Supplementary Material
DivClust: Controlling Diversity in Deep Clustering

Ioannis Maniadis Metaxas, Georgios Tzimiropoulos, Ioannis Patras
Queen Mary University of London
Mile End road, E1 4NS London, UK
{i.maniadismetaxas, g.tzimiropoulos, i.patras}@qmul.ac.uk

1. Hyperparameters & Hyperparameter Tuning

Diversity target D^T: The diversity target D^T, set by the user, is used to indicate how diverse the user wants the clusterings learned by DivClust to be. Specifically, given a similarity metric D, D^T represents an upper bound to inter-clustering similarity. That is, for a target D^T, the expectation is that the measured inter-clustering similarity D^R of the clusterings learned by the model should be $D^R \leq D^T$. In the paper, we measure inter-clustering similarity D with the avg. NMI between pairs of clusterings, as shown in Eq. 6. Other similarity metrics, however, are also applicable, under the assumption that they decrease monotonically as the dynamic threshold d decreases.

Results presented in paper Tab. 3 demonstrate the effectiveness and robustness of DivClust for various diversity targets, both in terms of successfully controlling diversity and in terms of producing good consensus clustering outcomes. We note, however, that, in the context of ensemble clustering, identifying the optimal degree of inter-clustering diversity is an open problem [3, 4] and *beyond* the scope of this work, which proposes a robust method for *controlling* diversity in deep clustering frameworks.

Memory bank size M: As mentioned in Sec. 3 of the paper, in order to update the upper similarity threshold d, the inter-clustering similarity score D^R of the learned clusterings must be calculated. This can be highly inefficient for large datasets, as this operation can have very high computational cost. Therefore, to mitigate this problem, we measure inter-clustering similarity over a memory bank, rather than over the entire dataset. Specifically, the memory bank stores cluster assignments for the M samples last seen by the model. The size M of the memory bank should be sufficient for the memory bank to contain a representative subset of the dataset, while taking into account the inherent trade-off with regard to performance. In all our experiments we set the size of the memory bank to $M = 10,000$, which we find sufficient, as our largest datasets (CIFAR10 and CIFAR100) have 60,000 samples.

Dynamic upper bound update interval T: The dynamic upper bound d is updated regularly, based on the measured inter-clustering similarity D^R, estimated over the memory bank. Specifically, it decreases when $D^R > D^T$ and increases otherwise, as outlined in paper Eq. 7. That calculation and the update of d are executed every T steps, set to $T = 20$ in all our experiments. Increasing this value would lead to more frequent updates of d and a corresponding increase in the computational cost of DivClust, as the inter-clustering similarity D^R would be measured more times during training. We found that $T = 20$ provides frequent enough updates to achieve the desired diversity target D^T across datasets and deep clustering frameworks, with acceptable computational cost.

Upper bound momentum hyperparameter m: This parameter regulates how big the steps of the upper bound threshold d in either direction are, when the diversity target D^T is/is not satisfied. We note that higher values might lead to instability due to large changes in d, however we again found that our initial choice of $m = 0.01$ worked well across datasets and frameworks.

The default values for the hyperparameters M, T and m were fixed and proved robust across datasets and base clustering frameworks. We note that *no hyperparameter tuning* was found to be necessary when incorporating DivClust to the deep clustering frameworks PICA [5], IIC [6] and CC [8], which highlights DivClust's plug-and-play nature. Indeed, other than duplicating the projection heads of each architecture to produce multiple clusterings, in our experiments we used the same hyperparameters as those reported in the respective papers of the base deep clustering frameworks, including the number of training epochs. More specifically, all three frameworks (IIC [6],

Dataset	Samples	Image size	Classes
CIFAR10	60,000	32X32	10
CIFAR100	60,000	32X32	100 (20)
ImageNet-Dogs	19,500	96X96	15
ImageNet-10	13,000	96X96	10

Table 1. A summary of the datasets used in the paper. We note that for CIFAR100 we use the 20 superclasses for evaluation.

PICA [5] and CC [8]) use a ResNet-34 architecture. IIC and PICA use Sobel preprocessing on all inputs and a linear projection head, while CC uses a 2-layer MLP projection head. CC resizes all images to $224X224$. IIC and CC train for 1,000 epochs, while PICA trains for 200. More details can be found in the respective papers.

2. Datasets

In this section we provide details for the datasets used in this work. We note that, in all cases, we train and evaluate on both the train and test sets, following convention in deep clustering works. A summary of the datasets is provided in Tab. 1.

CIFAR10 [7]: An image dataset with 60,000 images, split to 50,000 and 10,000 between the train and test sets. The dataset has 10 classes, and the size of the images is 32X32.

CIFAR100 [7]: An image dataset with 60,000 images, split to 50,000 and 10,000 between the train and test sets. The dataset has 100 classes, organized in 20 superclasses, and the size of the images is 32X32. Following previous works, we evaluate with the 20 superclasses.

ImageNet-Dogs [1]: A dataset consisting of 19,500 images of dogs organized in 15 classes. Samples were extracted from the ImageNet [2] dataset, and their size is 96X96.

Imagenet-10 [1]: A dataset of 13,000 96X96 images in 10 randomly chosen classes, extracted from the ImageNet [2] dataset. We note that we use the same classes as previous works [1, 8] for fair comparisons.

3. Complexity & Runtime

Complexity: As stated in paper Sec. 5, the complexity of DivClust is $O(nK^2C^2)$, where n is the batch size, K is the number of clusterings, and C is the number of clusters in each clustering. Importantly, given fixed hyperparameters n, K and C, the computational cost of DivClust is fixed, regardless of the size of the model and the dimensionality of the input data. Therefore, DivClust is scalable to large datasets and deep learning architectures.

K	D^T	T	Time (h)	Time Increase (%)
1	1.	-	39.1	0
20	1.	-	40.5	3%
20	0.9	20	44.6	14%

Table 2. Runtimes of CC, for 1000 epochs, with CIFAR100 and image size 224X224 during training.

Runtime Analysis: To analyze the practical impact of DivClust we first present runtimes with CC [8] on CIFAR100 in Tab. 2. The experiments were conducted with CC's default settings of 1000 epochs and images resized to 224X224 during training. We present results for $K = 1$ clustering (the default CC framework), $K = 20$ *without* DivClust (where $D^T = 1$ so the diversity loss is not used and d is not updated), and $K = 20$ *with* DivClust ($D^T = 0.9$). The update interval for d is set to the default $T = 20$. We note that, in terms of runtime, the specific value of the diversity target D^T does not have an impact, as long as $D^T < 1$. To provide a more robust analysis of DivClust's components with regard to their computational cost, in Tab. 3 we explore the impact of a) the dimensionality of the input data, and b) the frequency of the updates of d. Specifically, we train CC for 10 epochs (2,340 steps) with the standard image size for CIFAR100, namely 32X32, and include results for a less frequent update of d, where $T = 200$. All experiments were conducted on a single RTX6000 GPU.

For completeness, in addition to the experiments of Tabs. 2 and 3, which were conducted specifically for runtime analysis while ensuring that interference in their machine was kept at a minimum, we present approximate runtime figures for each dataset and framework *with DivClust* in Tab. 4.

Conclusions: Based on the complexity of the framework and the results presented in Tabs. 2 and 3, we note the following:

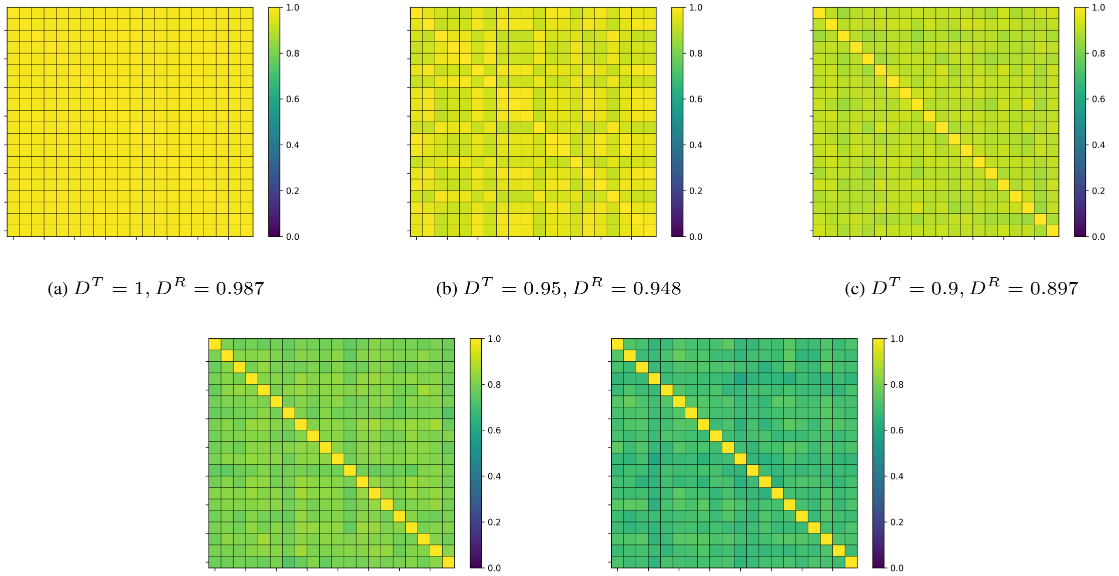
- The practical impact of DivClust in terms of increased training time is very small. Specifically, as seen in Tab. 2, CC with DivClust requires 44.6 hours to train, as apposed to 39.1 hours without DivClust (a 14% increase). For comparison, the alternative of running the model 20 times would require *32 days*, and

K	D^T	T	Time (s)	Time Increase (%)
1	1.	-	141	0
20	1.	-	161	14%
20	0.9	200	166	17%
20	0.9	20	209	48%

Table 3. Runtimes of CC, for 10 epochs, with CIFAR100 and image size 32X32 during training.

Method	CIFAR10	CIFAR100	ImageNet-10	ImageNet-Dogs
IIC [6]	21	-	-	-
PICA [5]	6.5	-	-	-
CC [8]	44	44.5	14	22

Table 4. Runtimes in hours for various models and datasets, for 20 clusterings with DivClust, using the experiment configurations proposed in the respective papers.



(a) $D^T = 1, D^R = 0.987$

(b) $D^T = 0.95, D^R = 0.948$

(c) $D^T = 0.9, D^R = 0.897$

(d) $D^T = 0.8, D^R = 0.807$

(e) $D^T = 0.7, D^R = 0.696$

Figure 1. Visualizations of inter-clustering similarity for ImageNet-10 for various diversity targets D^T. Specifically, the heatmaps in each figure represent the NMI between individual clusterings in the corresponding clustering set. For each D^T, we also report the measured avg. inter-clustering NMI D^R of the learned clusterings. The figure illustrates how reduced diversity targets D^T (and, accordingly, reduced inter-clustering similarity D^R) result in more diverse clusterings. Best seen in color.

would offer no control over the outcome in terms of inter-clustering diversity.

- Given that the computational cost of DivClust is independent of the model's backbone, its relative impact decreases for larger models and/or input dimensionality, given fixed n, C and K. That is evident by comparing Tabs. 2 and 3, where increasing the size of the input images from 32X32 (Tab. 3) to 224X224 (Tab. 2) decreases the relative runtime increase from 48% to 14%, as the backbone's load increases while DivClust's remains fixed. This makes DivClust well suited for deep model architectures.
- Experiments for $K = 20$ *without* DivClust ($D^T = 1$) were faster than experiments *with* DivClust ($D^T < 1$) by a small margin, which is to be expected. However, as was shown in Sec. 4 of the paper, without DivClust clusterings tend to converge to the same solution. Therefore, this approach is unsuitable for producing multiple, diverse clusterings, and, by extension, unsuitable for consensus clustering.

Overall, the computational cost produced by DivClust is very small relative to that of the base deep clustering models. Furthermore, the relative impact of DivClust decreases for larger architectures. Therefore, DivClust can be considered to be a highly efficient and scalable method for producing diverse clusterings in the context of deep clustering.

4. Visualizing inter-clustering diversity

To illustrate the impact of DivClust, we present in Fig. 1 visualizations of the diversity between clusterings, for sets of clusterings produced by DivClust. Each subfigure in Fig. 1 corresponds to a set of 20 clusterings produced by DivClust combined with CC, trained on ImageNet-10 for various diversity targets D^T. Specifically, the subfigures

Dataset	D^T	CIFAR10			CIFAR100			ImageNet-10			ImageNet-Dogs		
Metric	NMI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
CC-Kmeans	-	0.654	0.698	0.523	0.429	0.405	0.235	0.792	0.841	0.669	0.457	0.444	0.284
CC-Kmeans/S	-	0.674	0.69	0.554	0.428	0.402	0.228	0.792	0.842	0.673	0.456	0.444	0.283
CC-Kmeans/F	-	0.684	0.762	0.599	0.438	0.409	0.210	0.797	0.847	0.685	0.458	0.444	0.285
CC	-	0.705	0.790	0.637	0.431	0.429	0.266	0.859	0.893	0.822	0.445	0.429	0.274
DeepCluE	-	0.727	0.764	0.646	0.472	0.457	0.288	0.882	0.924	0.856	0.448	0.416	0.273
Mean	1.	0.678	0.763	0.604	0.418	0.427	0.257	0.859	0.895	0.824	0.457	0.451	0.297
Max		0.679	0.763	0.605	0.423	0.427	0.261	0.861	0.896	0.825	0.459	0.453	0.299
DivClust A		0.678	0.763	0.604	0.418	0.425	0.257	0.858	0.894	0.823	0.458	0.453	0.298
DivClust B		0.678	0.763	0.604	0.418	0.424	0.267	0.858	0.895	0.823	0.459	0.452	0.298
DivClust C		0.678	0.763	0.604	0.418	0.424	0.257	0.86	0.895	0.825	0.459	0.451	0.298
Mean	0.95	0.678	0.762	0.603	0.43	0.435	0.276	0.87	0.914	0.848	0.459	0.449	0.296
Max		0.688	0.773	0.616	0.433	0.447	0.28	0.914	0.963	0.92	0.461	0.452	0.298
DivClust A		0.683	0.768	0.61	0.43	0.434	0.276	0.916	0.964	0.922	0.452	0.461	0.298
DivClust B		0.679	0.762	0.603	0.431	0.435	0.277	0.863	0.898	0.828	0.46	0.451	0.297
DivClust C		0.677	0.76	0.602	0.431	0.434	0.276	0.891	0.936	0.878	0.461	0.451	0.297
Mean	0.9	0.703	0.794	0.644	0.422	0.43	0.262	0.861	0.903	0.832	0.471	0.479	0.323
Max		0.731	0.818	0.681	0.429	0.438	0.27	0.917	0.965	0.924	0.483	0.493	0.34
DivClust A		0.731	0.817	0.681	0.42	0.429	0.259	0.917	0.965	0.924	0.453	0.486	0.335
DivClust B		0.708	0.799	0.653	0.422	0.431	0.262	0.866	0.908	0.837	0.477	0.486	0.33
DivClust C		0.678	0.789	0.641	0.422	0.426	0.258	0.879	0.92	0.859	0.48	0.487	0.332
Mean	0.8	0.675	0.782	0.632	0.419	0.417	0.26	0.816	0.84	0.754	0.455	0.45	0.296
Max		0.762	0.847	0.727	0.429	0.434	0.275	0.858	0.909	0.83	0.487	0.509	0.347
DivClust A		0.762	0.847	0.727	0.419	0.42	0.275	0.835	0.845	0.779	0.486	0.504	0.347
DivClust B		0.714	0.807	0.664	0.419	0.414	0.258	0.878	0.919	0.851	0.459	0.453	0.298
DivClust C		0.724	0.819	0.681	0.422	0.414	0.26	0.879	0.918	0.851	0.458	0.448	0.296
Mean	0.7	0.645	0.703	0.556	0.43	0.425	0.267	0.742	0.747	0.643	0.458	0.453	0.298
Max		0.704	0.789	0.678	0.459	0.469	0.304	0.798	0.83	0.743	0.49	0.512	0.352
DivClust A		0.677	0.773	0.621	0.441	0.446	0.286	0.798	0.83	0.743	0.476	0.46	0.318
DivClust B		0.665	0.725	0.621	0.434	0.438	0.272	0.875	0.916	0.837	0.492	0.456	0.315
DivClust C		0.71	0.815	0.675	0.44	0.437	0.283	0.85	0.90	0.819	0.516	0.529	0.376

Table 5. Results combining DivClust with CC for various diversity targets D^T and for various methods of extracting single clustering solutions. We underline DivClust results that outperform the single-clustering baseline CC.

Method	Clusterings	D^T	Mean Acc.	Max. Acc.	Cons. Acc.
CC	1	-	0.893	0.893	0.893
CC-20x	20	-	0.891	0.895	0.894
DivClust	20	1.	0.895	0.896	0.895
	20	0.95	**0.914**	0.963	**0.936**
	20	0.9	0.903	**0.965**	0.92
	20	0.8	0.84	0.909	0.918
	20	0.7	0.747	0.83	0.9

Table 6. Results on Imagenet-10 for the baseline single-clustering method CC, for 20 clusterings learned by training CC 20 times with different seeds (**CC-20x**), and for DivClust with various diversity targets D^T. We note the best results with **bold**.

consist of 20X20 matrices, where each value (i, j) represents the NMI between clusterings i and j, with higher values corresponding to more similar clusterings.

In Fig. 1, one can see that decreasing the diversity target D^T indeed results to less similar clusterings. Furthermore, one can see that the similarities between pairs of clusterings

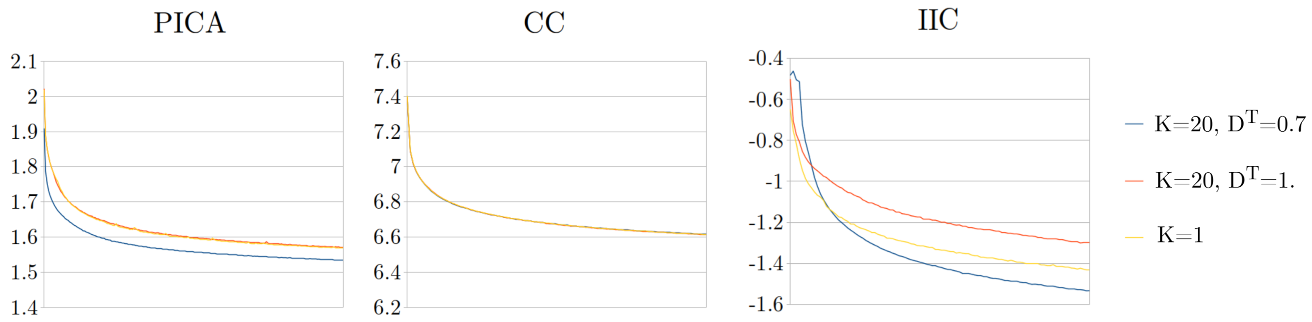


Figure 2. The training loss L_{total} for PICA, CC and ICC, trained on CIFAR10 to learn a single clustering (K=1), multiple clusterings *without* diversity (K=20, $D^T = 1$) and multiple clusterings *with* diversity (K=20, $D^T = 0.7$). Best seen in color.

are not uniform. That is, they are not all equally diverse with each other. This reflects the fact that DivClust controls the *avg.* inter-clustering similarity, therefore individual pairs of clusterings may have a higher similarity score than D^T, as long as the avg. similarity score D^R is lower than D^T. We note that it is trivial to modify DivClust's loss to enforce diversity between each pair of clusterings. However, for the purposes of consensus clustering, the more relaxed constraint of controlling diversity on the aggregate was preferred.

5. Extended CC results

In this section, detailed results are presented for experiments combining DivClust with CC. Following the methodology outlined in Section 4 of the paper, Tab. 5 includes results for CIFAR10, CIFAR100, ImageNet-Dogs and ImageNet-10, reported for each of the three proposed methods for extracting single clustering solutions, namely **DivClust A** (selecting the clustering k with the lowest loss $L_{main}(k)$), **DivClust B** (applying consensus clustering), and the method we found to be the most robust, **DivClust C** (selecting the 10 best clusterings in terms of their loss, and applying consensus clustering on them). In Tab. 5, we also include the mean/max values of each metric over the clustering ensembles produced for each setting, noting that, in practice, identifying clusterings whose performance matches those values is non-trivial, as we assume that we do not have access to the labels.

Finally, in Tab. 6, we present results on Imagenet-10 for DivClust trained with various diversity targets D^T, comparing it with the single-clustering baseline CC and with a clustering ensemble produced by training a single-clustering model 20 times with different seeds (**CC-20x**). In all cases, the consensus clustering solution was produced by identifying the 10 best performing clusterings of each set with regard to their loss, and applying the SCCBG [9] consensus clustering algorithm. Tab. 6 demonstrates that, despite requiring approximately 20X more training time, producing the ensemble from multiple individually trained models leads to minimal performance gains over the baseline, as opposed to DivClust, which consistently outperforms the baseline in terms of consensus clustering accuracy.

6. Joint optimization and convergence analysis

To further demonstrate that DivClust can be straightforwardly integrated in deep clustering frameworks, we analyze its behavior with regard to the training loss and its convergence. Specifically, in Fig. 2, we present the total loss L_{main} during training for the three deep clustering frameworks CC [8], PICA [5] and IIC [6]. The frameworks are applied on CIFAR10 and trained to learn a) a single clustering, b) multiple clusterings (K=20) without diversity requirements ($D^T = 1$), and c) multiple clusterings with diversity ($D^T = 0.7$).

We observe that different frameworks do not behave in exactly the same way. Specifically, while CC's loss curve remains virtually identical in all three examined cases, PICA and IIC converge to different loss values when DivClust is active (i.e. when $D^T = 0.7$). We attribute this to the frameworks' different objectives and architectures. However, in all cases, the loss converges smoothly, which indicates that our proposed loss L_{div} can be optimized jointly with each framework's base loss L_{main} without requiring adjustments and without disturbing the training process.

References

[1] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887, 2017. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[3] Xiaoli Z Fern and Wei Lin. Cluster ensemble selection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 1(3):128–141, 2008. 1

[4] Stefan T Hadjitodorov, Ludmila I Kuncheva, and Ludmila P Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006. 1

[5] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8849–8858, 2020. 1, 2, 3, 5

[6] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019. 1, 3, 5

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[8] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *2021 AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 1, 2, 3, 5

[9] Peng Zhou, Liang Du, and Xuejun Li. Self-paced consensus clustering with bipartite graph. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2133–2139, 2021. 5