

1 Appendix A: Algorithm

Algorithm box 1

Algorithm 1: A single query round of Pretext-based Active Learning (PAL)

Result: Set of additional samples to be labeled \mathcal{D}_Q
Data: Labeled pool $\mathcal{D}_L := \{\mathbf{X}_L, Y_L\}$, unlabeled pool $\mathcal{D}_U := \{\mathbf{X}_U\}$, query size N
Set: Num. epochs E_Q and E_S , num. sub-queries K , task network f_θ , Scoring network with 2 heads g_ϕ and h_ψ
Training task and scoring networks
for $t \in \{1, \dots, E_Q\}$ **do**
 for $\{\mathbf{x}_l, y_l\} \in \mathcal{D}_L$ **do**
 $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(f_\theta(\mathbf{x}_l), y_l)$ # Task network, \mathcal{L} represents cross-entropy loss
 $\psi \leftarrow \psi - \eta \nabla_\psi \mathcal{L}(h_\psi(\mathbf{x}_l), y_l)$ # Scoring network
 for $i \in \{0, 1, 2, 3\}$ **do**
 $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}(g_\phi(\text{rot}_{90i}(\mathbf{x}_l)), i)$ # Scoring network
for $\mathbf{x}_u \in \mathcal{D}_U$ **do**
 Use g, h to compute and save $S_S(\mathbf{x}_u), S_C(\mathbf{x}_u)$
Diversity-based sub-query sampling
Initialize: $\mathcal{D}_Q = \emptyset; \phi' = \phi$
for $k \in \{1, \dots, K\}$ **do**
 for $n \in \{1, \dots, \frac{N}{K}\}$ **do**
 if $k == 1$ **then**
 $\mathbf{x}_q \leftarrow \arg \min_{\mathbf{x}_u \in \mathcal{D}_U} S_S(\mathbf{x}_u) + \lambda_1 S_C(\mathbf{x}_u)$
 else
 $\mathbf{x}_q \leftarrow \arg \min_{\mathbf{x}_u \in \mathcal{D}_U} S_S(\mathbf{x}_u) + \lambda_1 S_C(\mathbf{x}_u) + \lambda_2 S_D(\mathbf{x}_u)$
 $\mathcal{D}_Q \leftarrow \mathcal{D}_Q \cup \{\mathbf{x}_q\}$
 $\mathcal{D}_U \leftarrow \mathcal{D}_U - \{\mathbf{x}_q\}$
 for $t \in \{1, \dots, E_S\}$ **do**
 for $\mathbf{x}_q \in \mathcal{D}_Q$ **do**
 for $i \in \{0, 1, 2, 3\}$ **do**
 $\phi' \leftarrow \phi' - \eta \nabla_{\phi'} \mathcal{L}(g_{\phi'}(\text{rot}_{90i}(\mathbf{x}_q)), i)$
 for $\mathbf{x}_u \in \mathcal{D}_U$ **do**
 Use $g_{\phi'}$ to compute and save $S_D(\mathbf{x}_u)$
 Get oracle to label \mathcal{D}_Q and update \mathcal{D}_L

2 Appendix B: Sampling New Classes

In Section 4.3 we evaluated PAL in the setting when new classes are introduced on-the-fly during the active learning based sampling. We started of with a biased initial pool where some of the classes are removed in the initial labeled data pool. Here we show the results in the same setting on a segmentation task on Cityscapes dataset. Out of the 19 classes in Cityscapes dataset, we removed the annotations of the bus and the train classes from the initial labeled data pool. For clarity, all further query rounds have access to all the class annotations.

Figure 1 compares the mean Intersection over Union (mIoU) of PAL when applied on the task of semantic segmentation with random sampling in this setting. PAL is able to improve its mIoU quickly because it is able to sample more images with higher area of missing annotations compared to random sampling, as shown in Fig 4 in the main paper.

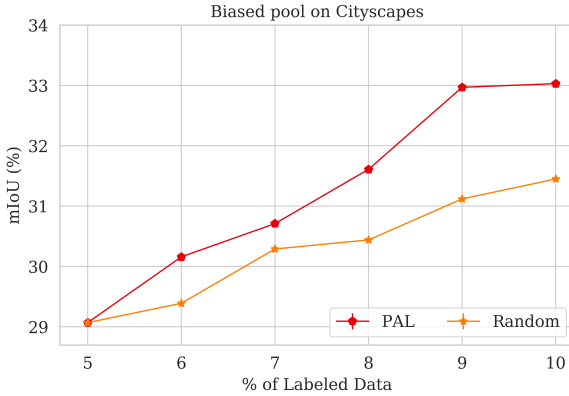


Figure 1: PAL performance with biased initial pool of only seventeen out of nineteen classes: The mean Intersection over Union (mIoU) of PAL trained with biased initial pool improves quickly as compared to that of random sampling

3 Appendix C: Hyperparameters

Using validation, the relative importance hyperparameters (λ_1, λ_2) in Equation 4 of the main paper were selected from $\{0.5, 1.0\}$. Learning rates were in the range $[10^{-1}, 10^{-4}]$. Optimizers were selected from $\{\text{ADAM}, \text{SGD}\}$. The hardware included an NVIDIA GeForce GTX 1080 GPU running CUDA 10.2 and cuDNN 7.6 using PyTorch.

We share the hyperparameters used for training the task and the scoring models for our different experiments in Table 1. All hyperparameters were obtained through a grid search. The hyperparameters λ_1 and λ_2 of Equation 4 in Section 3.3 were selected from $\{0.5, 1.0\}$. Learning rates α_T for the task model and α_S for the scoring model were selected from the range $[10^{-1}, 10^{-4}]$. Optimizers were selected from $\{\text{Adam}, \text{SGD}\}$.

Dataset	α_T	α_S	task & scoring model epochs	batch size	λ_1	λ_2	optimizer
CIFAR-10	0.01	0.01	100	64	1	1	SGD
SVHN	0.01	0.01	100	64	1	1	SGD
Cityscapes	0.01	0.01	50	8	0.5	0	SGD
Caltech-101	0.01	0.01	100	32	1	1	SGD

Table 1: Parameters for experiments on various datasets

4 Appendix D: The Hybrid Score

Proposition 1: *Negative of KL-divergence of a class PMF from a uniform distribution can overshadow the confusion score from S_S , but entropy cannot.*

Proof: Consider a binary classification problem for analysis, with p as the predicted probability score by the task network for the correct class. When the unlabeled sample is almost correctly classified with $p \rightarrow 1$, we get the following for the hybrid confusion score:

$$\begin{aligned} S(\mathbf{x}_u) &= S_S(\mathbf{x}_u) + \lambda S_C(\mathbf{x}_u) \\ \lim_{p \rightarrow 1} S &= \lim_{p \rightarrow 1} \left(S_S - \frac{\lambda}{2} \log \left(\frac{1}{2p} \right) - \frac{\lambda}{2} \log \left(\frac{1}{2(1-p)} \right) \right) \\ &= S_S - \frac{\lambda}{2} \log \left(\frac{1}{2} \right) - \frac{\lambda}{2} \lim_{p \rightarrow 1} \log \left(\frac{1}{2(1-p)} \right) \\ &= -\infty. \end{aligned}$$

On the other hand, if S_C is replaced by the entropy of the PMF h_ψ , then the hybrid score S_E would be finite because:

$$\begin{aligned} \lim_{p \rightarrow 1} S_E &= \lim_{p \rightarrow 1} (S_S - \lambda p \log(p) - \lambda(1-p) \log(1-p)) \\ &= S_S - 0 - \lambda \lim_{p \rightarrow 1} (1-p) \log(1-p) \\ &= S_S - \lambda \lim_{p \rightarrow 1} \frac{\log(1-p)}{\frac{1}{(1-p)}} = S_S, \end{aligned}$$

using L'Hôpital's rule to equate the second term to 0. □

An added advantage of using a multi-task setting for the scoring network is getting better ordinal estimates of a true latent score due to an ensemble-like effect, as long as the correlations between the two components of the score and their correlation with the underlying score are positive. This can follow from the following proposition:

Proposition 2: *There exists a trade-off parameter that maximizes the correlation between the true underlying score and the hybrid score, which is greater than or equal to the correlation of the true score with either of the components, as long as all correlations between the scores are positive.*

Proof: Note that the requirement of a positive correlation is only a weak one for any reasonably trained networks g_ϕ and h_ψ , as we empirically show in Table 1 in the main paper. Now, without loss of generality, let us assume that some monotonic transformations of the true underlying score, the self-supervision score, and the classification score give standardized random variables u , v , and w respectively, such that their means $\mu_u = \mu_v = \mu_w = 0$, and their variances $\sigma_u^2 = \sigma_v^2 = \sigma_w^2 = 1$. Further, we assume that the covariances σ_{uv} , σ_{uw} , and σ_{vw} are positive. Let an analog of the hybrid score s be a positive combination of the two given by $s = \alpha v + \sqrt{1 - \alpha^2} w$, where $\alpha \in [0, 1]$ has a monotonic relation with the $\lambda \geq 0$ in the hybrid score, and the variance $\sigma_s^2 = 1$. Then, the correlation between u and s , which is the same as the cosine between them, is $\mathbf{E}[u.s] = \alpha \sigma_{uv} + \sqrt{1 - \alpha^2} \sigma_{uw}$. If we maximize this correlation by setting its derivative with respect to α to zero, we get:

$$\begin{aligned}
\frac{d\mathbf{E}[u.s]}{d\alpha} &= 0 \\
\Rightarrow \frac{d}{d\alpha} \left(\sigma_{uv}\alpha + \sigma_{uw}\sqrt{1-\alpha^2} \right) &= 0 \\
\Rightarrow \sigma_{uv} + \frac{-\alpha}{\sqrt{1-\alpha^2}} \sigma_{uw} &= 0 \\
\Rightarrow \sigma_{uv}^2(1-\alpha^2) &= \sigma_{uw}^2 \alpha^2 \\
\Rightarrow \alpha &= \pm \frac{\sigma_{uv}}{\sqrt{\sigma_{uv}^2 + \sigma_{uw}^2}}
\end{aligned}$$

Clearly, a maxima for $\mathbf{E}[u.s]$ exists, because its second derivative is negative for $\alpha^* = \frac{\sigma_{uv}}{\sqrt{\sigma_{uv}^2 + \sigma_{uw}^2}}$ when the covariances are positive, and $\alpha^* \in (0, 1)$. \square

5 Appendix E: Robustness to scoring network architecture

We observed that changing the backbone architecture of the scoring network (for ex. from ResNet-18 to VGG-16) does not cause a significant change in the performance of PAL, as shown in Figure 2.

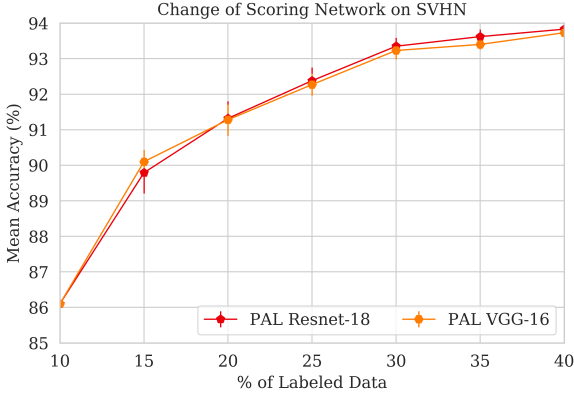


Figure 2: A change in the backbone architecture of the scoring network has no significant effect on the performance of PAL on SVHN