}$.

	ResNet34	ShuffleNetV2
k	$f_t^{(k)}(x)$	$f_s^{(k)}(x)$
1	$64 \times 32 \times 32$	$24 \times 32 \times 32$
2	$512 \times 16 \times 16$	$116 \times 16 \times 16$
3	$1024 \times 8 \times 8$	$232 \times 8 \times 8$
4	$2048 \times 4 \times 4$	$464 \times 4 \times 4$

A.4.1 Convolve Architecture.

In this method, the representations from the teacher and the student are concatenated together and passed through a series of layers to get the score. For global information maximization, the final representations from both networks is concatenated together to get $[f_s(x), f_t(x)]$. This vector is then passed to a fully connected network with two 512-unit hidden layers, each followed by a $ReLU$ non-linearity (ref. table 6). The output is then passed through another linear layer to obtain the final score.

Table 6. The architecture of the discriminator used for global information maximization. Here LL denotes Linear Layer and $d(v)$ refers to the number of dimensions in vector v.

Input	Operation	Output
$[f_t(x), f_s(x)]$	LL + ReLU	O_1
O_1	LL + ReLU	O_2
O_2	LL	score

For local information maximization, we replicate the final representation from the teacher $f_t(x)$ to match the $m_K \times m_K$ size of the student's last intermediate feature map ($f_s^{(K)}(x)$). The resulting replicated tensor is then concatenated with $f_s^{(K)}(x)$ to get $[f_t(x), f_s^{(K)}(x)]$ which serves as the input for the critic function (ref. table 7).

Table 7. The architecture of the discriminator used for local and feature mutual information maximization. Note that for feature mutual information maximization the input at the first layer is $[f_t^{(k)}(x), f_s^{(k)}(x)]$.

Input	Operation	Output
$[f_t(x), f_s^{(K)}(x)]$	1×1 Conv + ReLU	O_1
O_1	1×1 Conv + ReLU	O_2
O_2	1×1 Conv	scores

Similarly, consider feature mutual information maximization, for each pair in the set $\mathcal{R}$ we use a distinct discriminator $T_{\omega_f}^{(k)}$. For a given k, each pair of intermediate feature representations in the set $\mathcal{R}$ are concatenated together to get $[f_t^{(k)}(x), f_s^{(k)}(x)]$. Which is then passed through two convolutional (1×1 kernels and 512 filters) where each layer is followed by a ReLU non-linearity. The output obtained is then further passed into a convolutional layer (1×1 kernels and 1 filter) to give $m_k \times m_k$ scores (ref. table 7).

A.4.2 Project and Dot Architecture.

In this method, the representations from both the teacher and the student are first projected using an appropriate projection architecture with a linear shortcut. The dot-product of these projections is then computed to get the score. Positive and negative pairs of representations are passed through the discriminator to get respective scores to be passed into equation (2) to get the estimates on the lower bound of the mutual information. One-dimensional representations are projected using the architecture described in table 8, whereas for two-dimensional intermediate feature maps, projection architecture described in table 9 is used.

Table 8. The projection architecture used for one-dimensional inputs. Here, LL denotes linear layer while LN denotes layer normalization. Both $f_t(x)$ and $f_s(x)$ are projected using this architecture and their dot product is computed to get scores.

Input	Operation	Output
$f_t(x)$ or $f_s(x)$	LL + ReLU + LL	O_1
$f_t(x)$ or $f_s(x)$	LL + ReLU	O_2
$O_1 + O_2$	LN	$proj$

Therefore, for (1) global information maximization, both $f_t(x)$ and $f_s(x)$ are projected using the one-dimensional projection architecture, for (2) local information maximization, the final teacher representation, $f_t(x)$, is projected using the one-dimensional projection architecture and duplicated to match the size of the projected intermediate student representation projected using the two-dimensional projection architecture, a dot product of these outputs is then computed to get the scores, while for (3) feature information maximization, both representations in each pair of the set $\mathcal{R}$ is projected using a respective two-dimensional projection architecture.

A.5. ImageNet results

In this experiment we train a student ResNet-18 with a pre-trained teacher ResNet-34 on the ImageNet dataset (ILSVRC). Note that we do not perform any hyper-parameter tuning specifically for this configuration and use the same values we obtained for the CIFAR-100 dataset i.e.

Table 9. The projection architecture used for two-dimensional inputs. Here, LL denotes linear layer while LN denotes layer normalization.

Input	Operation	Output
$f_s^{(k)}(x)$	1×1 Conv + ReLU + LL	O_1
$f_s^{(k)}(x)$	1×1 Conv + ReLU	O_2
$O_1 + O_2$	LN	$proj$

Table 10. Observed top-1 validation accuracy (in %) of the student network on the ImageNet dataset using our method (MIMKD) and other distillation frameworks. In similar settings, the more recent Contrastive Representation Distillation (CRD) method reports comparable performance with an improvement of $+1.42$ from a student network [24].

Student Network **Teacher Network**	**ResNet-18** **ResNet-34**
Student Accuracy	68.88
Teacher Accuracy	$72.82_{+3.94}$
Knowledge Distill. (KD)	$69.66_{+0.78}$
Attention Transfer (AT)	$69.70_{+0.82}$
MIMKD (this work)	**70.32**$_{+1.44}$

$\alpha = 0.9$, $\lambda_g = 0.2$, $\lambda_l = 0.8$, $\lambda_f = 0.8$. We observed that our method is able to reduce the gap between the teacher and the student performance by 1.44%. Results are presented in Table 10.

A.6. Shallow CNN Architectures

In this section, we describe our experiments where we distill knowledge from a standard teacher network into a shallow custom-designed CNN. This is done to demonstrate that it is feasible to design and distill information into lightweight models such that they perform competitively with standard CNN architectures while running faster. For our experiments we use 2 shallow CNNs; (1) Conv-4 with 4 convolutional-blocks followed by average pooling operation and a linear layer, where each convolutional-block is made-up of a convolutional layer with kernel size 3×3 and stride 2 followed by batch-normalization and a ReLU nonlinearity, (2) Conv-4-MP which has 4 convolutions blocks followed by average pooling and a linear layer at the end, where each convolutional-block contains a convolutional layer with kernel size 3×3 and stride 1 followed by batch-normalization, ReLU and a max-pooling layer. These architectures were chosen as they are compact and run relatively faster on standard CPUs. Table 11 compiles our results compared to other distillation methods for custom-designed shallow CNN architectures. Notice how a simple model such as Conv-4-MP becomes competitive with ShuffleNetV2's base student accuracy. Our method is able to outperform all other methods in this setup. Additionally, we can see that distillation is most successful with ResNet-32x4 as the teacher than for other architectures. This could be because of the larger gap in the baseline accuracy of the networks. Under this more controlled experiment with fixed students, larger gaps between student-teacher pairs also led to larger gains after distillation.

A.7. Computational cost and negative sampling.

We contextualize the memory and computational overhead of our work with respect to CRD. Our global MI objective has the same footprint as CRD (i.e. an additional 600MB over standard Resnet18 training for storing negatives). In addition, our feature and local MI objective use projection layers which add an additional 100MB of GPU memory. As the computation of our JSD-based objective is computationally trivial, we observe negligible reduction in training speed wrt CRD (2.2 epochs/hr v. 2.4 epochs/hr). Note that no additional memory is used for sampling negatives for local and feature information maximization. The 4096 negatives are only used for global MI as storing 1-D representations is relatively inexpensive.

A.8. Ablation Study

In this section we present additional accuracy landscape plots for our extensive ablation study that demonstrates the value of each component of our mutual information maximization objective. We use a ResNet-32x4 as the teacher network and ResNet-8x4 as the student network where the baseline accuracy of the teacher is 79.24% and that of the student network is 72.44%. The values of the hyper-parameters λ_g, λ_l and λ_f — that control the weight of the global, local and feature mutual information maximization objectives respectively – were varied between 0 and 1 with an increment of 0.25 while the weight for the cross-entropy loss, α was set to 1. The following contour plots shows the test accuracy landscape with respect to a pair of hyper-parameters when the third hyper-parameter is set to distinct values. Overall, this demonstrates the value of maximizing region-consistent local and feature-level mutual information between representations in addition to just global information maximization.

Table 11. Observed test accuracy (in %) of shallow student networks trained with teacher networks of higher capacity and standard architectures on the CIFAR100 dataset using our methods MIMKD and other distillation frameworks.

Student Net.	**Conv-4**			**Conv-4-MP**		
Teacher Net.	**ResNet-110**	**VGG-13**	**ResNet-32x4**	**ResNet-110**	**VGG-13**	**ResNet-32x4**
Student Acc.	59.97	59.97	59.97	66.09	66.09	66.09
Teacher Acc.	73.82$_{+13.85}$	74.62$_{+14.65}$	79.24$_{+19.27}$	73.82$_{+7.73}$	74.62$_{+8.53}$	79.24$_{+13.15}$
FitNets	60.58$_{+0.61}$	61.81$_{+1.84}$	62.89$_{+2.92}$	67.38$_{+1.29}$	66.52$_{+0.43}$	67.21$_{+1.12}$
AT	61.65$_{+1.68}$	62.16$_{+2.19}$	63.10$_{+3.13}$	67.52$_{+1.43}$	66.21$_{+0.12}$	66.03$_{-0.06}$
VID	61.93$_{+1.96}$	62.49$_{+2.52}$	63.45$_{+3.48}$	67.76$_{+1.67}$	67.40$_{+1.31}$	67.86$_{+1.77}$
KD	61.98$_{+2.01}$	62.10$_{+2.13}$	62.87$_{+2.90}$	67.51$_{+1.42}$	67.84$_{+1.75}$	68.04$_{+1.95}$
CRD	62.13$_{+2.16}$	62.54$_{+2.57}$	63.76$_{+3.79}$	67.96$_{+1.87}$	68.06$_{+1.97}$	68.52$_{+2.43}$
MIMKD (ours)	**62.91**$_{+2.94}$	**62.95**$_{+2.98}$	**64.32**$_{+4.35}$	**68.77**$_{+2.68}$	**68.91**$_{+2.82}$	**69.09**$_{+3.00}$

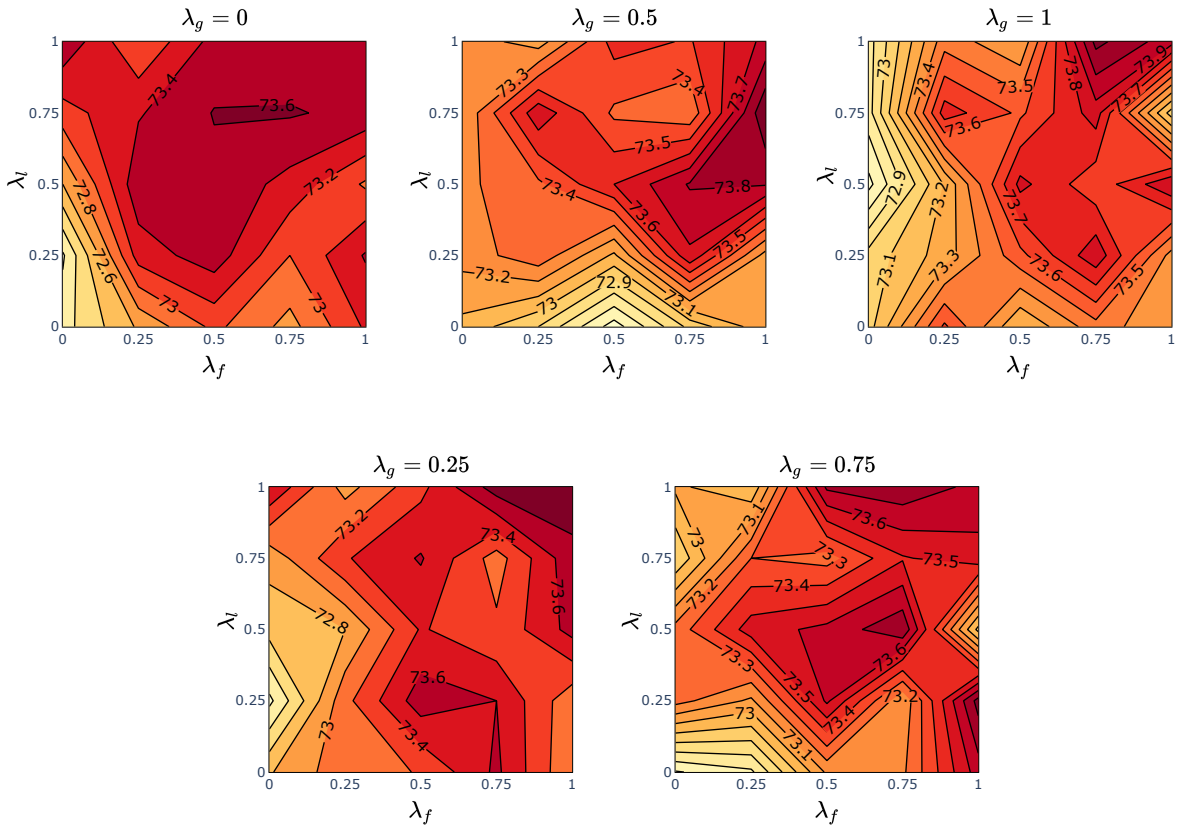

Figure 4. Results from the ablation studies on CIFAR100 dataset using a student resnet8x4 (baseline acc. 72.44%) with teacher resnet32x4 (baseline acc. 79.24%). Contour lines represent the final test accuracy of the student. Grid search was performed by varying the values of $\lambda_f, \lambda_g, \lambda_l$ from 0 to 1 with increments of 0.25. In each plot, the accuracy landscape is shown with λ_g set to a constant value.

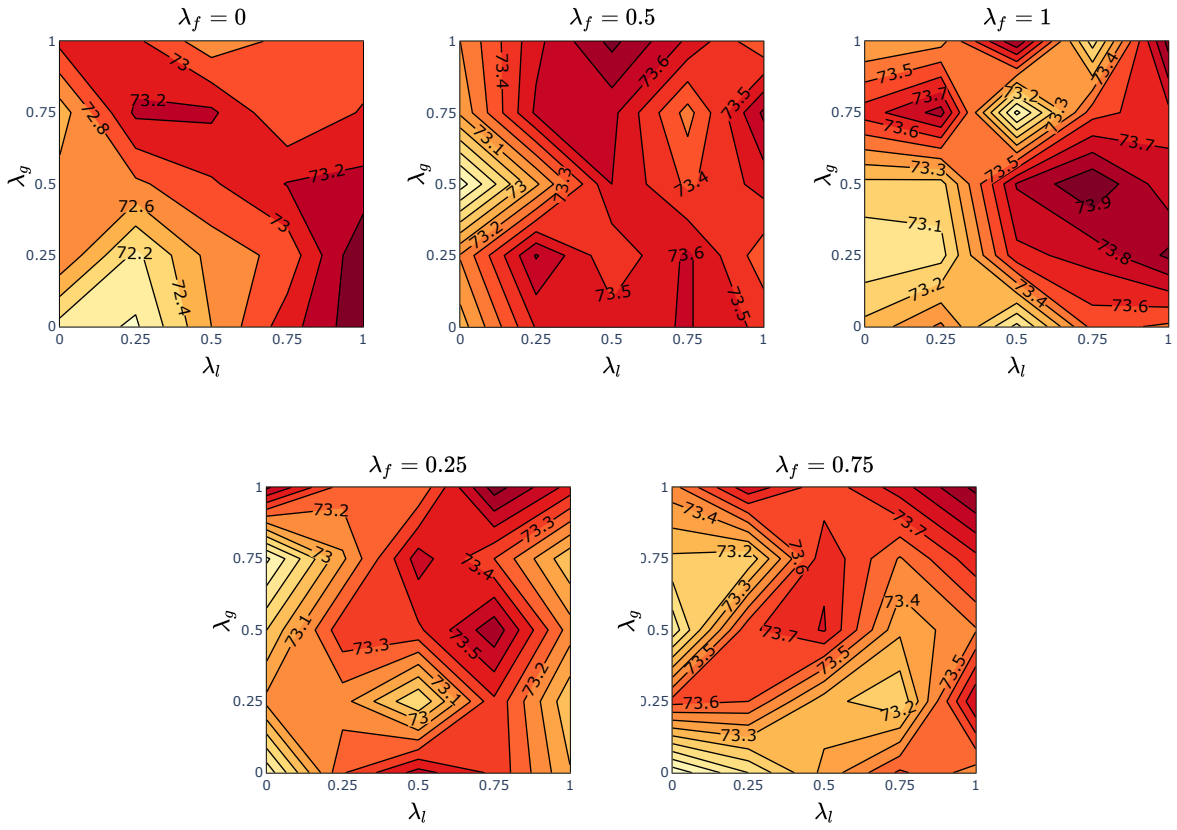

Figure 5. Results from the ablation studies on CIFAR100 dataset using a student resnet8x4 (baseline acc. 72.44%) with teacher resnet32x4 (baseline acc. 79.24%). Contour lines represent the final test accuracy of the student. Grid search was performed by varying the values of $\lambda_f, \lambda_g, \lambda_l$ from 0 to 1 with increments of 0.25. In each plot, the accuracy landscape is shown with λ_f set to a constant value.

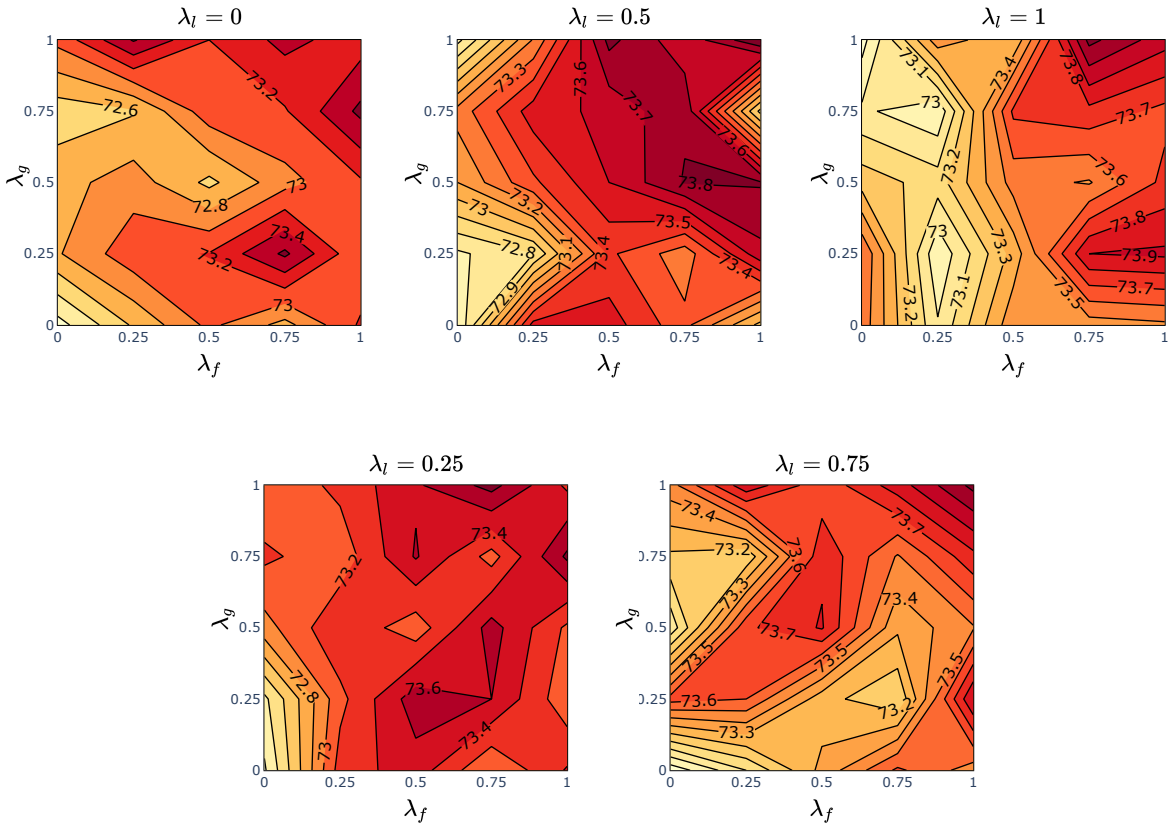

Figure 6. Results from the ablation studies on CIFAR100 dataset using a student resnet8x4 (baseline acc. 72.44%) with teacher resnet32x4 (baseline acc. 79.24%). Contour lines represent the final test accuracy of the student. Grid search was performed by varying the values of $\lambda_f, \lambda_g, \lambda_l$ from 0 to 1 with increments of 0.25. In each plot, the accuracy landscape is shown with λ_l set to a constant value.