DEAL: Difficulty-aware Active Learning for Semantic Segmentation

Shuai Xie[1], Zunlei Feng[1], Ying Chen[1], Songtao Sun[1], Chao Ma[1], and Mingli Song[1]

Zhejiang University, Hangzhou, China
{shuaixie, zunleifeng, lynesychen, songtaosun, chaoma,
brooksong}@zju.edu.cn

Abstract. Active learning aims to address the paucity of labeled data by finding the most informative samples. However, when applying to semantic segmentation, existing methods ignore the segmentation difficulty of different semantic areas, which leads to poor performance on those hard semantic areas such as tiny or slender objects. To deal with this problem, we propose a semantic Difficulty-awarE Active Learning (DEAL) network composed of two branches: the common segmentation branch and the semantic difficulty branch. For the latter branch, with the supervision of segmentation error between the segmentation result and GT, a pixel-wise probability attention module is introduced to learn the semantic difficulty scores for different semantic areas. Finally, two acquisition functions are devised to select the most valuable samples with semantic difficulty. Competitive results on semantic segmentation benchmarks demonstrate that DEAL achieves state-of-the-art active learning performance and improves the performance of the hard semantic areas in particular.

Keywords: Active learning · Semantic segmentation · Difficulty-aware

1 Introduction

Semantic segmentation is a fundamental task for various applications such as autonomous driving [1, 2], biomedical image analysis [3–5], remote sensing [6] and robot manipulation [7]. Recently, data-driven methods have achieved great success with large-scale datasets [8, 9]. However, tremendous annotation cost has become an obstacle for these methods to be widely applied in practical scenarios. Active Learning (AL) can be the right solution by finding the most informative samples. Annotating those selected samples can support sufficient supervision information and reduce the requirement of labeled samples dramatically.

Previous methods can be mainly categorized into two families: uncertainty-based [10–13] and representation-based [14–16]. However, many works [10, 12, 14, 16] are only evaluated on image classification benchmarks. There has been considerably less work specially designed for semantic segmentation. Traditional uncertainty-based methods like Entropy [17] and Query-By-Committee (QBC)

(a) Image   (b) Semantic difficulty map

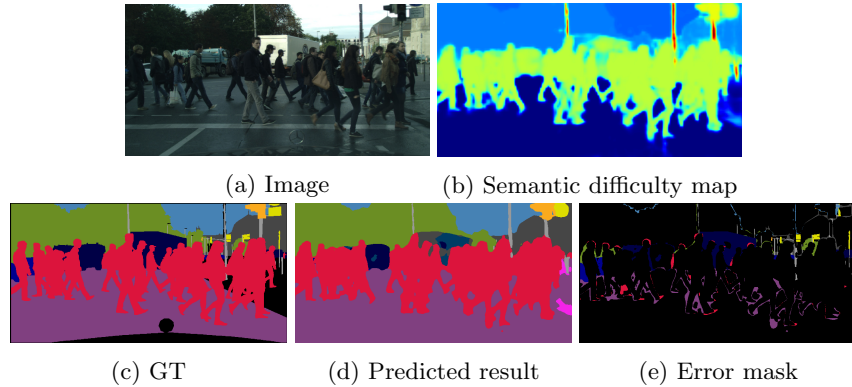(c) GT   (d) Predicted result   (e) Error mask

Fig. 1: (a) Image from Cityscapes [8]. (b) Semantic difficulty map. The colder color represents the easier semantic areas such as road, sky, and buildings. The warmer color represents the harder semantic areas such as poles and traffic signs. (c) GT. (d) Predicted result. (e) Error mask generated with (c) and (d) according to Eq. (1). It's a binary image, coloring for better visualization.

[18] have demonstrated their effectiveness in semantic segmentation [19, 20]. However, all of them are solely based on the uncertainty reflected on each pixel, without considering the semantic difficulty and the actual labeling scenarios.

In this paper, we propose a semantic Difficulty-awarE Active Learning (DEAL) method taking the semantic difficulty into consideration. Due to the class imbalance and shape disparity, a noticeable semantic difficulty difference exists among the different semantic areas in an image. To capture this difference, we adopt a two-branch network composed of a semantic segmentation branch and a semantic difficulty branch. For the former branch, we adopt the common segmentation network. For the latter branch, we leverage the wrong predicted result as the supervision, which is termed as the *error mask*. It's a binary image where the right and wrong positions have a value 0 and 1, respectively. As illustrated in Fig. 1(e), we color these wrong positions for better visualization. Then, a pixel-wise probability attention module is introduced to aggregate similar pixels into areas and learn the proportion of misclassified pixels as the difficulty score for each area. Finally, we can obtain the semantic difficulty map in Fig. 1(b).

Then two acquisition functions are devised based on the map. One is Difficulty-aware uncertainty Score (DS) combining the uncertainty and difficulty. The other is Difficulty-aware semantic Entropy (DE) solely based on the difficulty. Experiments show that the learned difficulty scores have a strong connection with the standard evaluation metric IoU. And DEAL can effectively improve the overall AL performance and the IoU of the hard semantic classes in particular.

In summary, our major contributions are as follows: 1) Proposing a new AL framework incorporating the semantic difficulty to select the most informative samples for semantic segmentation. 2) Utilizing *error mask* to learn the semantic difficulty. 3) Competitive results on CamVid [21] and Cityscapes [8].

2 Related Work

AL for semantic segmentation The core of AL is measuring the informativeness of the unlabelled samples. Modern AL methods can be mainly divided into two groups: uncertainty-based [10–13] and representation-based [14–16]. The latter views the AL process as an approximation of the entire data distribution and query samples to increase the data diversity, such as Core-set [14] and VAAL [15], which can be directly used in semantic segmentation. There are also some methods specially designed for semantic segmentation, which can also be divided into two groups: image-level [4, 11, 19, 22] and region-level [23, 24, 20].

Image-level methods use the complete image as the sampling unit. [4] propose suggestive annotation (SA) and train a group of models on various labeled sets obtained with bootstrap sampling and select samples with the highest variance. [11] employ MC dropout to measure uncertainty for melanoma segmentation. [19] adopt QBC strategy and propose a cost-sensitive active learning method for intracranial hemorrhage detection. [22] build a batch mode multi-clue method, incorporating edge information with QBC strategy and graph-based representativeness. All of them are based on a group of models and time-consuming when querying a large unlabeled data pool.

Region-level methods only sample the informative regions from images. [23] combines the MC dropout uncertainty with an effort estimation regressed from the annotation click patterns, which is hard to access for many datasets. [24] propose ViewAL and use the inconsistencies in model predictions across viewpoints to measure the uncertainty of super-pixels, which is specially designed for RGB-D data. [20] model a deep Q-network-based query network as a reinforcement learning agent, trying to learn sampling strategies based on prior AL experience. In this work, we incorporate the semantic difficulty to measure the informativeness and select samples at the image level. Region-level method will be studied in the future.

Self-attention mechanism for semantic segmentation The self-attention mechanism is first proposed by [25] in the machine translation task. Now, it has been widely used in many tasks [25–28] owing to its intuition, versatility and interpretability [29]. The ability to capture the long-range dependencies inspires many semantic segmentation works designing their attention modules. [30] use a point-wise spatial attention module to aggregate context information in a self-adaptive manner. [31] introduce an object context pooling scheme to better aggregate similar pixels belonging to the same object category. [32] replace the non-local operation [33] into two consecutive criss-cross operations and gather long-range contextual information in the horizontal and vertical directions. [34] design two types of attention modules to exploit the dependencies between pixels and channel maps. Our method also uses the pixel-wise positional attention mechanism in [34] to aggregate similar pixels.

3   Method

Before introducing our method, we first give the definition of the AL problem. Let (x^a, y^a) be an annotated sample from the initial annotated dataset $\mathcal{D}^a$ and x^u be an unlabeled sample from a much larger unlabeled data pool $\mathcal{D}^u$. AL aims to iteratively query a subset $\mathcal{D}^s$ containing the most informative m samples $\{x_1^u, x_2^u, ..., x_m^u\}$ from $\mathcal{D}^u$, where m is a fixed budget.

In what follows, we first give an overview of our difficulty-aware active learning framework, then detail the probability attention module and loss functions, finally define two acquisition functions.

3.1   Difficulty-aware Active Learning

To learn the semantic difficulty, we exploit the *error mask* generated from the segmentation result. Our intuition is that these wrong predictions are what our model "feels" difficult to segment, which may have a relation with the semantic difficulty. Thus, we build a two-branch network generating semantic segmentation result and semantic difficulty map in a multi-task manner, best viewed in Fig. 2. The first branch is a common segmentation network, which can be used to generate the *error mask*. The second branch is devised to learn the semantic difficulty map with the guidance of *error mask*. This two-branch architecture is inspired by [13]. In their work, a loss prediction module is attached to the task model to predict a reliable loss for x^u and samples with the highest losses are selected. While in our task, we dig deeper into the scene and analyze the semantic difficulty of each area.

As illustrated in Fig. 2, the first branch can be any semantic segmentation network. Assume S^* is the output of softmax layer and S^p is the prediction result after the argmax operation. With the segmentation result S^p and GT S^g, the *error mask* M^e can be computed by:

$$M_k^e = \begin{cases} 1 & \text{if } S_k^p \neq S_k^g, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where S_k^p and S_k^g denote the kth pixel value of the segmentation result and GT, M_k^e is the kth pixel value of the *error mask*.

The second branch is composed of two parts: a probability attention module and a simple 1×1 convolution layer used for binary classification. The softmax output of the first branch S^* is directly used as the input of the second branch, which are C-channel probability maps and C is the number of classes. We denote it as $P \in \mathcal{R}^{C \times H \times W}$ and P_k is the probability vector for the kth pixel. Using probability maps is naive but accompanied with two advantages. First, pixels with similar difficulty tend to have similar P_k. Second, pixels of the same semantic tend to have similar P_k. Combined with a pixel-wise attention module, we can easily aggregate these similar pixels and learn similar difficulty scores for them. In our learning schema, the performance of the second branch depends much on the output of the first branch. However, there is no much difference if we
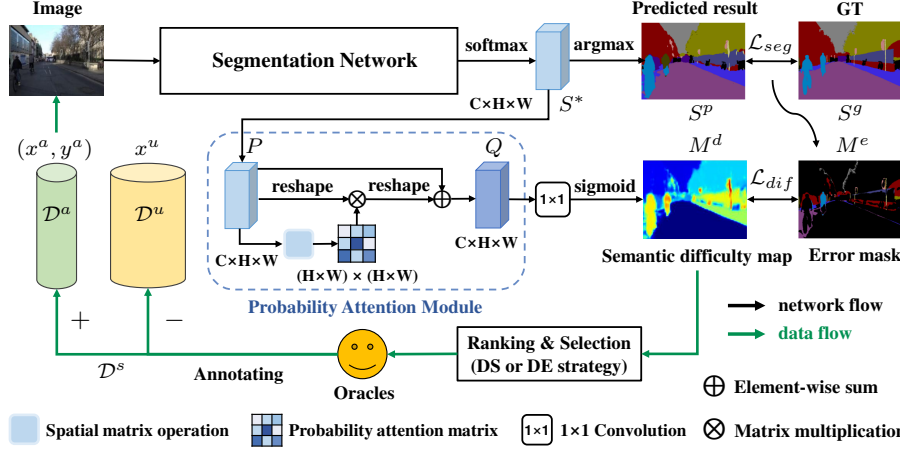
Fig. 2: Overview of our difficulty-aware active learning framework for semantic segmentation. The first branch is a common semantic segmentation network. The second branch is composed of a praobability attention module and a 1×1 convolution. $\mathcal{D}^a$ and $\mathcal{D}^u$ are the annotated and unlabeled data, $\mathcal{D}^s$ is a subset selected from $\mathcal{D}^u$. P and Q are the probability maps before and after attention. $\mathcal{L}_{seg}$ and $\mathcal{L}_{dif}$ are the loss functions described in Eq. 3 and Eq. 4. DS and DE strategies are detailed in Sec. 3.4.

branch these two tasks earlier and learn the independent features. We validate this opinion in Sec. 5.2.

The semantic difficulty learning process can be imagined into two steps. Firstly, we learn a binary segmentation network with the supervision of *error mask* M^e. Each pixel will learn a semantic difficulty score. Secondly, similar pixels are aggregated into an area so that this score can be spread among them. Finally, we can obtain the semantic difficulty map M^d.

3.2 Probability Attention Module

In this section, we detail the probability attention module (PAM) in our task. Inspired by [34], we use this module to aggregate pixels with similar softmax probability. Given the probability maps $P \in \mathcal{R}^{C \times H \times W}$, we first reshape it to $P \in \mathcal{R}^{C \times K}$, where $K = H \times W$. Then the probability attention matrix $A \in \mathcal{R}^{K \times K}$ can be computed with $P^T P$ and a softmax operation as below:

$$A_{ji} = \frac{exp(P_i^T \cdot P_j)}{\sum_{i=1}^{K} exp(P_i^T \cdot P_j)},$$

$$Q_j = \gamma \sum_{i=1}^{K} (A_{ji} P_i) + P_j, \tag{2}$$

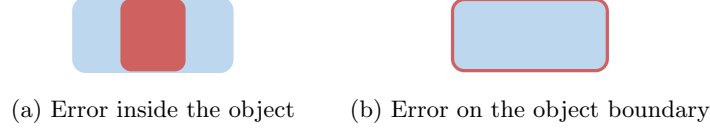(a) Error inside the object   (b) Error on the object boundary

Fig. 3: Two typical errors in semantic segmentation. The right and wrong areas are in blue and red, best viewed in color.

where A_{ji} is the ith pixel's impact on j^{th} pixel, P_j is the original probability vector of the jth pixel and Q_j is the one after attention, γ is a learnable weight factor. Finally, we can get the probability maps $Q \in \mathcal{R}^{C \times H \times W}$ after attention.

Let's take the segmentation result of the two bicyclists in Fig. 2 to explain the role of PAM as it reflects two typical errors in semantic segmentation: (1) error inside the object (the smaller one b_1); (2) error on the object boundary (the larger one b_2), as shown in Fig. 3. Assume our attention module can aggregate pixels from the same object together, the right part of the object learns 0 while the wrong part learns 1. Since b_1 has a larger part of wrong areas, it tends to learn larger difficulty scores than b_2. Similar to objects, pixels from the same semantic, such as road, sky and buildings can also learn similar difficulty scores. Ablation study in Sec. 5.1 also demonstrates that PAM can learn more smooth difficulty scores for various semantic areas.

Some traditional methods also employ the softmax probabilities to measure the uncertainty, such as least confidence (LC) [35], margin sampling (MS) [36] and Entropy [17]. The most significant difference between our method and these methods is that we consider difficulty at the semantic level with an attention module, rather than measuring the uncertainty of each pixel alone. QBC [18] can use a group of models, but it still stays at the pixel level. To clearly see the difference, we compare our semantic difficulty map with the uncertainty maps of these methods in Fig. 4. The first row are uncertainty maps generated with

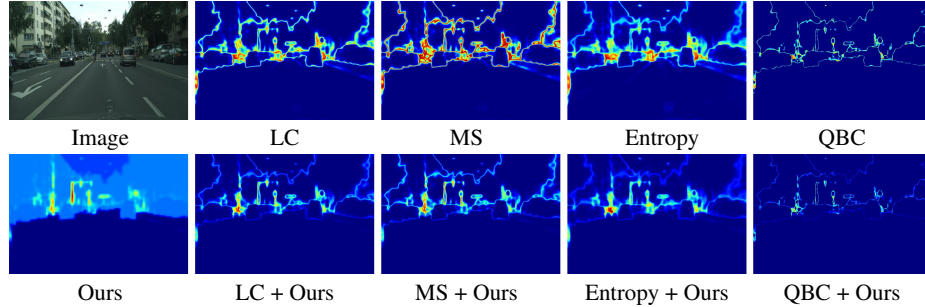| Image | LC | MS | Entropy | QBC |
| Ours | LC + Ours | MS + Ours | Entropy + Ours | QBC + Ours |

Fig. 4: First row: Image from Cityscapes [8] and traditional uncertainty maps. Second row: Our semantic difficulty map and difficulty-aware uncertainty maps. The warmer color means the higher uncertainty.

these methods, which are loyal to the uncertainty of each pixel. For example, some pixels belonging to *sky* can have the same uncertainty with *traffic light* and *traffic sign*. Supposing an image has many pixels with high uncertainty belonging to the easier classes, it will be selected by these methods. While our semantic difficulty map (first in the second row) can serve as the difficulty attention and distinguish more valuable pixels. As illustrated in the second row, combined with our difficulty map, the uncertainty of the easier semantic areas like *sky* is suppressed while the harder semantic areas like *traffic sign* is reserved.

3.3   Loss Functions

Loss of Semantic Segmentation To make an equitable comparison with other methods, we use the standard cross-entropy loss for the semantic segmentation branch, which is defined as:

$$\mathcal{L}_{seg}(S^*, S^g) = \frac{1}{K} \sum_{k=1}^{K} \ell(S_k^*, S_k^g) + R(\theta), \tag{3}$$

where S_k^* and S_k^g are the segmentation output and ground truth for pixel k, $\ell(\cdot)$ is the cross-entropy loss, K is the total pixel number, and R is the L2-norm regularization term.

Loss of Semantic Difficulty For the semantic difficulty branch, we use an inverted weighted binary cross-entropy loss defined below, as there is a considerable imbalance between the right and wrong areas of *error mask*.

$$\mathcal{L}_{dif}(M^d, M^e) = -\frac{1}{K} \sum_{k=1}^{K} \lambda_1 M_k^e log(M_k^d) + \lambda_2(1 - M_k^e)log(1 - M_k^d),$$
$$\lambda_1 = \frac{\sum_{k}^{K} \mathbf{1}(M_k^e = 0)}{K}, \lambda_2 = 1 - \lambda_1, \tag{4}$$

where M_k^d and M_k^e are the difficulty prediction and *error mask* ground truth for pixel k, $\mathbf{1}(\cdot)$ is the indicator function, and λ_1 and λ_2 are dynamic weight factors.

Final Loss Our final training objective is a combination of Eq. 3 and Eq. 4, which is computed as:

$$\mathcal{L} = \mathcal{L}_{seg} + \alpha \mathcal{L}_{dif}, \tag{5}$$

where α is a weight factor and set to 1 in the experiments.

3.4   Acquisition Functions

Samples from $\mathcal{D}^u$ are usually ranked with a scalar score in AL. However, semantic segmentation is a dense-classification task, many methods output a score for each pixel on the image, including our semantic difficulty map. Thus, it's quite significant to design a proper acquisition function. Below are two functions we have designed.

(a) Image   (b) GT   (c) Semantic difficulty map

(d) Level 1   (e) Level 2   (f) Level 3   (g) Level 4   (h) Level 5   (i) Level 6
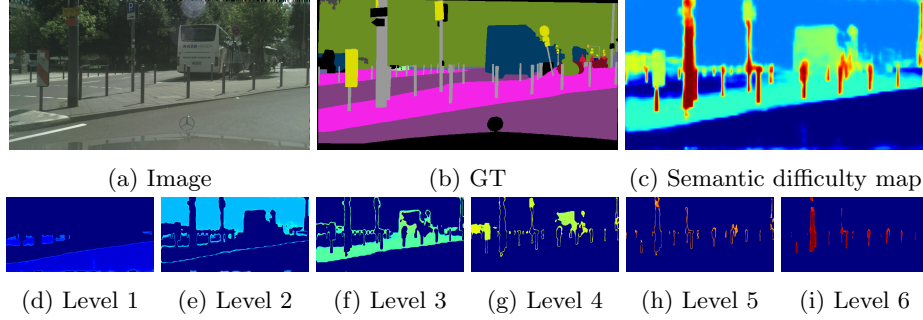
Fig. 5: Quantified difficulty levels. Semantics in (a) are quantified into 6 difficulty levels based on (c). (d) Level 1: road. (e) Level 2: vegetation. (f) Level 3: sidewalk and a part of bus. (g) Level 4: traffic light and the rear window of bus. (h) Level 5: boundaries of poles. (i) Level 6: poles. The background of images in (d-i) is in dark blue, representing pixels not falling in the corresponding level.

Difficulty-aware uncertainty Score (DS) Assume M^c is the uncertainty map generated with traditional methods like Entropy, we can define the equation below to make each pixel aware of its semantic difficulty.

$$\mathcal{S}_{DS} = \frac{1}{K} \sum_{k=1}^{K} M_k^c \cdot M_k^d, \tag{6}$$

where M_k^c and M_k^d are the uncertainty score and difficulty score of the kth pixel, K is the total pixel number, $\mathcal{S}_{DS}$ is the average difficulty-aware uncertainty score for selecting samples with the highest uncertainty.

Difficulty-aware semantic Entropy (DE) This acquisition function is inspired by the laddered semantic difficulty reflected on M^d. Usually, pixels from the same semantic area have almost the same semantic difficulty scores, best viewed in Fig. 5(c). In this example, we quantify the difficulty of pixels in Fig. 5(a) into 6 levels in Fig. 5(d-i), with difficulty scores gradually increasing from level 1 to level 6. Generally, if we quantify the difficulty in an image into L levels, the difficulty entropy acquisition function can be defined below to query samples with more balanced semantic difficulty, which can be viewed as a representation-based method at the image scale.

$$\mathcal{S}_{DE} = -\sum_{l=1}^{l=L} \frac{K_l}{K} log(\frac{K_l}{K}), \tag{7}$$

where K_l is the number of pixels falling in the lth difficulty level, K is the total pixel number, L is the quantified difficulty levels, $\mathcal{S}_{DE}$ is the difficulty-aware semantic entropy for selecting samples with more balanced semantic difficulty. Our full algorithm of DEAL is shown in Algorithm 1.

Algorithm 1: Difficulty-aware Active Learning Algorithm

Input: $\mathcal{D}^a$, $\mathcal{D}^u$, budget m, AL query times N, initialized network parameter Θ
Input: iterations T, weight factor α, quantified difficulty levels L (optional)
Output: $\mathcal{D}^a$, $\mathcal{D}^u$, Optimized Θ
1: **for** $n = 1, 2, ..., N$ **do**
2:    Train the two-branch difficulty learning network on $\mathcal{D}^a$
3:    **for** $t = 1, 2, ..., T$ **do**
4:       Sample (x^a, y^a) from $\mathcal{D}^a$
5:       Compute the segmentation output S^* and result S^p
6:       Obtain M^e according to Eq. 1
7:       Compute the difficulty prediction M^d
8:       Compute $\mathcal{L}_{seg}$, $\mathcal{L}_{dif}$, $\mathcal{L}$ according to Eq. 3, Eq. 4, Eq. 5
9:       Update Θ using gradient descent
10:    **end for**
11:    Rank x^u based on Eq. 6 or Eq. 7
12:    Select $\mathcal{D}^s$ with top m samples
13:    Annotate $\mathcal{D}^s$ by oralces
14:    $\mathcal{D}^a \leftarrow \mathcal{D}^a + \mathcal{D}^s$
15:    $\mathcal{D}^u \leftarrow \mathcal{D}^u - \mathcal{D}^s$
16: **end for**
17: **return** $\mathcal{D}^a$, $\mathcal{D}^u$, Optimized Θ

---

4   Experiments and Results

In this section, we first describe the datasets we use to evaluate our method and the implementation details, then the baseline methods, finally compare our results with these baselines.

4.1   Experimental Setup

Datasets We evaluate DEAL on two street scene semantic segmentation datasets: CamVid [21] and Cityscapes [8]. For Cityscapes, we randomly select 300 samples from the training set as the validation set, and the original validation set serves as the test set, same to [15]. The detailed configurations are list in Table 1. For each dataset, we first randomly sample 10% data from the training set as the initial annotated dataset $\mathcal{D}^a$, then iteratively query 5% new data $\mathcal{D}^s$ from the remaining training set, which serves as the unlabeled data pool $\mathcal{D}^u$. Considering samples in the street scenes have high similarities, we first randomly choose a subset from $\mathcal{D}^u$, then query m samples from the subset, same to [37].

Table 1: Evaluation datasets.

Dataset	Classes	Train	Valid	Test	Initial labeled	Budget	Image Size
CamVid [21]	11	370	104	234	40	20	360×480
Cityscapes [8]	19	2675	300	500	300	150	688×688

Implementation Details We adopt the Deeplabv3+ [38] architecture with a Mobilenetv2 [39] backbone. For each dataset, we use mini-batch SGD [40] with momentum 0.9 and weight decay $5e^{-4}$ in training. The batch size is 4 and 8 for CamVid and Cityscapes, respectively. For all methods and the upper bound method with the full training data, we train 100 epochs with an unweighted cross-entropy loss function. Similar to [38], we apply the "poly" learning rate strategy and the initial learning rate is 0.01 and multiplied by $(1 - \frac{iter}{max\_iter})^{0.9}$. To accelerate the calculation of the probability attention module, the input of the difficulty branch is resized to 80×60 and 86×86 for CamVid and Cityscapes.

4.2   Evaluated Methods

We compare DEAL with the following methods. *Random* is a simple baseline method. *Entropy* and *QBC* are two uncertainty-based methods. *Core-set* and *VAAL* are two representation-based methods. *DEAL (DS)* and *DEAL (DE)* are our methods with different acquisition functions.

- **Random**: each sample in $\mathcal{D}^u$ is queried with uniform probability.
- **Entropy** (Uncertainty): we query samples with max mean entropy of all pixels. [13] and [20] have verified this method is quite competitive in image classification and segmentation tasks.
- **QBC** (Uncertainty): previous methods designed for semantic segmentation, like [4, 11, 22, 23], all use a group of models to measure uncertainty. We use the efficient MC dropout to represent these methods and report the best performance out of both the max-entropy and variation-ratio acquisition functions.
- **Core-set** (Representation): we query samples that can best cover the entire data distribution. We use the global average pooling operation on the encoder output features of Deeplabv3+ and get a feature vector for each sample. Then *k-Center-Greedy* strategy is used to query the most informative samples, and the distance metric is l_2 distance according to [14].
- **VAAL** (Representation): as a new state-of-the-art task-agnostic method, the sample query process of VAAL is totally separated from the task learner. We use this method to query samples that are most likely from $\mathcal{D}^u$ and then report the performance with our task model.
- **DEAL (DS)**: out method with DS acquisition function. We employ *Entropy* uncertainty maps in our experiments.
- **DEAL (DE)**: out method with DE acquisition function. The quantified difficulty levels L are set 8 and 10 for CamVid and Cityscapes, respectively.

4.3   Experimental Results

The mean Intersection over Union (mIoU) at each AL stage: 10%, 15%, 20%, 25%, 30%, 35%, 40% of the full training set, are adopted as the evaluation metric. Every method is run 5 times and the average mIoUs are reported.

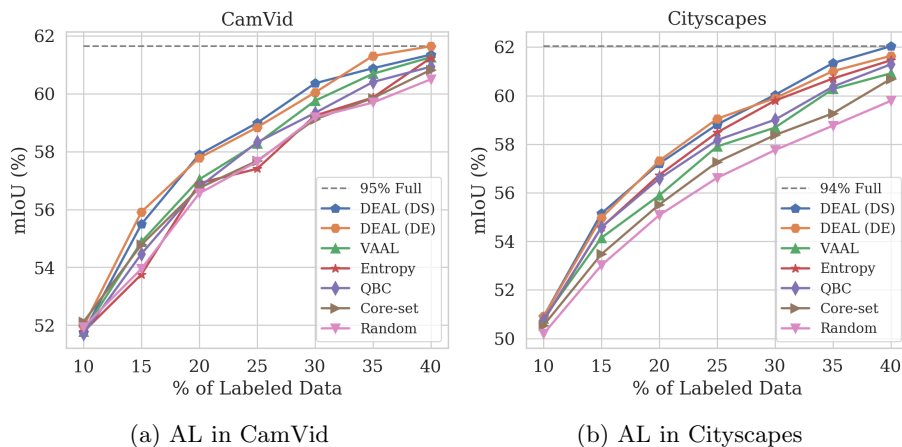(a) AL in CamVid   (b) AL in Cityscapes

Fig. 6: DEAL performance on CamVid [21] and Cityscapes [8]. Every method is evaluated by the average mIoU of 5 runs. The dashed line represents the upper performance we can reach compared with the full training data.

Results on CamVid Fig. 6(a) shows results on a small dataset CamVid. Both *DEAL (DS)* and *DEAL (DE)* outperform baseline methods at each AL stage. We can obtain a performance of 61.64% mIoU with 40% training data, about 95% of the upper performance with full training data. *Entropy* can achieve good results at the last stage. However, it's quite unstable and depends much on the performance of current model, making it behave poorly and exceeded by *Random* at some stages. On the contrary, *DEAL (DS)* behaves better with the difficulty attention. *QBC* has a more stable growth curve as it depends less on the single model. Representation-based methods like *VAAL* and *Core-set* behave much better at earlier stages like 15% and 20%. However, *Core-set* lags behind later while *VAAL* still works well. Also, the experiment results suggest that the data diversity is more important when the entire dataset is small.

Results on Cityscapes Fig. 6(b) shows results on a larger dataset Cityscapes. The budget is 150 and all methods have more stable growth curves. When the budget is enough, *Entropy* can achieve better performance than other baseline methods. Consistently, with semantic difficulty, both *DEAL (DS)* and *DEAL (DE)* outperform other methods. Table 2 shows the per-class IoU for each method at the last AL stage (40% training data). Compared with *Entropy*, our method are more competitive on the difficult classes, such as *pole*, *traffic sign*, *rider* and *motorcycle*. For representation-based methods, the gap between *Core-set* and *VAAL* is more obvious, suggesting that *Core-set* is less effective when the input has a higher dimension. And *VAAL* is easily affected by the performance of the learned variational autoencoder, which introduces more uncertainty to the active learning system. If continue querying new data, our method will reach the upper performance of full training data with about 60% data.

Table 2: Per-class IoU of and mIoU [%] on Cityscapes original validation set with 40% training data. For clarity, only the average of 5 runs are reported, and the best and the second best results are highlighted in **bold** and **<u>underline bold</u>**.

Method	Road	Side-walk	Build-ing	Wall	Fence	Pole	Traffic Light	Traffic Sign	Vege-tation	Terrain
Random	96.03	72.36	86.79	43.56	44.22	36.99	35.28	53.87	86.91	54.58
Core-set	96.12	72.76	87.03	44.86	**45.86**	35.84	34.81	53.07	87.18	53.49
VAAL	**<u>96.22</u>**	**<u>73.27</u>**	86.95	**47.27**	43.92	37.40	36.88	54.90	87.10	54.48
QBC	96.07	72.27	87.05	**<u>46.89</u>**	44.89	37.21	37.57	54.53	87.51	55.13
Entropy	**96.28**	**73.31**	**87.13**	43.82	43.87	38.10	**37.74**	55.39	**<u>87.52</u>**	53.68
DEAL (DS)	96.21	72.72	86.94	46.11	44.22	**<u>38.18</u>**	**<u>37.62</u>**	**55.66**	87.34	**<u>55.62</u>**
DEAL (DE)	95.89	71.69	**<u>87.09</u>**	45.61	**<u>44.94</u>**	**38.29**	36.51	**<u>55.47</u>**	**87.53**	**56.90**

	Sky	Pedes-trian	Rider	Car	Truck	Bus	Train	Motor-cycle	Bicycle	mIoU
Random	91.47	62.74	37.51	88.05	54.64	61.00	43.69	30.58	55.67	59.79
Core-set	91.89	62.48	36.28	87.63	57.25	**<u>67.02</u>**	**56.59**	29.34	53.56	60.69
VAAL	91.63	63.44	38.92	87.92	50.15	63.70	52.36	35.99	54.97	60.92
QBC	91.87	63.79	38.76	88.04	53.88	65.92	**<u>54.32</u>**	32.68	56.15	61.29
Entropy	**<u>92.05</u>**	**<u>63.96</u>**	38.44	**88.38**	**<u>59.38</u>**	64.64	50.80	36.13	**57.10**	61.46
DEAL (DS)	**92.10**	63.92	**40.39**	87.87	**59.85**	**67.32**	52.30	**<u>38.88</u>**	55.44	**62.04**
DEAL (DE)	91.78	**64.25**	**<u>39.77</u>**	**<u>88.11</u>**	56.87	64.46	50.39	**38.92**	**<u>56.69</u>**	**<u>61.64</u>**

5   Ablation Study

5.1   Effect of PAM

To further understand the effect of PAM, we first visualize the attention heatmaps of the wrong predictions in Fig. 7. For each row, three points are selected from *error mask* and marked as $\{1, 2, 3\}$ in Fig. 7(a,b,c). In the first row, point 1 is from *road* and misclassified as *bicyclist*, we can observe that its related classes are *bicyclist*, *road* and *sidewalk* in Fig. 7(d). Point 2 is from *buildings* and misclassified as *bicyclist*, too. Point 3 is from *sign symbol* and misclassified as *tree*, we can also observe its related semantic areas in Fig. 7(f).

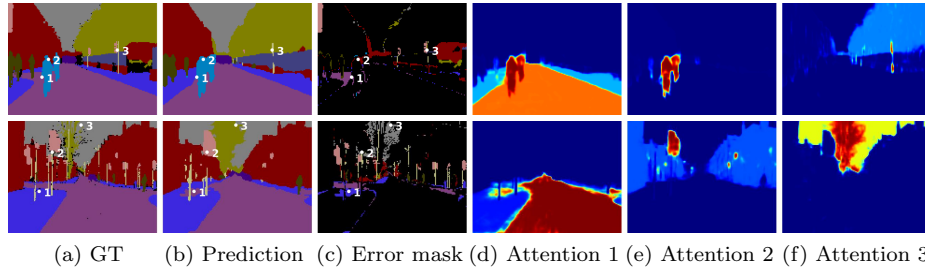(a) GT   (b) Prediction (c) Error mask (d) Attention 1 (e) Attention 2 (f) Attention 3

Fig. 7: Attention heatmaps of three selected wrong predicted pixels on CamVid [21]. (a) GT. (b) Prediction. (c) Error mask. (d,e,f) Attention heatmaps of the three selected points, which are marked as $\{1,2,3\}$ in (a,b,c). The warmer color means the more dependency.

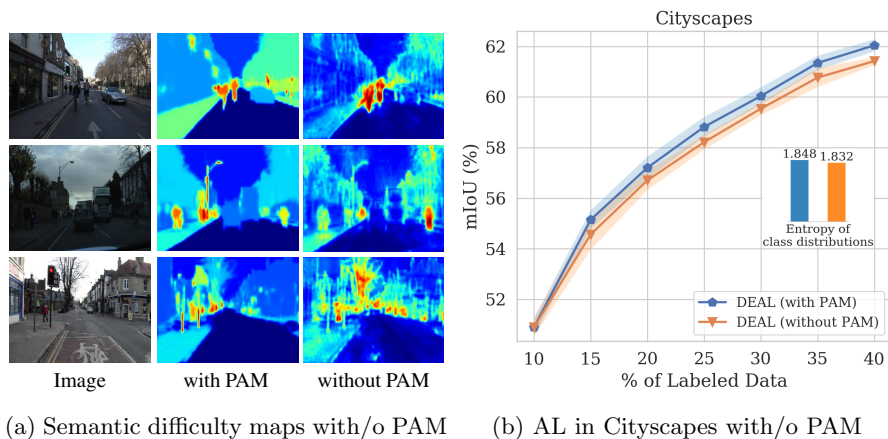(a) Semantic difficulty maps with/o PAM   (b) AL in Cityscapes with/o PAM

Fig. 8: Ablation study on PAM. (a) The first column are images from CamVid [21], the second and third columns are semantic difficulty maps learned with and without PAM. (b) DEAL performance on Cityscapes [8] with and without PAM. We report the mean and standard deviation of 5 runs and the average entropy of class distributions of all AL stages.

Then we conduct an ablation study by removing PAM and directly learning the semantic difficulty map without the attention among pixels. The qualitative results are shown in Fig. 8(a). Basically, without the long-range dependencies, pixels of the same semantic can learn quite different scores because the learned score of each pixel is more sensitive to its original uncertainty. Combined with PAM, we can learn more smooth difficulty map, which is more friendly to the annotators since the aggregated semantic areas are close to the labeling units in the real scenario. Also, we compare this ablation model with our original model on Cityscapes in Fig. 8 (b). DEAL with PAM can achieve a better performance at each AL stage. DEAL without PAM fails to find samples with more balanced semantic difficulty, which makes it get a lower entropy of class distributions.

5.2   Branch Position

In this section, we discuss the branch position of our framework. In our method above, the semantic difficulty branch is simply added after the segmentation branch. It may occur to us that if the segmentation branch performs poorly, the difficulty branch will perform poorly, too. These two tasks should be separated earlier and learn independent features. Thus, we modify our model architecture and branch out two tasks earlier at the boarder of encoder and decoder based on the Deeplabv3+[38] architecture, as shown in Fig. 9(a). Also, we compare the AL performance on Cityscapes with both architectures in Fig. 9(b). The performance of the modified version is slightly poor than the original version but still competitive with other methods. However, this modified version requires

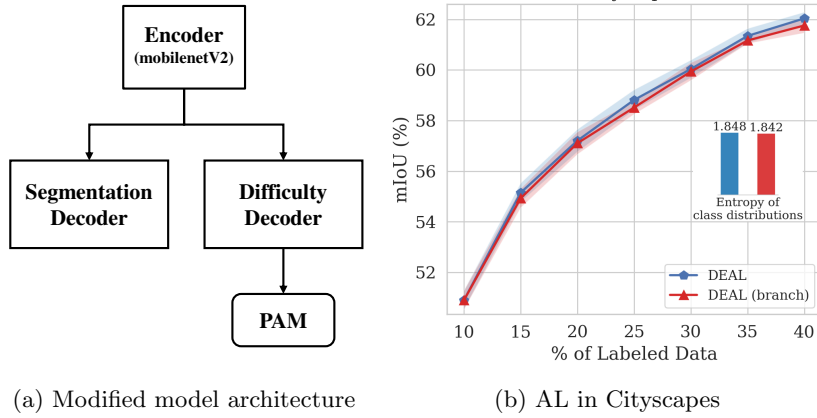(a) Modified model architecture   (b) AL in Cityscapes

Fig. 9: Ablation study on branch position. (a) Modified model architecture. (b) DEAL performance on Cityscapes with different model architectures. *DEAL (branch)* is the modified version. We also report the mean and standard deviation of 5 runs and the average entropy of class distributions of all AL stages.

more computations, while our original version is simple yet effective and can be easily plugged into any segmentation networks.

6   Conclusion and Future Work

In this work, we have introduced a novel Difficulty-awarE Active Learning (DEAL) method for semantic segmentation, which incorporates the semantic difficulty to select the most informative samples. For any segmentation network, the *error mask* is firstly generated with the predicted segmentation result and GT. Then, with the guidance of *error mask*, the probability attention module is introduced to aggregate similar pixels and predict the semantic difficulty maps. Finally, two acquisition functions are devised. One is combining the uncertainty of segmentation result and the semantic difficulty. The other is solely based on the difficulty. Experiments on CamVid and Cityscapes demonstrate that the proposed DEAL achieves SOTA performance and can effectively improve the performance of hard semantic areas. In the future work, we will explore more possibilities with the semantic difficulty map and apply it to region-level active learning method for semantic segmentation.

7   Acknowledgments

References

1. Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., Urtasun, R.: Multinet: Real-time joint semantic reasoning for autonomous driving. In: IEEE Intelligent Vehicles Symposium (IV). (2018) 1013–1020
2. Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., Wiesbeck, W., Dietmayer, K.: Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems (ITS) (2020)
3. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer (2015) 234–241
4. Yang, L., Zhang, Y., Chen, J., Zhang, S., Chen, D.Z.: Suggestive annotation: A deep active learning framework for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer (2017) 399–407
5. Zheng, H., Zhang, Y., Yang, L., Liang, P., Zhao, Z., Wang, C., Chen, D.Z.: A new ensemble learning framework for 3d biomedical image segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). Volume 33. (2019) 5909–5916
6. Azimi, S.M., Henry, C., Sommer, L., Schumann, A., Vig, E.: Skyscapes fine-grained semantic understanding of aerial scenes. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). (2019) 7393–7403
7. Puang, E.Y., Lehner, P., Marton, Z.C., Durner, M., Triebel, R., Albu-Schäffer, A.: Visual repetition sampling for robot manipulation planning. In: International Conference on Robotics and Automation (ICRA). (2019) 9236–9242
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 3213–3223
9. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 633–641
10. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Transactions on Circuits and Systems for Video Technology (CSVT) **27** (2016) 2591–2600
11. Gorriz, M., Carlier, A., Faure, E., Giro-i Nieto, X.: Cost-effective active learning for melanoma segmentation. In: Workshop of Advances in Neural Information Processing Systems. (2017)
12. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the International Conference on Machine Learning (ICML). (2017) 1183–1192
13. Yoo, D., Kweon, I.S.: Learning loss for active learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 93–102
14. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A coreset approach. In: International Conference on Learning Representations (ICLR). (2017)
15. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). (2019) 5972–5981

16. Gissin, D., Shalev-Shwartz, S.: Discriminative active learning. arXiv preprint arXiv:1907.06347 (2019)
17. Shannon, C.E.: A mathematical theory of communication. Bell system technical journal **27** (1948) 379–423
18. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Workshop on Computational learning theory. (1992) 287–294
19. Kuo, W., Häne, C., Yuh, E., Mukherjee, P., Malik, J.: Cost-sensitive active learning for intracranial hemorrhage detection. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer (2018) 715–723
20. Casanova, A., Pinheiro, P.O., Rostamzadeh, N., Pal, C.J.: Reinforced active learning for image segmentation. In: International Conference on Learning Representations (ICLR). (2020)
21. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters **30** (2009) 88–97
22. Tan, Y., Yang, L., Hu, Q., Du, Z.: Batch mode active learning for semantic segmentation based on multi-clue sample selection. In: Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM). (2019) 831–840
23. Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., Rother, C.: Cereals: cost-effective region-based active learning for semantic segmentation. In: British Machine Vision Conference (BMVC). (2018)
24. Siddiqui, Y., Valentin, J., Nießner, M.: Viewal: Active learning with viewpoint entropy for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2020) 9433–9443
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems (NeurIPS). (2017) 5998–6008
26. Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. In: International Conference on Learning Representations (ICLR). (2017)
27. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based lstm for aspect-level sentiment classification. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP). (2016) 606–615
28. Zhou, C., Bai, J., Song, J., Liu, X., Zhao, Z., Chen, X., Gao, J.: Atrank: An attention-based user behavior modeling framework for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). (2018)
29. Chaudhari, S., Polatkan, G., Ramanath, R., Mithal, V.: An attentive survey of attention models. ArXiv **abs/1904.02874** (2019)
30. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: European Conference on Computer Vision (ECCV). (2018) 267–283
31. Yuan, Y., Wang, J.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)
32. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 603–612
33. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 7794–7803

34. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 3146–3154
35. Settles, B.: Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences (2009)
36. Scheffer, T., Decomain, C., Wrobel, S.: Active hidden markov models for information extraction. In: International Symposium on Intelligent Data Analysis (IDA), Springer (2001) 309–318
37. Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M.: The power of ensembles for active learning in image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 9368–9377
38. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European conference on computer vision (ECCV). (2018) 801–818
39. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 4510–4520
40. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NeurIPS). (2012) 1097–1105