Active Learning by Feature Mixing (Supplements)

Amin Parvaneh[1] Ehsan Abbasnejad[1] Damien Teney[1,2] Reza Haffari[3],
Anton van den Hengel[1,4] Javen Qinfeng Shi[1]

[1]Australian Institute for Machine Learning, University of Adelaide
[2]Idiap Research Institute [3]Monash University [4]Amazon
{amin.parvaneh, ehsan.abbasnejad,javen.shi, anton.vandenhengel}@adelaide.edu.au
damien.teney@idiap.ch gholamreza.haffari@monash.edu

1. Methodology

Details of Eq. (2) in the main text. We can write the first-order Taylor expansion of the loss for an interpolation w.r.t. $\boldsymbol{z}^u$ as:

$$\ell\left(f_c\left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right), y^*\right) \approx \ell\left(f_c(\boldsymbol{z}^u), y^*\right) + \left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{z}^u\right)^\intercal . \nabla_{\boldsymbol{z}^u} \ell\left(f_c\left(\boldsymbol{z}^u\right), y^*\right) . \tag{1}$$

We also know that considering $\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u$, we will have

$$\begin{aligned} \tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{z}^u &= (\boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u) - \boldsymbol{z}^u \\ &= \boldsymbol{\alpha}\boldsymbol{z}^\star + \boldsymbol{z}^u - \boldsymbol{\alpha}\boldsymbol{z}^u - \boldsymbol{z}^u \\ &= \boldsymbol{\alpha}\boldsymbol{z}^\star - \boldsymbol{\alpha}\boldsymbol{z}^u \\ &= \boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u) . \end{aligned} \tag{2}$$

By replacing this in Eq. (1), we have

$$\ell\left(f_c\left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right), y^*\right) \approx \ell\left(f_c(\boldsymbol{z}^u), y^*\right) + \left(\boldsymbol{\alpha}\left(\boldsymbol{z}^\star - \boldsymbol{z}^u\right)\right)^\intercal . \nabla_{\boldsymbol{z}^u} \ell\left(f_c\left(\boldsymbol{z}^u\right), y^*\right) . \tag{3}$$

which uncovers Eq. (2) in the main text.

Details of Eq. (5) in the main text. As stated in section 3.3 of the main text, using a 2-norm constraint on $\boldsymbol{\alpha}$, we approximate the optimum interpolation ratio as

$$\boldsymbol{\alpha}^* = \underset{\|\boldsymbol{\alpha}\|_2 \leq \epsilon}{\arg\max} \left(\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\right)^\intercal . \nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*). \tag{4}$$

By multiplying both sides of the constraint in Eq. 4 by $\|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2$, we have

$$\|\boldsymbol{\alpha}\|_2 \, \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \epsilon \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2.$$

Based on Cauchy-Schwartz inequality, we know that $\|\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \|\boldsymbol{\alpha}\|_2 \, \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2$. Thus, we can infer

$$\|\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \epsilon \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 = \epsilon'.$$

Therefore, we can change the optimisation problem to

$$\boldsymbol{\alpha}^* = \underset{\|\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \epsilon'}{\arg\max} \left(\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\right)^\intercal . \nabla_{\boldsymbol{z}^u} \ell\left(f_c(\boldsymbol{z}^u), y^*\right) .$$

We can use the dual norm [3] of the above equation to approximate the optimum value for $\boldsymbol{u} = \boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)$, which is

$$\boldsymbol{u}^* = \epsilon' \frac{\nabla_{\boldsymbol{z}^u} \ell\left(f_c(\boldsymbol{z}^u), y^*\right)}{\|\nabla_{\boldsymbol{z}^u} \ell\left(f_c(\boldsymbol{z}^u), y^*\right)\|_2}. \tag{5}$$

After replacing the actual values for $\boldsymbol{u}$ and ϵ', we have

$$\boldsymbol{\alpha}^* \approx \epsilon \frac{\|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)}{\|\nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)\|_2} \oslash (\boldsymbol{z}^\star - \boldsymbol{z}^u), \tag{6}$$

which reveals Eq. (5) in the main text ($\oslash$ indicates element-wise division).

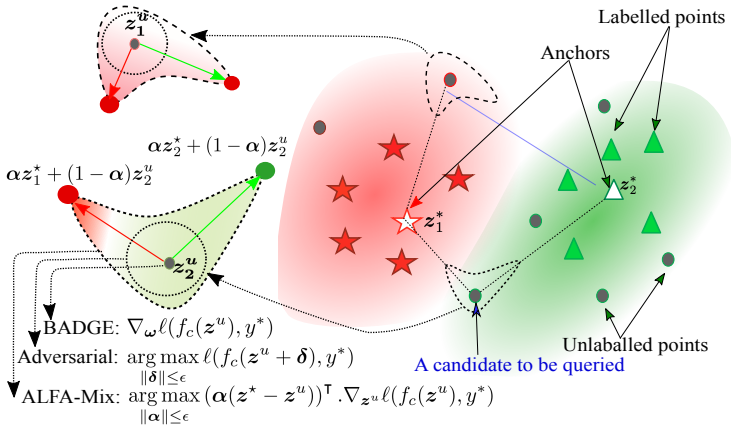
It is worth mentioning that the denominator in Eq. 6 cancels out when utilised for the interpolation and as such does not present any divide by zero problem in our approach. Consider that we use $\boldsymbol{\alpha}^*$ in the following:

$$\begin{aligned} \tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}^*} &= \boldsymbol{\alpha}^*\boldsymbol{z}^\star + (1-\boldsymbol{\alpha}^*)\boldsymbol{z}^u \\ &= \boldsymbol{z}^u + \boldsymbol{\alpha}^*(\boldsymbol{z}^\star - \boldsymbol{z}^u) , \end{aligned} \tag{7}$$

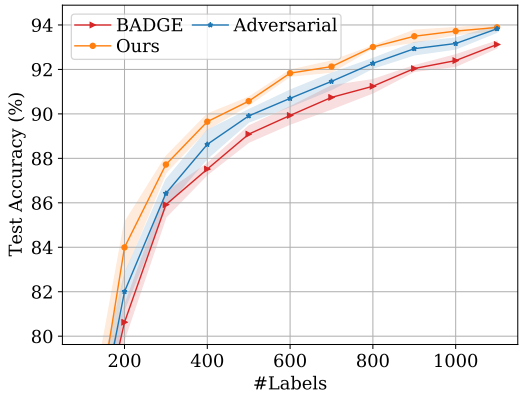
where its value is obtained from the closed-form solution in Eq. 6. When considering both together, it is easy to see that the denominator simply cancels out with $(\boldsymbol{z}^\star - \boldsymbol{z}^u)$. In practice, adding a very small constant to the denominator provides numerical stability and resolves this issue.

1.1. Relations Between ALFA-Mix and Other Baselines

Using gradients in BADGE: From Eq. (3) in the main text we can understand that when the prediction is accurate and



(a) Sampling strategies.



(b) Results on MNIST dataset using an MLP and a small budget of size 100 at each round.

Figure 1. A comparative depiction of our approach (ALFA-Mix) vs. BADGE vs. adversarial in the latent space: Since ours considers interpolations in the direction of the anchor points and proportional to their distance, it better evaluates the consistency of the predictions in the latent space. When points are less consistent, it is more intuitive to consider them as candidates to be queried (*e.g.* z_2^u in this figure is inconsistent after the interpolation, and hence likely to be queried).

confident, small movements of the latent representation towards different directions (declared by anchors) should not change the prediction. Otherwise, as per right-hand-side of the equation, either the surface has changed dramatically or the unlabelled features is far from the labelled representations (*i.e.* the features of the unlabelled instance are novel). This is one of the major differences of our approach when compared with BADGE that only relies on the gradients of the unlabelled instances (Figure. 1).

Adversarial perturbation of features: To show the importance of the feature interpolations with labelled representations in our approach, we also considered using adversarial noise as an alternative perturbation mechanism. For that, we examined adding small values of noise $\boldsymbol{\delta}$ to the latent representations of each unlabelled point (instead of using interpolations with anchors) to find inconsistencies in their predicted labels. Following Eq. (3) and Eq. (4) in the main text, we set the objective for finding the optimum noise vector $\boldsymbol{\delta}^*$ as:

$$\boldsymbol{\delta}^* = \underset{\|\boldsymbol{\delta}\| \le \epsilon}{\arg\max}\, \ell(f_c(\boldsymbol{z}^u + \boldsymbol{\delta}), y^*). \tag{8}$$

Similarly, using a first-order Taylor expansion w.r.t. $\boldsymbol{z}^u$ and its dual norm, we can approximate the optimum noise as

$$\boldsymbol{\delta}^* \approx \epsilon \frac{\nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)}{\|\nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)\|_2}. \tag{9}$$

After constructing a candidate set of unlabelled samples whose predicted labels are not consistent after the adversarial feature perturbation, we conduct clustering to sample a diverse set from the candidate set (similar to ALFA-Mix).

Interestingly, as depicted in Figure. 1b, although the adversarial approach shows better performance in comparison to BADGE, it falls behind considerably when compared to ALFA-Mix. We believe that the main advantage of ALFA-Mix is the consideration of both the novelty of the features and the extent of gradient at each unlabelled point. It is worth mentioning that ALFA-Mix is able to identify more inconsistencies all over the decision boundary (Figure. (6c) in the main text).

Distribution matching. Denote $\Delta = \mathbb{E}_{p(\boldsymbol{z}^l|\mathcal{D}^l)}\left[\boldsymbol{z}^l\right] - \mathbb{E}_{p(\boldsymbol{z}^u|\mathcal{D}^u)}\left[\boldsymbol{z}^u\right]$ if we had the distributions in the latent space. We know that based on the definition of the interpolation between a pair of labelled and unlabelled samples (*i.e.* $\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\boldsymbol{z}^l + (1-\boldsymbol{\alpha})\boldsymbol{z}^u$), we can have

$$\boldsymbol{z}^u = \frac{1}{1-\boldsymbol{\alpha}} \left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\boldsymbol{z}^l\right).$$

By taking the expectation from both side of the above equation for all the labelled samples we have

$$\boldsymbol{z}^u = \mathop{\mathbb{E}}_{p(\boldsymbol{z}^l|\mathcal{D}^l)} \left[\frac{1}{1-\boldsymbol{\alpha}} \left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\boldsymbol{z}^l\right)\right].$$

After replacing this in the definition of Δ, it is easy to show that:

$$\Delta = \frac{1}{(1-\boldsymbol{\alpha})} \left(\mathop{\mathbb{E}}_{p(\boldsymbol{z}^l|\mathcal{D}^l)} \left[\boldsymbol{z}^l\right] - \mathop{\mathbb{E}}_{p(\boldsymbol{z}^u|\mathcal{D}^u)} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{z}^l|\mathcal{D}^l)} \left[\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right] \right] \right).$$

That is, the interpolation operation we used here only affects difference of the expectation of distributions with a constant factor. When seen in light of Eq. (1) in the main text, it acts as a simple surrogate for a divergence measure. In fact, this

Dataset	Pool Size	Label Size	Input	Initial Instances	Budgets	Architectures	Initialisations
MNIST [8]	50,000	10	28×28	100	100, 1000	MLP, LeNet-5	Random, Continue**
EMNIST [5]	124,800	26	28×28	260	260, 2650	MLP, LeNet-5	Random, Continue
SVHN [9]	50,000	10	32×32	100	100, 1000	ResNet-18, DenseNet-121	Random
CIFAR10 [7]	50,000	10	32×32	100	100, 1000	ResNet-18, DenseNet-121	Random
DomainNet-Real-10*	4,673	10	224×224	100	100	ResNet-18, DenseNet-121	Pre-trained
DomainNet-Real-20*	8,615	20	224×224	200	200	ResNet-18, DenseNet-121	Pre-trained
CIFAR100 [7]	50,000	100	32×32	1000	1000	ViT-Small	Pre-trained
Mini-ImageNet [13]	48,000	100	84×84	1000	1000	ViT-Small	Pre-trained
DomainNet-Real [11]	122,563	345	224×224	3450	3450	ViT-Base, ResNet-18, DenseNet-121	Pre-trained
OpenML_6	18,000	26	16	100	100	MLP	Random
OpenML_155	50,000	9	10	100	100	MLP	Random

Table 1. A summary of diverse AL settings that we used in our image and non-image experiments. Overall, 30 different settings were utilised in our experiments to compare AL methods in various conditions.
* These are two small subsets of DomainNet-Real that has been used to compare AL methods on small datasets with high-resolution images.
**"Continue" represents the setting where the weights of the network initialise from those of the network trained in the previous round.

relates our approach to other AL methods that their focus is on finding the distributional difference between labelled and unlabelled samples [4, 14].

Gradient-based interpolation optimisation. Following [1, 10], we could have utilised iterative gradient-based optimisation to find the optimum interpolation ratios (instead of the closed-form solution used in ALFA-Mix). For that, motivated by the condition in the Eq. (6) in the main text where we are interested in instances whose predictions flip with an interpolation in the latent space, we can choose $\boldsymbol{\alpha}$ as a solution to the following:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha} \in [0, \alpha_{\max}]^D}{\arg\max}\ \ell(f_c(\boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u), y^*), \qquad (10)$$
$$\text{s.t.} \quad y^* = \underset{k \in \{1,\dots,K\}}{\arg\max}\ f_c^k(\boldsymbol{z}^u), \quad \forall \boldsymbol{z}^u \in \boldsymbol{Z}^u, \quad \boldsymbol{z}^\star \in \boldsymbol{Z}^\star,$$

where $\alpha_{\max}$ is a hyper-parameter governing the feature mixing ratios. Intuitively, this optimisation chooses the hardest case of $\boldsymbol{\alpha}$ for each unlabelled instance and anchor. We perform few iterations of projected gradient descent to optimise $\boldsymbol{\alpha}$. Our empirical studies reveal similar performances when using this objective in comparison to the closed-form one. However, the time required for the iterative gradient-based approach is much more than the closed-form one (*i.e.* when using 5 iterations of gradient update, it is 5x slower than ALFA-Mix).

2. Experiments

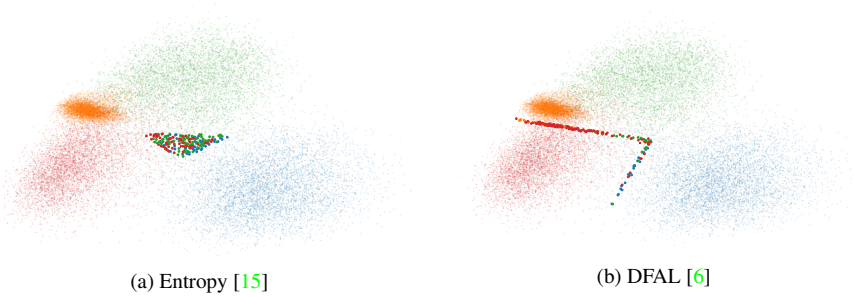
2.1. Comparison matrix

We demonstrate the performance comparison between every pair of AL methods over various settings in a penalty matrix proposed in [2]. Each cell of the matrix reveals the number of settings in which the method shown in the column is outperformed by the ones indicated in the row. It should be noted that each setting consists of conducting R rounds of AL with a specific labelling budget size B and using a particular model architecture on a single dataset. Since we repeat each setting with 5 different random seeds, at each round r in the setting we use t-score of the difference between the test performances ($d^r_{i,j} = a^r_i - a^r_j$) of each pair of AL methods (i, j) over the 5 repeats:

$$c^r_{i,j} = \frac{\sqrt{5}\mu^r}{\sigma^r}, \qquad (11)$$
$$\mu^r = \frac{1}{5}\sum_{m=1}^{5} d^r_{i,j}, \quad \sigma^r = \sqrt{\frac{1}{5}\sum_{m=1}^{5}(d^r_{i,j} - \mu^r)^2},$$

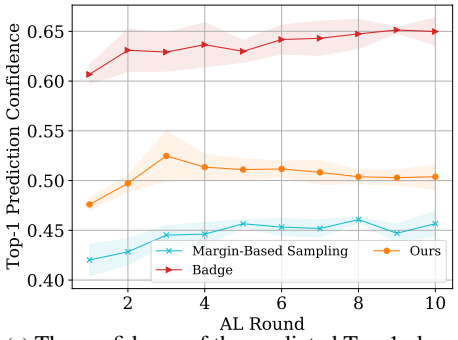
Factor	Variety	#Settings	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE
Data Type	Image	28	74%	63%	69%	80%	65%	36%	33%
	OpenML	2	95%	100%	90%	100%	100%	80%	55%
Architecture	MLP	8	98%	93%	96%	100%	99%	73%	64%
	LeNet-5	5	100%	70%	74%	98%	66%	34%	14%
	ResNet-18	7	61%	63%	60%	80%	51%	27%	24%
	DenseNet-121	7	67%	59%	64%	83%	56%	24%	23%
	ViT	3	100%	83%	100%	100%	100%	43%	60%
Initialisation	Random	18	72%	69%	65%	89%	62%	35%	29%
	Pre-Training	9	99%	72%	97%	91%	88%	40%	43%
	Continue	3	100%	100%	90%	100%	87%	83%	57%
Budget	Small	22	86%	84%	86%	92%	82%	52%	46%
	Large	8	74%	43%	53%	88%	46%	11%	9%
Overall		30	83%	73%	77%	91%	72%	41%	36%

Table 2. The percentage of the AL rounds in different settings where ALFA-Mix outperforms other baselines, considering their victory scores [2]. The chart of the same results is depicted in Figure. 3 of the main text.

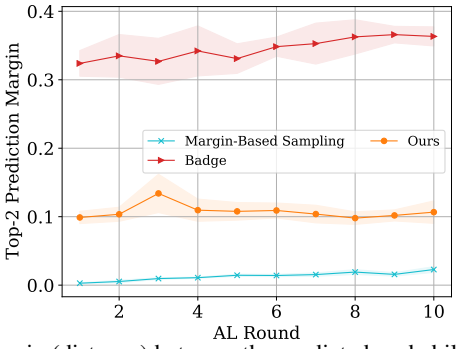


(a) Entropy [15] (b) DFAL [6]

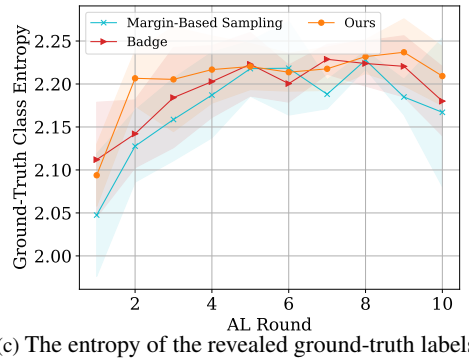
Figure 2. Visualization of sample selection behaviours of some AL methods in the latent space (other methods can be found in Figure (2) of the main text).



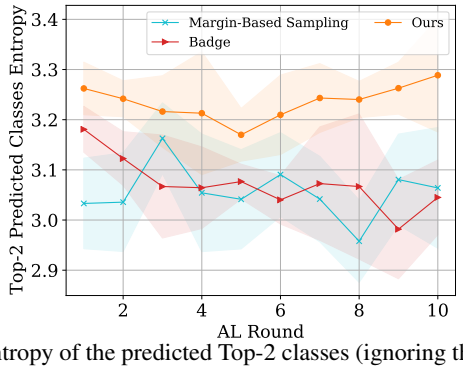
(a) The confidence of the predicted Top-1 class.



(b) The margin (distance) between the predicted probabilities of the Top-2 classes.



(c) The entropy of the revealed ground-truth labels.



(d) The entropy of the predicted Top-2 classes (ignoring the order of them).

Figure 3. Uncertainty and diversity of the selected samples for labelling. All experiments are done on MNIST dataset using LeNet-5 model and a small budget of size 100.

where a_i^r and a_j^r are the test performances of methods i and j respectively at AL round r. Similar to [2], we also used a threshold of 2.776 for this score to decide if method i wins over method j. After clarifying the winner at each round of the setting, we calculate $C_{i,j} = \sum_{r=1}^{R} \mathbb{1}_{c_{i,j}^r > 2.776}/R$ as the final victory score of AL method i over method j in that specific setting. Additionally, to compute the matrix over multiple settings, we simply report the element-wise sum of all the individual matrices.

2.2. Sampling Diversity and Uncertainty

To have a better understanding with regards to the effectiveness of our approach in selecting an uncertain and diverse set of samples for labelling, we compare some characteristics of the selected batch of instances at each AL round comparing our method with those from BADGE [2] and Margin-Based Sampling[1] [12] (Figure 3).

Comparing the confidence and Top-2 prediction margins of the selected unlabelled samples, depicted in Figures 3a and 3b respectively, we can see that the uncertainty level of the selected samples by our method is closer to the highest possible value in comparison to BADGE sampling. Please note that in contrast to what Margin-Based Sampling is doing, we do not explicitly enforce our approach to select samples close to the decision boundaries. On the other hand,

[1]Margin-Based Sampling is another AL method based on uncertainty. It selects samples with the lowest distance between the predicted probabilities for the Top-2 classes (called margin). It should be noted that BADGE has shown significantly better performance compared to Margin-Based Sampling in prior works [2].

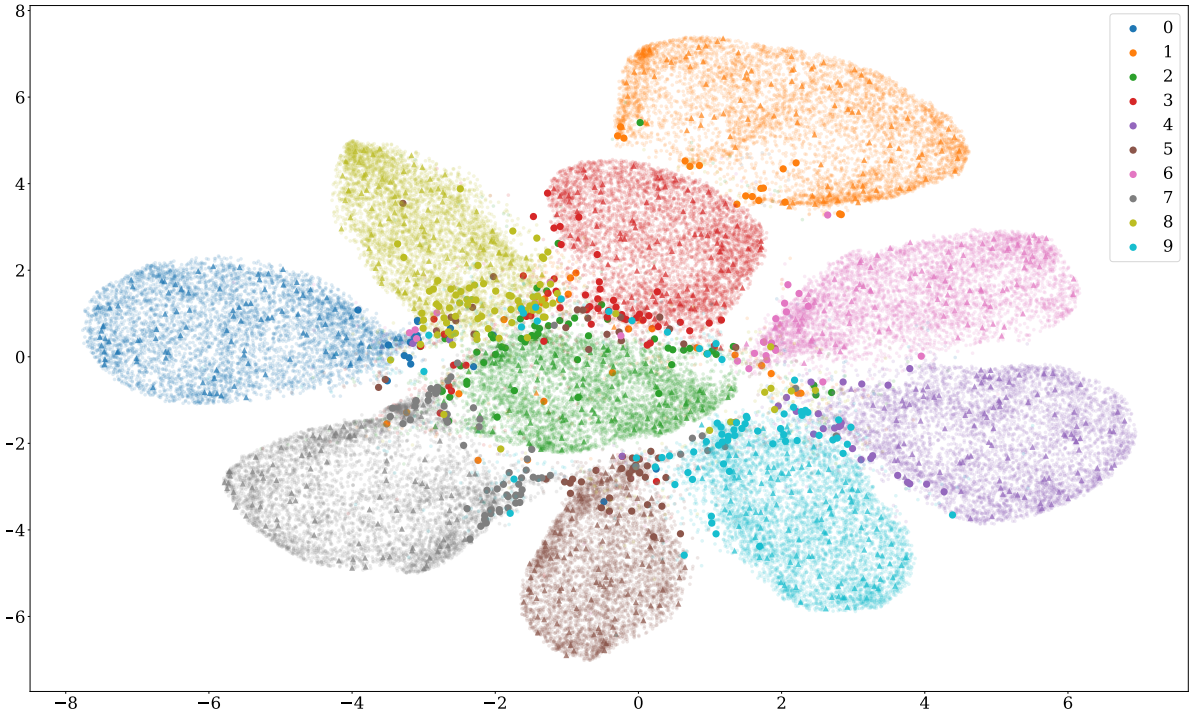
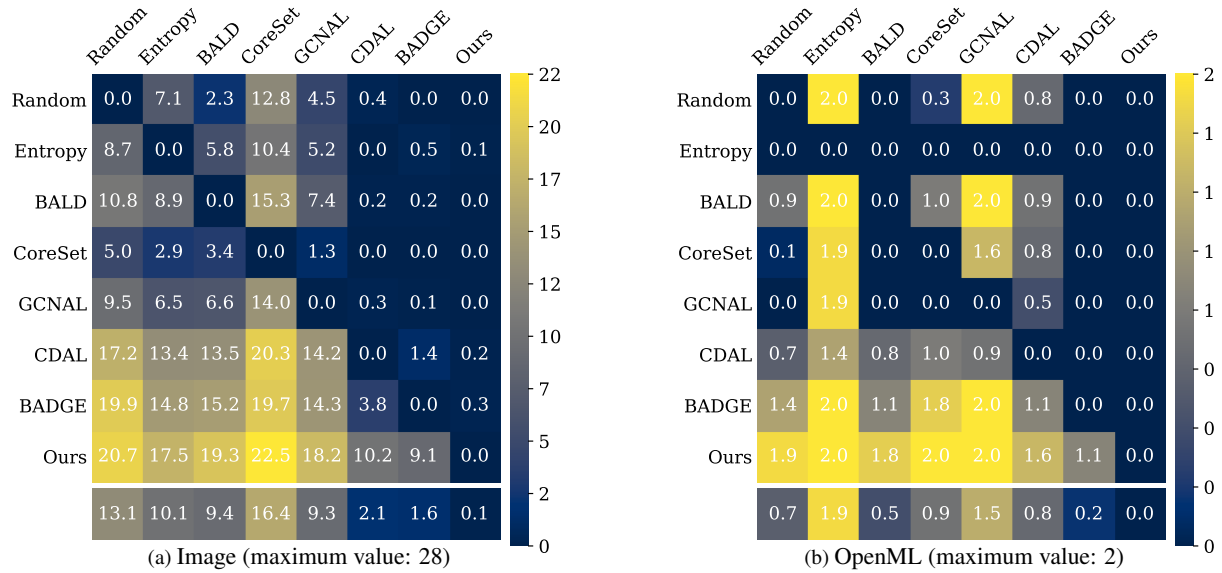


Figure 4. The t-SNE visualisation of the sample selection of our proposed method on MNIST dataset using LeNet-5. The model is trained based on 500 random labelled set (shown as triangles) and is provided with a budget of size 500 to (depicted as bold circles).



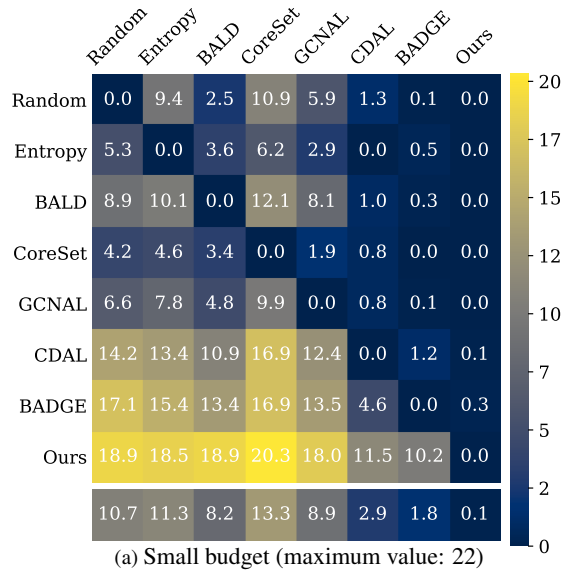
(a) Image (maximum value: 28) (b) OpenML (maximum value: 2)

Figure 5. Pairwise comparison of different AL approaches based on the type of data. The maximum value of each cell for each setting is also provided in the captions.

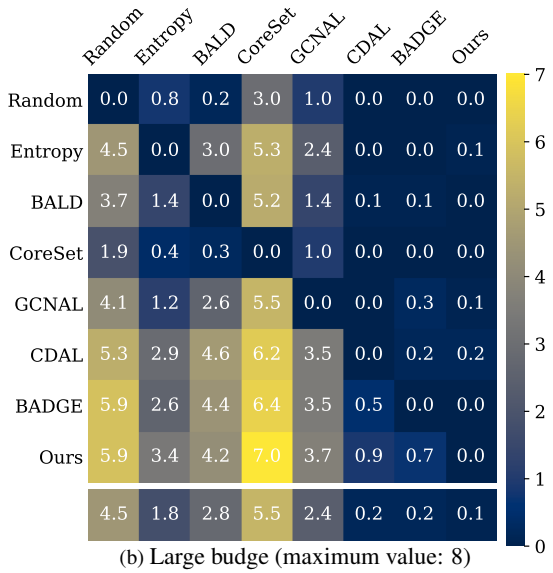
considering the higher entropy values in the ground-truth labels of the selected set and their Top-2 predicted classes, we can realise the capability of our proposed method in selecting a diverse set of unlabelled samples in terms of their true class labels and their position with regard to the decision boundaries. All in all, as depicted in Fig. 4, our method is able to exploit both uncertainty and diversity concepts to select a diverse set of samples that lie close to decision boundaries, which leads to significantly higher performances.

2.3. More Ablations

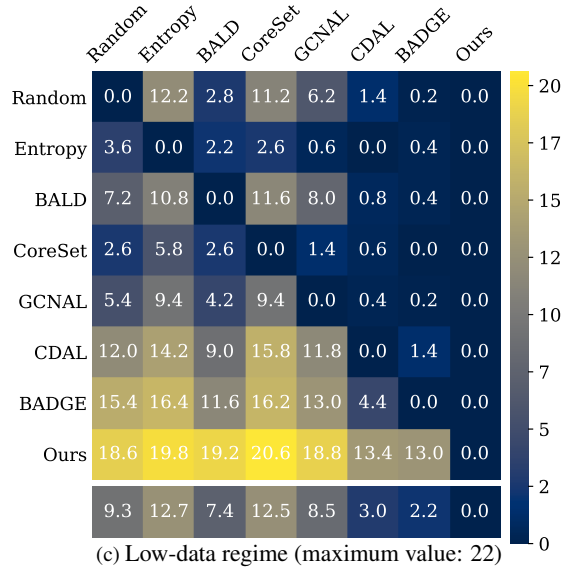
In addition to providing the percentage with which our approach outperforms others in each setting (Table. 2), we report the pairwise comparison of all the AL methods across various choices of data (Fig. 5), budget size (Fig. 6), model architecture (Fig. 7) and network initialisation method (Fig. 8. Further, in Figure 6c, we provide the pairwise comparisons in low-data regimes. Considering the values in the rows and columns corresponding to our approach, we can



(a) Small budget (maximum value: 22)



(b) Large budge (maximum value: 8)



(c) Low-data regime (maximum value: 22)

Figure 6. Pairwise comparison of different AL approaches based on different sizes of budget. The maximum value of each cell for each setting is also provided in the captions.

infer that our approach consistently outperforms all other baselines regardless of the architecture, dataset selection, network initialisation and budget size and is rarely beaten by others.

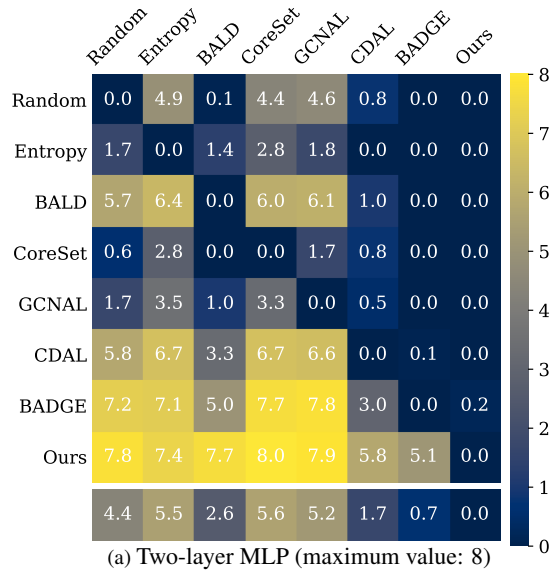
2.4. All the Experiments

We compare our approach with other baselines over a total of 285 AL rounds in 30 different settings, with each setting identified by a specific combination of dataset, budget size, model architecture, and model initialisation method. Table 2 demonstrates details of each setting we employed in our experiments.

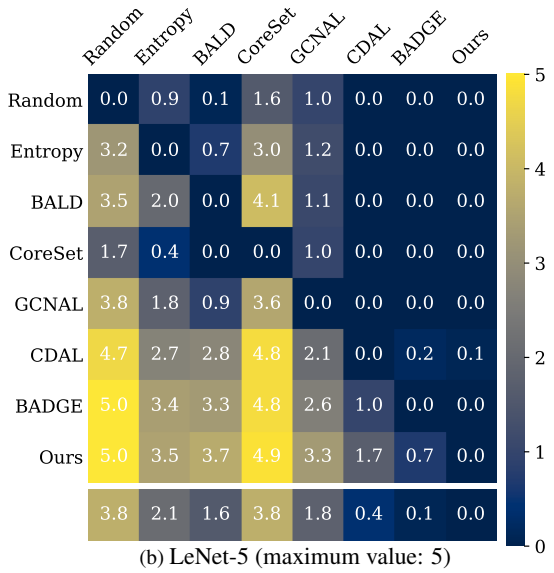
In our approach, we set $\epsilon = \frac{0.2}{\sqrt{D}}$, where D is the dimentionality of $\boldsymbol{\alpha}$ vector. Considering the norm condition in Eq. 4 in the main text, we relate the scale of ϵ to D to easily utilise the same hyper-parameter across different networks with representations of variable dimensions.

All the experiments for small datasets were carried out on a NVIDIA GEFORCE GTX 1080 Ti, while for larger datasets we used an NVIDIA QUADRO RTX 8000. It is worth mentioning that for the video experiments, we utilised two NVIDIA V100 GPUs in parallel.

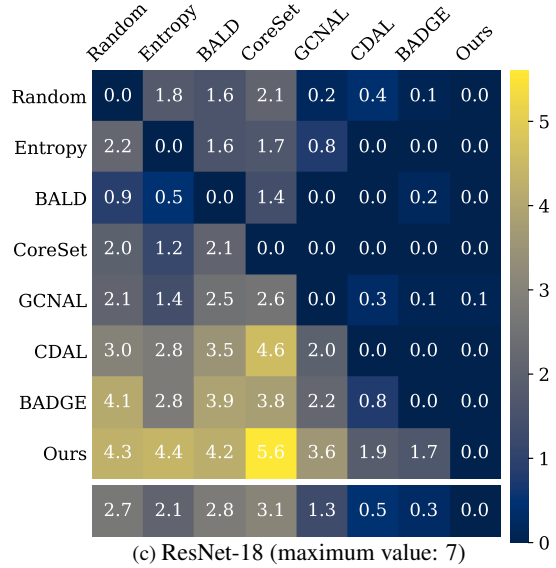
We borrowed the implementations of the baselines from their publicly provided codes. The MLP network we employed in our experiments follows the architecture proposed in [2]: a two-layer Perceptron with ReLU activations and an embedding dimension of size 256 for image datasets (*i.e.* MNIST and EMNIST). Similarly, we expanded the embedding dimensionality to 1024 for OpenML datasets. We include the accuracy curves over the unseen test set for all the settings.



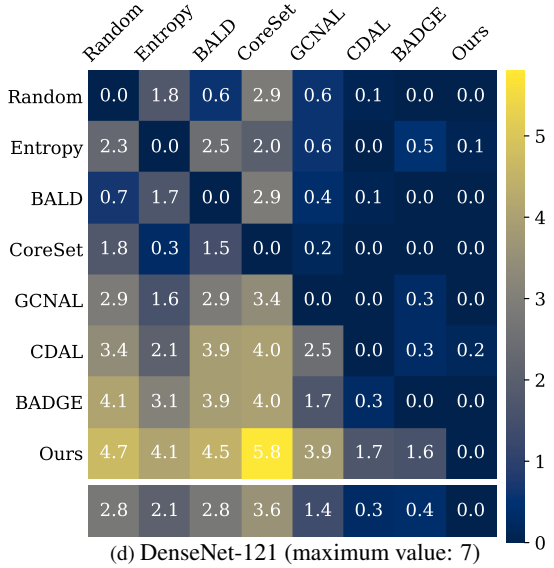
(a) Two-layer MLP (maximum value: 8)



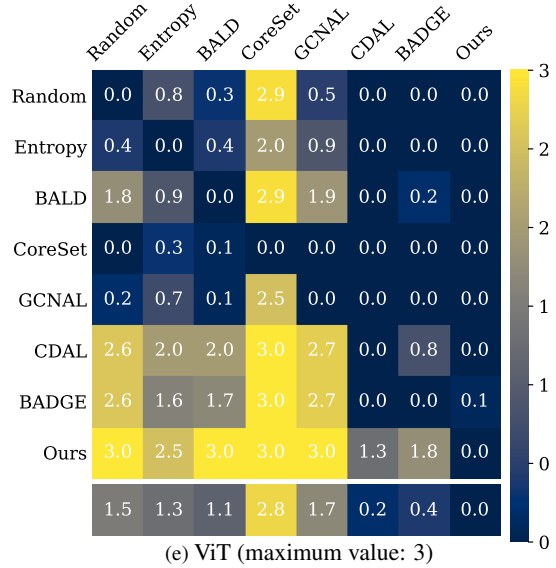
(b) LeNet-5 (maximum value: 5)



(c) ResNet-18 (maximum value: 7)

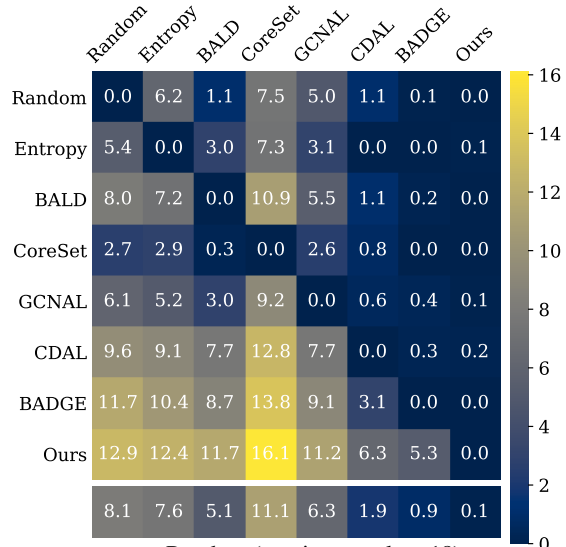


(d) DenseNet-121 (maximum value: 7)

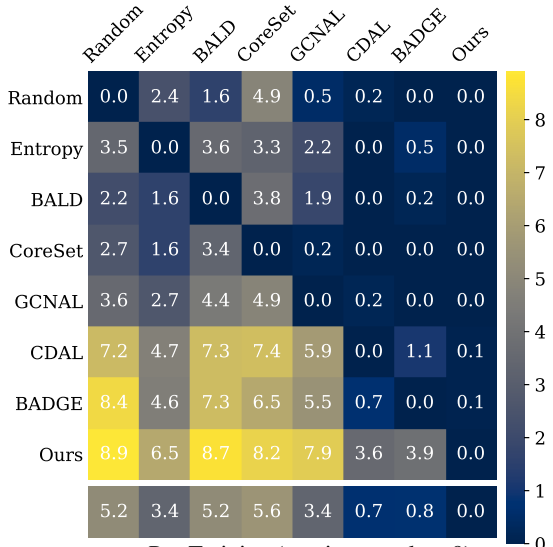


(e) ViT (maximum value: 3)

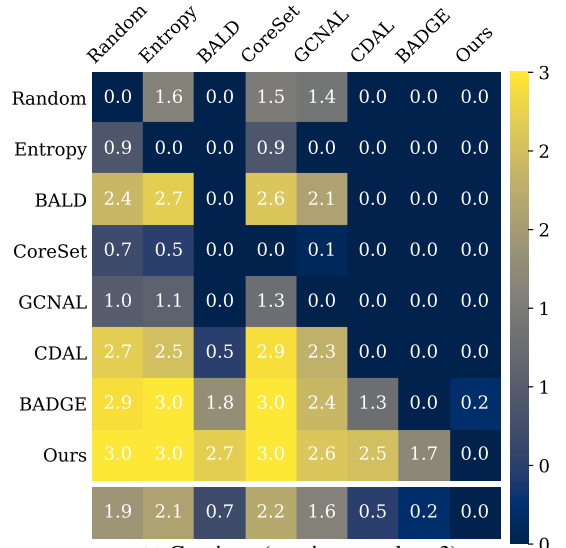
Figure 7. Pairwise comparison of different AL approaches based on different model architectures. The maximum value of each cell for each setting is also provided in the captions.



(a) Random (maximum value: 18)



(b) Pre-Training (maximum value: 9)



(c) Continue (maximum value: 3)

Figure 8. Pairwise comparison of different AL approaches based on different sizes of budget. The maximum value of each cell for each setting is also provided in the captions.

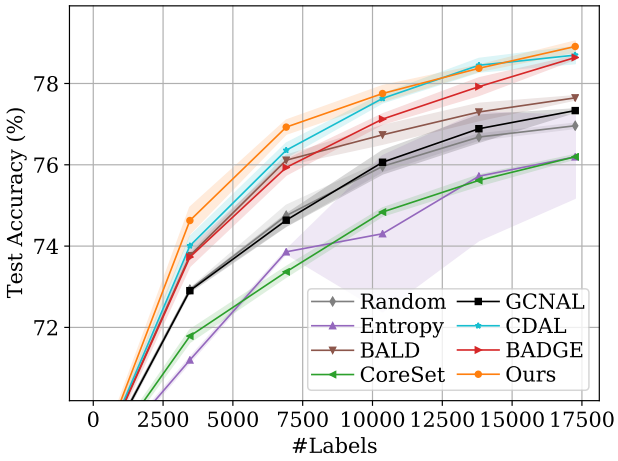


Figure 9. Small Budget, ViT-Base, DomainNet-Real

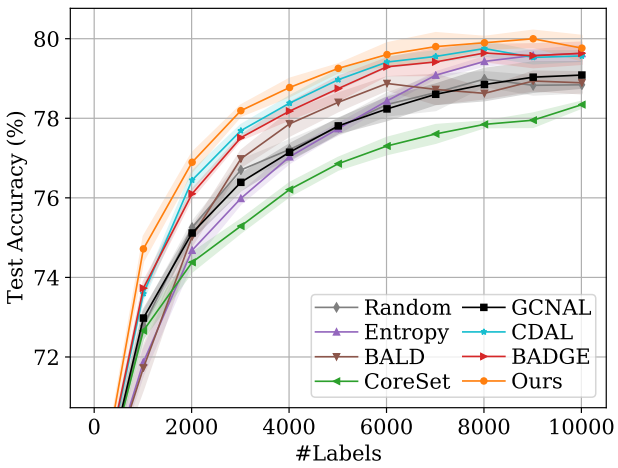


Figure 10. Small Budget, ViT-Small, Mini-ImageNet

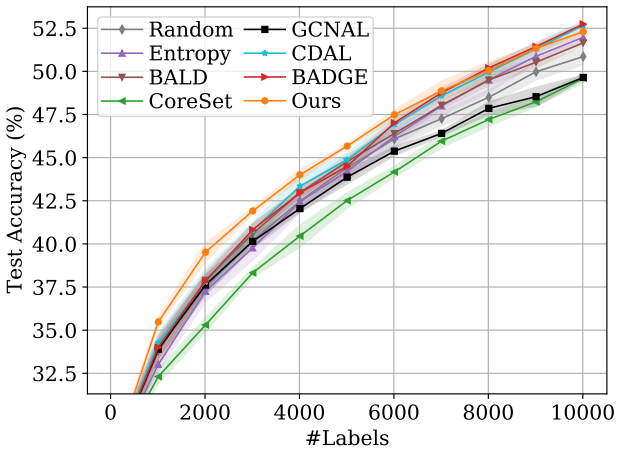


Figure 11. Small Budget, ViT-Small, CIFAR100

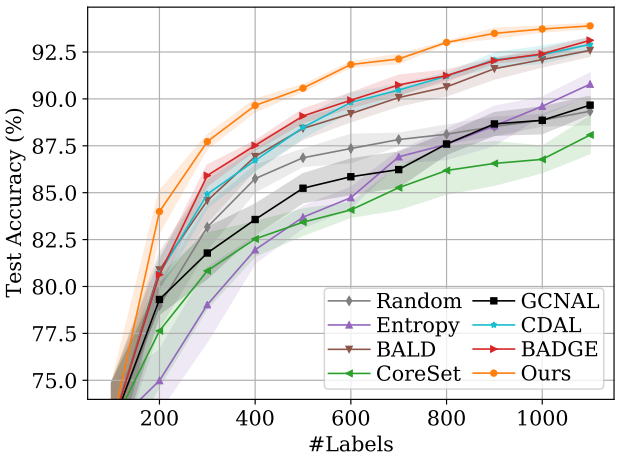


Figure 12. Small Budget, MLP, MNIST

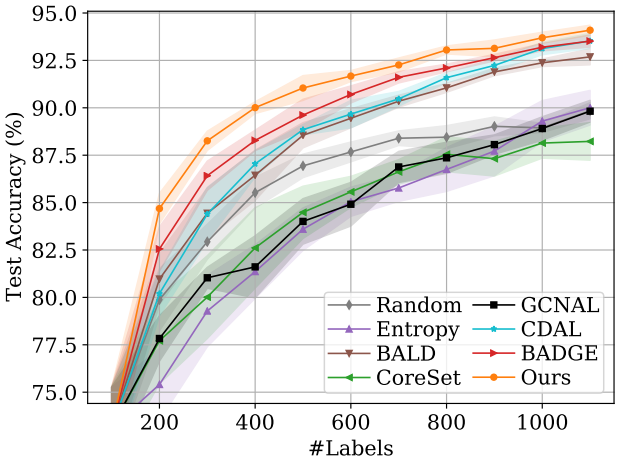


Figure 13. Small Budget, MLP, MNIST, Continue

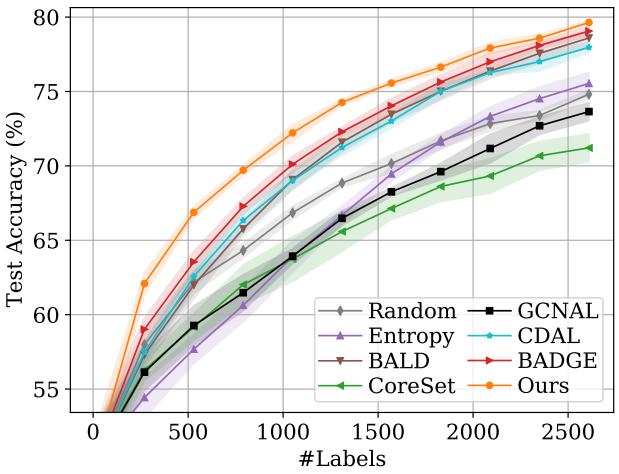


Figure 14. Small Budget, MLP, EMNIST

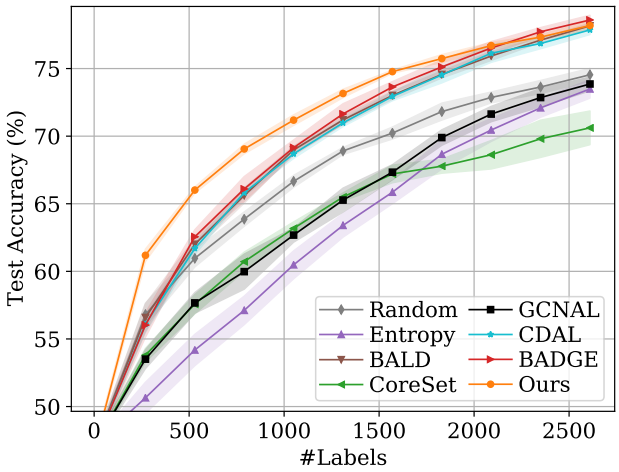


Figure 15. Small Budget, MLP, EMNIST, Continue

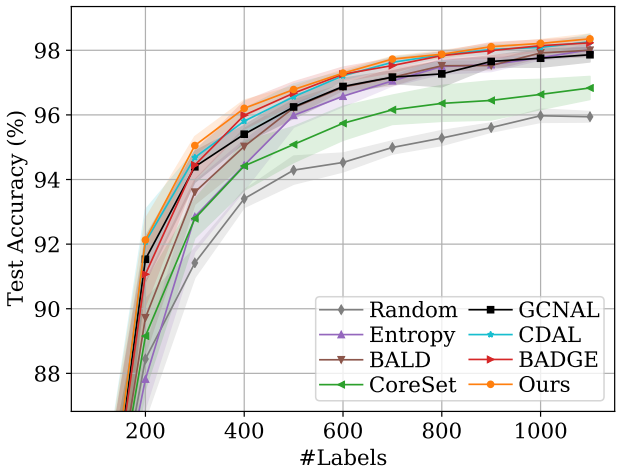


Figure 16. Small Budget, LeNet-5, MNIST

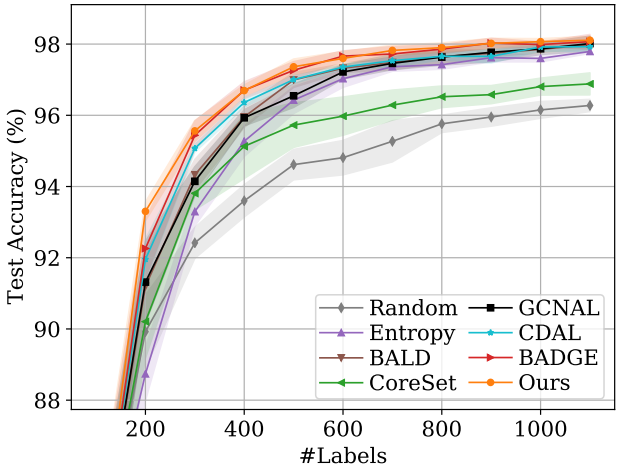


Figure 17. Small Budget, LeNet-5, MNIST, Continue

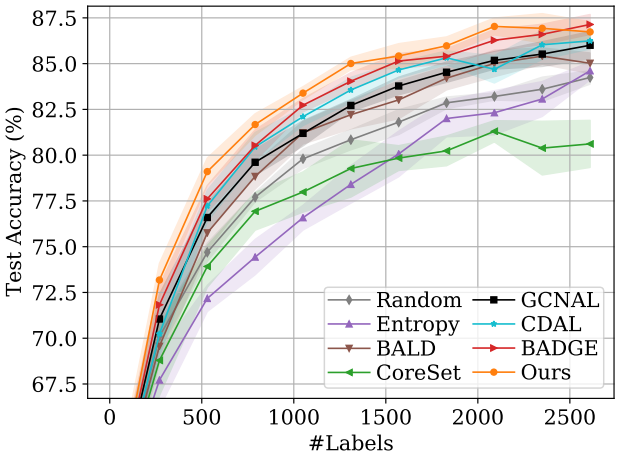


Figure 18. Small Budget, LeNet-5, EMNIST

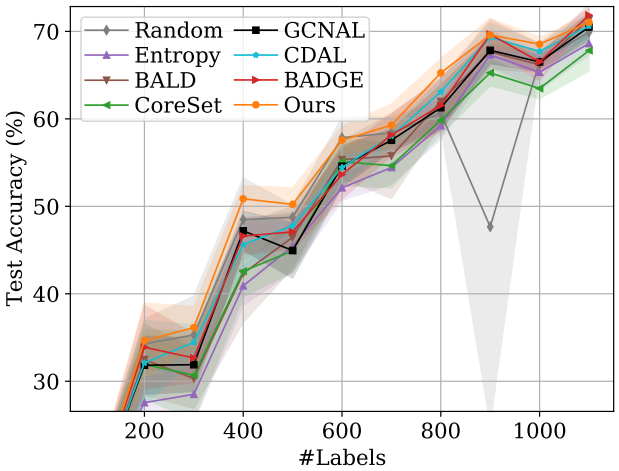


Figure 19. Small Budget-ResNet-18, SVHN

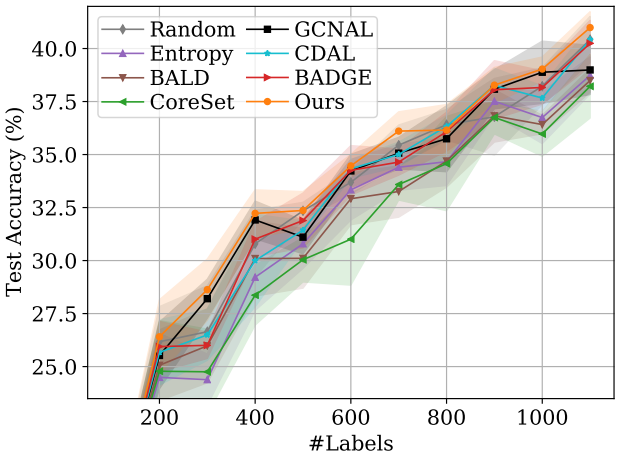


Figure 20. Small Budget, ResNet-18, CIFAR10

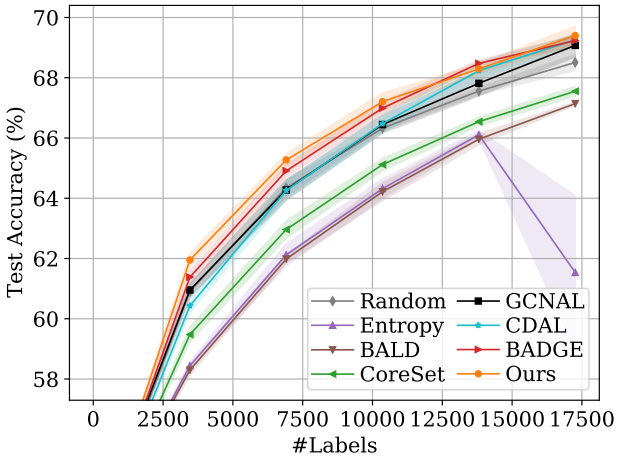


Figure 21. Small Budget, ResNet-18, DomainNet-Real

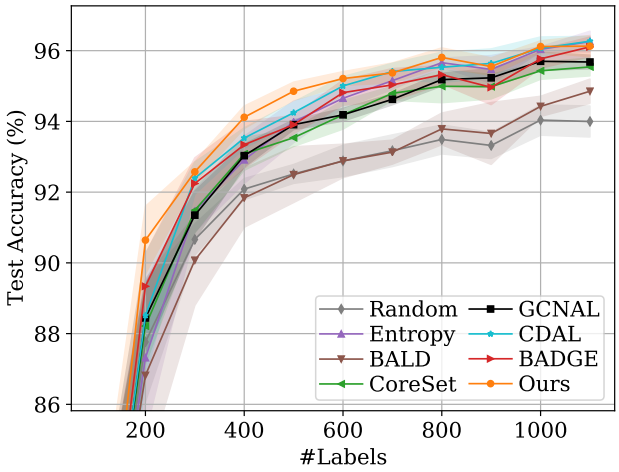


Figure 22. Small Budget, ResNet-18, DomainNet-Real-10

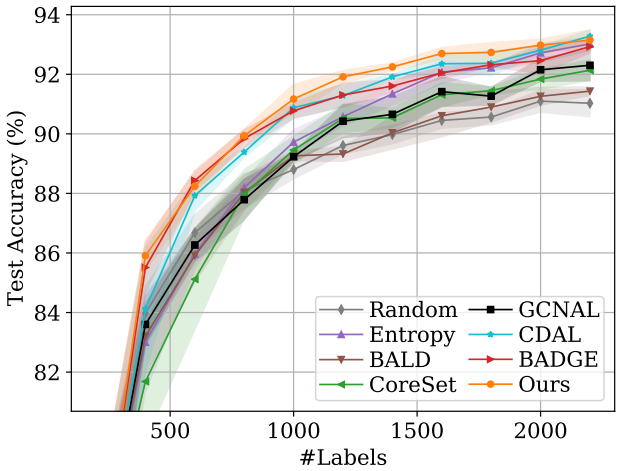


Figure 23. Small Budget, ResNet-18, DomainNet-Real-20

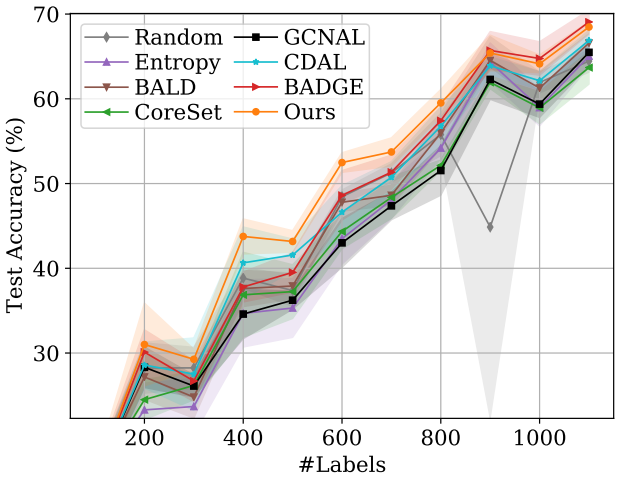


Figure 24. Small Budget, DenseNet-121, SVHN

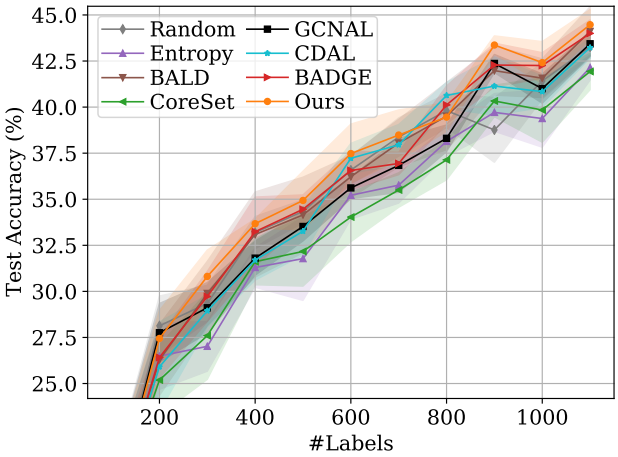


Figure 25. Small Budget, DenseNet-121, CIFAR10

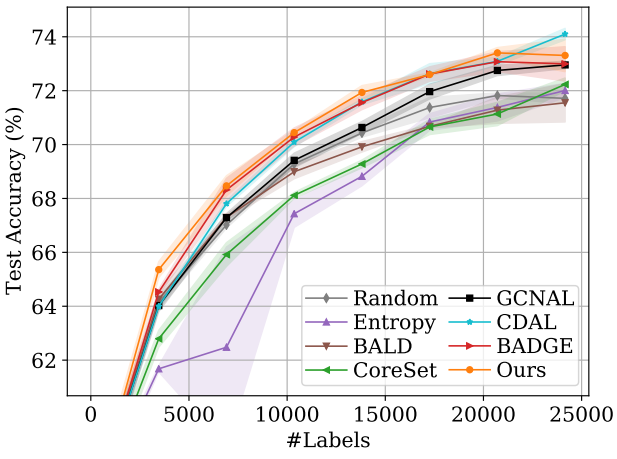


Figure 26. Small Budget, DenseNet-121, DomainNet-Real

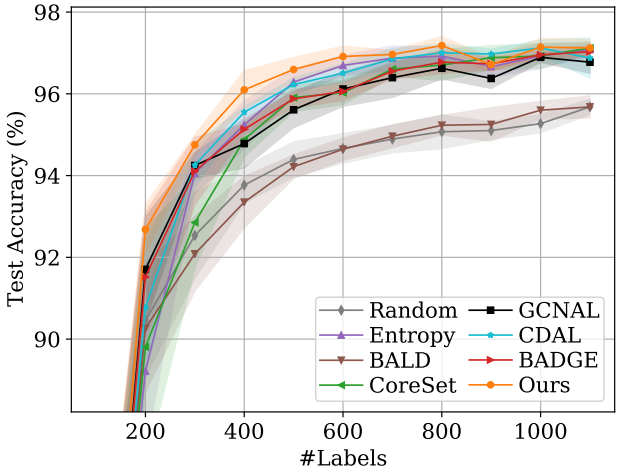


Figure 27. Small Budget, DenseNet-121, DomainNet-Real-10

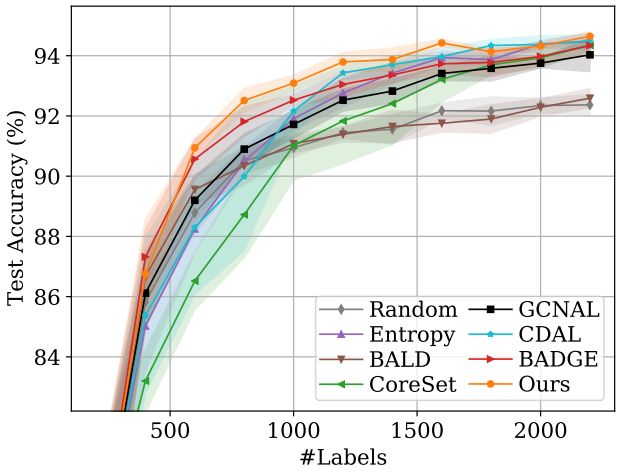


Figure 28. Small Budget, DenseNet-121, DomainNet-Real-20

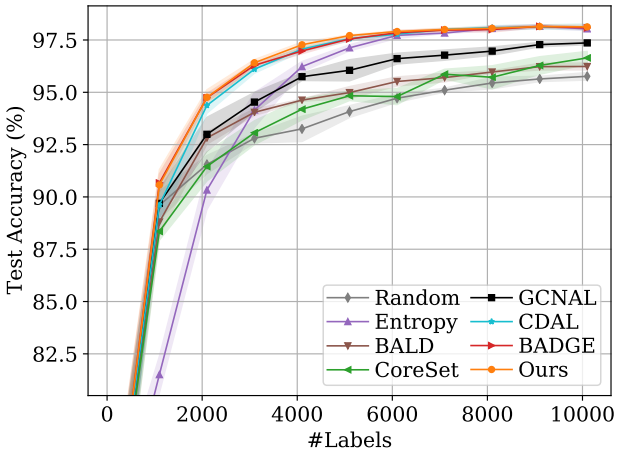


Figure 29. Large Budget, MLP, MNIST

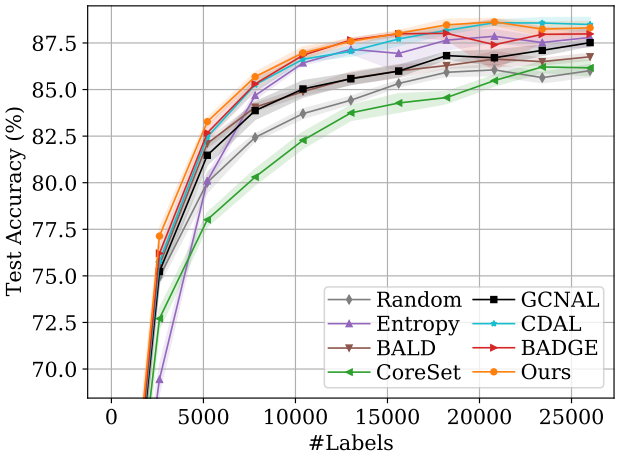


Figure 30. Large Budget,MLP, EMNIST

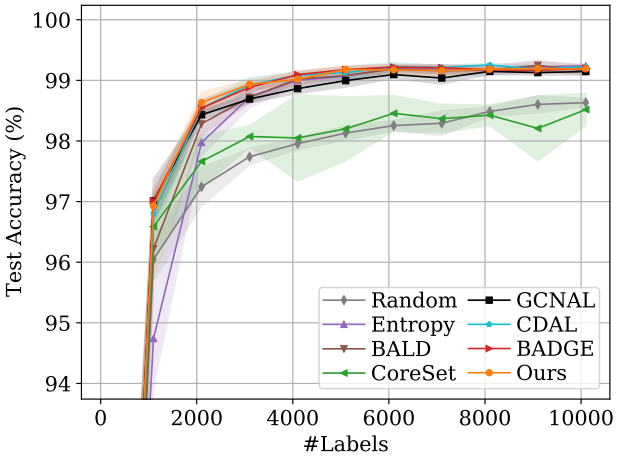


Figure 31. Large Budget, LeNet-5, MNIST

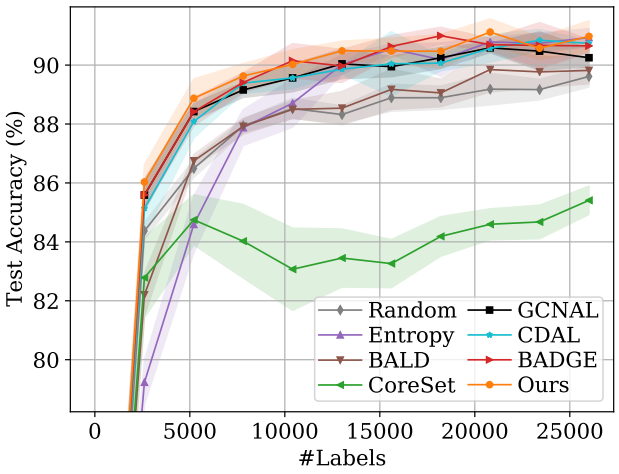


Figure 32. Large Budget, LeNet-5, EMNIST

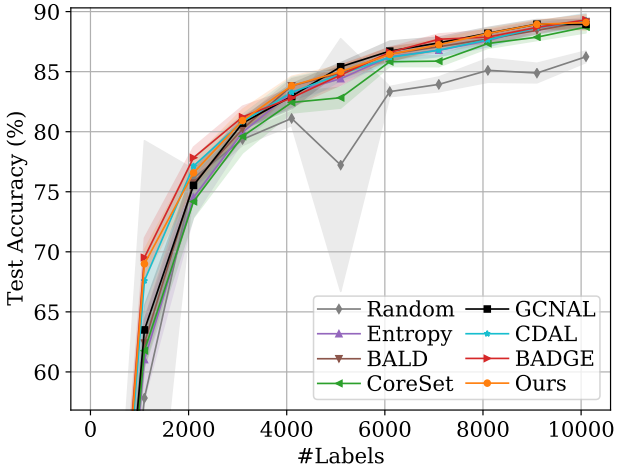


Figure 33. Large Budget, ResNet-18, SVHN

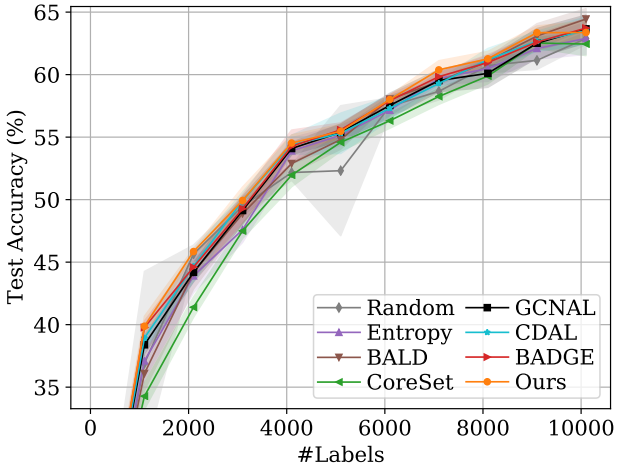


Figure 34. Large Budget, ResNet-18, CIFAR10

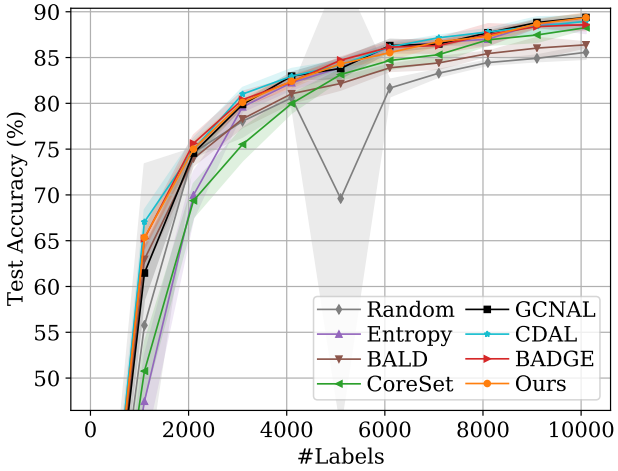


Figure 35. Large Budget, DenseNet-121, SVHN

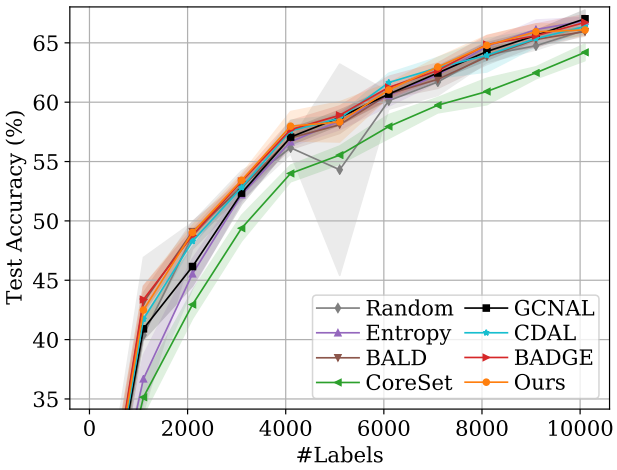


Figure 36. Large Budget, DenseNet-121, CIFAR10

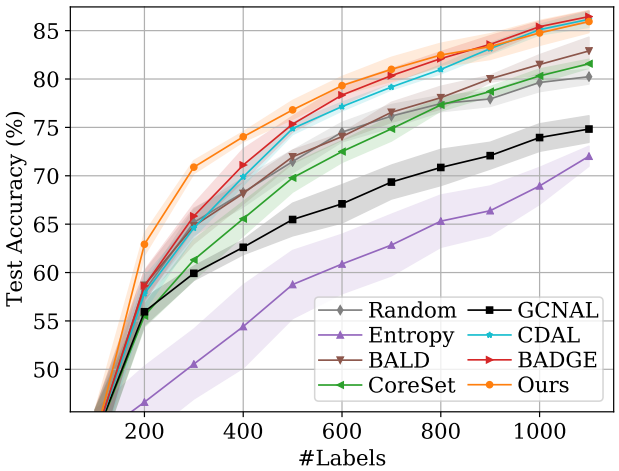


Figure 37. Small Budget, MLP, OpenML-6

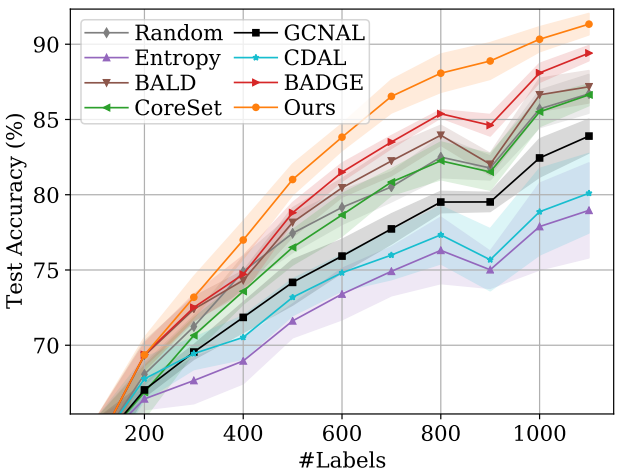


Figure 38. Small Budget, MLP, OpenML-155

References

[1] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[2] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020. 3, 4, 6

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. 1

[4] Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. Sequential graph convolutional network for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9583–9592, June 2021. 3

[5] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017. 3

[6] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. In *arXiv:1802.09841*, 2018. 4

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 3

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3

[9] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 3

[10] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Javen Shi, and Anton van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. In *Advances in Neural Information Processing Systems*, volume 33, 2020. 3

[11] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 3

[12] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 413–424, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 4

[13] Hugo Larochelle Sachin Ravi. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. 3

[14] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[15] D. Wang and Y. Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 112–119, 2014. 4