Joint Selection and Classification for Active Query

- Hajin Shim, Sung Ju Hwang, Eunho Yang. Joint Active Feature Acquisition and Classification with Variable-Size Set Encoding. NeruIPS, 2018.

- Donggeun Yoo, In So Kweon. Learning Loss for Active Learning. CVPR, 2019.

KAIST, South Korea

Active Feature Acquisition

- At $t = 0$, we start with an empty acquired set $\mathcal{O}_0 := \emptyset$
- At every time step t, we choose the subset of unselected features, $\mathcal{S}_t^{(i)} \subseteq \{1, \ldots, p\} \setminus \mathcal{O}_{t-1}^{(i)}$

  examine the values of missing entries $S_t^{(i)}$ at the cost $\boldsymbol{c}_t^{(i)} := \sum_{j \in \mathcal{S}_t^{(i)}} c_j$

$$\mathcal{O}_t^{(i)} := \mathcal{S}_t^{(i)} \cup \mathcal{O}_{t-1}^{(i)}$$

Active Feature Acquisition

$$f_\theta \qquad \boldsymbol{x} = (x_1, x_2, \ldots, x_p) \longrightarrow y \in \mathcal{Y} := \{1, 2, \ldots, K\}$$

$$\underset{\theta,\vartheta}{\text{minimize}} \; \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(f_\theta\left(\boldsymbol{x}^{(i)}, \boldsymbol{z}_\vartheta^{(i)}\right), y^{(i)}\right) + \lambda \boldsymbol{c}^\top \boldsymbol{z}_\vartheta^{(i)} \qquad (1)$$

$\boldsymbol{z}_\vartheta \in \{0,1\}^p$: encodes whether each feature is acquired at the end

$\boldsymbol{c}$: costs vector for every feature

Alternating minimization:

- Solving (1) with respect to θ_k given υ_{k-1} is trivial. The only challenge here is to find a way of efficiently handing missing features.
- Solving (1) with respect to υ_k given θ_{k-1} is not trivial

Define a Markov decision process (MDP) to solve (1) w.r.t. υ

RL Constructions

state the concatenation of $\boldsymbol{z}_t$ and $\boldsymbol{x}_t$

action $\{1, \ldots, p, \emptyset\}$

reward negative acquisition cost

$(\boldsymbol{s}_0, a_0, r_1, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_T, a_T = \emptyset, r_{T+1})$ r_{t+1} is set as $-\lambda c_{a_t}$ for $t = 0, \ldots, T-1$

final reward r_{T+1} $-\mathcal{L}(f_{\theta_{k-1}}(\boldsymbol{x}_T, \boldsymbol{z}_T), y)$

Theorem 1. *Consider some policy π_{ϑ_k} parameterized by ϑ. Suppose that this policy π_{ϑ_k} is the optimal of Markov decision process described in Section 3.1 given the classifier parameter θ_{k-1}. Then, π_{ϑ_k} is also the optimal solution of (1) with respect to ϑ given θ_{k-1}.*

Feature-level Set Encoding for Joint Learning

Feature-level Set Encoding

state-action value function Q (parameterized by ϑ)
classifier C (parameterized by θ)
} share the input $\boldsymbol{s}_t$

At every time t, $\boldsymbol{s}_t := (\boldsymbol{x}_t, \boldsymbol{z}_t)$ $\boldsymbol{h}_t := \texttt{Enc}(\boldsymbol{s}_t)$

$$\boldsymbol{q}_t := Q(\boldsymbol{h}_t), \quad Q(\boldsymbol{s}_t, a) := [\boldsymbol{q}_t]_a \quad \text{for every action } a,$$
$$\boldsymbol{p}_t := C(\boldsymbol{h}_t), \quad \mathbb{P}(y|\boldsymbol{s}_t) := \texttt{softmax}(\boldsymbol{p}_t).$$

Set Encoding
i) MLP called reading block, which maps each set element x_i to the real vector m_i
ii) a process block that processes m_i repeatedly with LSTM and the attention mechanism to produce the final set embedding

Learning and inference

Each agent runs for n steps according to its policy based on current Q and gets the experience $(\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}_{t+1}, a_{t+1}, \ldots, \boldsymbol{s}_{t+n})$

After n steps running, Q and C are updated based on the running history.

Learning Q: double DQN

Learning C: $-\log C_{y_{\texttt{true}}}(\boldsymbol{s}_t)$ where $C_{y_{\texttt{true}}}(\boldsymbol{s}_t)$

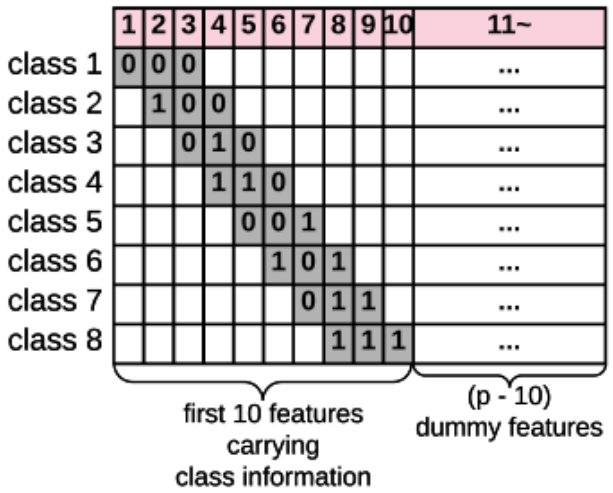
Inference. At test phase, the start state might be completely empty or partially known. Our RL agent determines which features to acquire by greedily selecting the action with the maximum Q-value until $\emptyset$ is chosen. When $\emptyset$ is selected, C makes a prediction based on the acquired features.

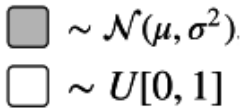
Algorithm 1 Learning process of our framework

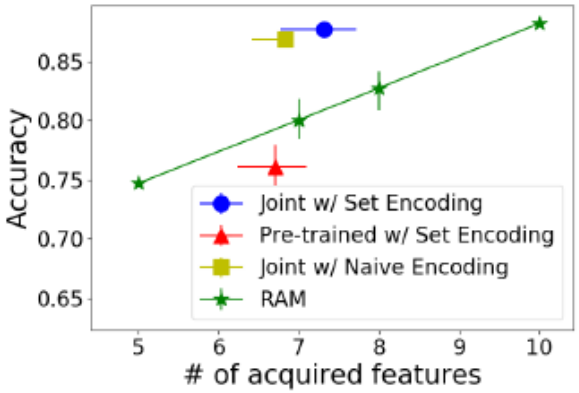
Input: training dataset (X, y) with full information
Ramdomly initialize model parameters θ of C, ϑ of Q
Initialize target network weights $\vartheta' \leftarrow \vartheta$
Pretrain C with full and randomly dropped features with probability 0.5
Initialize exploration
repeat
  // Run N agents for mini-batch $\{x^{(i)}\}$ in parallel
  $t_{start} \leftarrow t$, get state $s_t = (x, z_t)$
  repeat
    Take action a_t according to the ϵ-greedy based on $Q(s_t, a; \vartheta)$
    if a_t is feature acquisition action **then**
      Receive $r_t = -\lambda c_{a_t}$ and feature value x_{a_t}
      $m_t \leftarrow 1$
    else
      Receive $r_t = -\mathcal{L}(f_\theta(s_t), y)$
      $m_t \leftarrow 0$, sample new state $s_{t+1} = (x, \mathbf{0})$
    end if
    $t \leftarrow t + 1$
  until $t - t_{start} == n$
  $R \leftarrow Q(s_t, \text{argmax}_a\, Q(s_t, a; \vartheta); \vartheta')$ (double DQN)
  for $j \in \{t-1, \ldots, t_{start}\}$ **do**
    $R \leftarrow r_j + m_t R$
    $d\vartheta \leftarrow d\vartheta + \frac{\partial (R - Q(s_j, a_j; \vartheta))^2}{\partial \vartheta}$
    $d\theta \leftarrow d\theta + \frac{\partial \mathcal{L}(f_\theta(s_j), y)}{\partial \theta}$
  end for
  update ϑ with $d\vartheta$ and then update θ with $d\theta$
  if t mod $I_{target} == 0$ **then**
    $\vartheta' \leftarrow \vartheta$
  end if
until $t == t_{max}$

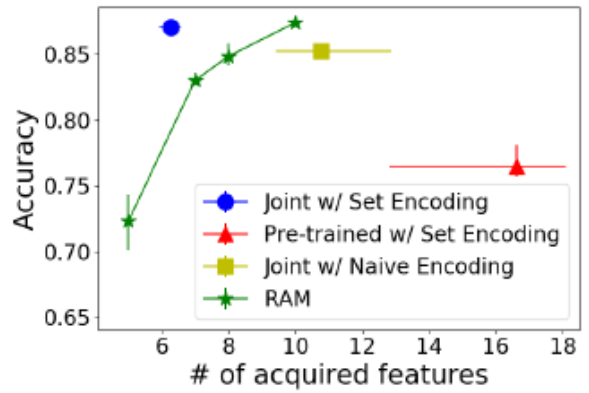
Experiment

Simulated dataset: CUBE-σ









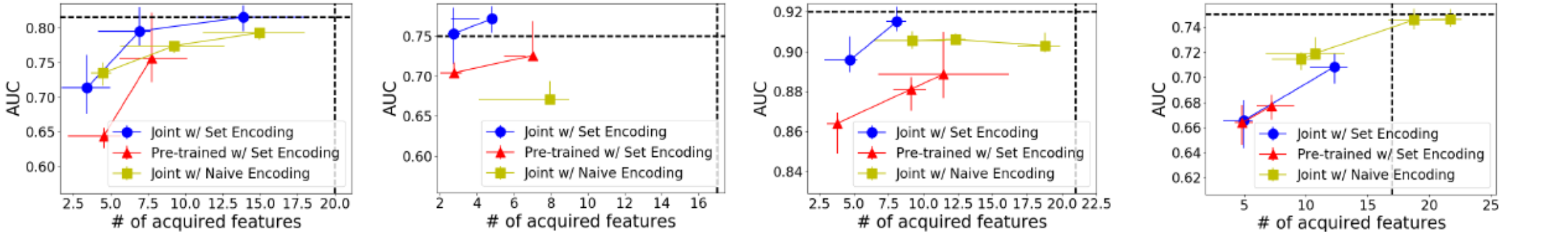
(a) The CUBE dataset

(b) CUBE-0.3 with 20 features (c) CUBE-0.3 with 100 features

Figure 2: (a) The CUBE dataset consists of p-dimensional real valued vectors in 8 classes. The first 10 features only carry class information with three normally distributed entries in the different locations (dimmed in the figure with written mean values) according to the classes. The rest of them are just uniformly random. (b),(c) Comparisons of RAM, our proposed model and its variant on CUBE-0.3. We report averages with 1st and 3rd quantile as error bars of 10 times run.

Experiment

PhysioNet challenge 2012 dataset: classification



(a) Mortality (b) Length of stay < 3 (c) Cardiac condition (d) Recovery from surgery

Figure 3: The comparison on Physionet 2012 data. We report averages with 1st and 3rd quantile as error bars of 10 times run. The black dotted line indicates the result of MLP with group norm.

Experiment

Medical check-up data

Table 2: Result on the check-up data. We report prediction AUC (top) and the average number of acquired features (bottom) and cost (USD; in brackets).

model / task	Baselines	Joint w/ Naive Encoding		Pre-trained w/ Set Encoding		Joint w/ Set Encoding	
	MLP	uniform cost	real cost	uniform cost	real cost	uniform cost	real cost
Fatty Liver	0.812 35 (577.7)	0.775 6.81 (167.0)	0.785 8.7 (20.1)	0.774 6.2 (127.9)	0.779 4.8 (9.2)	0.795 10.3 (149.8)	0.792 7.5 (13.8)
IMT	0.755 22 (1,629.8)	0.742 5.12 (109.9)	0.704 14.22 (11.3)	0.721 3.94 (191.5)	0.760 23.71 (83.8)	0.756 5.57(309.8)	0.750 4.79 (10.0)

Conclusion

- A cost-aware sequential feature selection can be used in situations where the features are not provided in full and each collection of features incurs variable cost, such as with medical examination.
- To solve this problem, we formulated it into an optimization problem of simultaneously minimizing the prediction loss and the feature acquisition costs, and derived a joint learning framework for the classifier and the RL agent.
- We validated our model on both synthetic and real medical datasets to confirm the superiority of proposed feature level set embedding under our joint learning framework.

Introduction

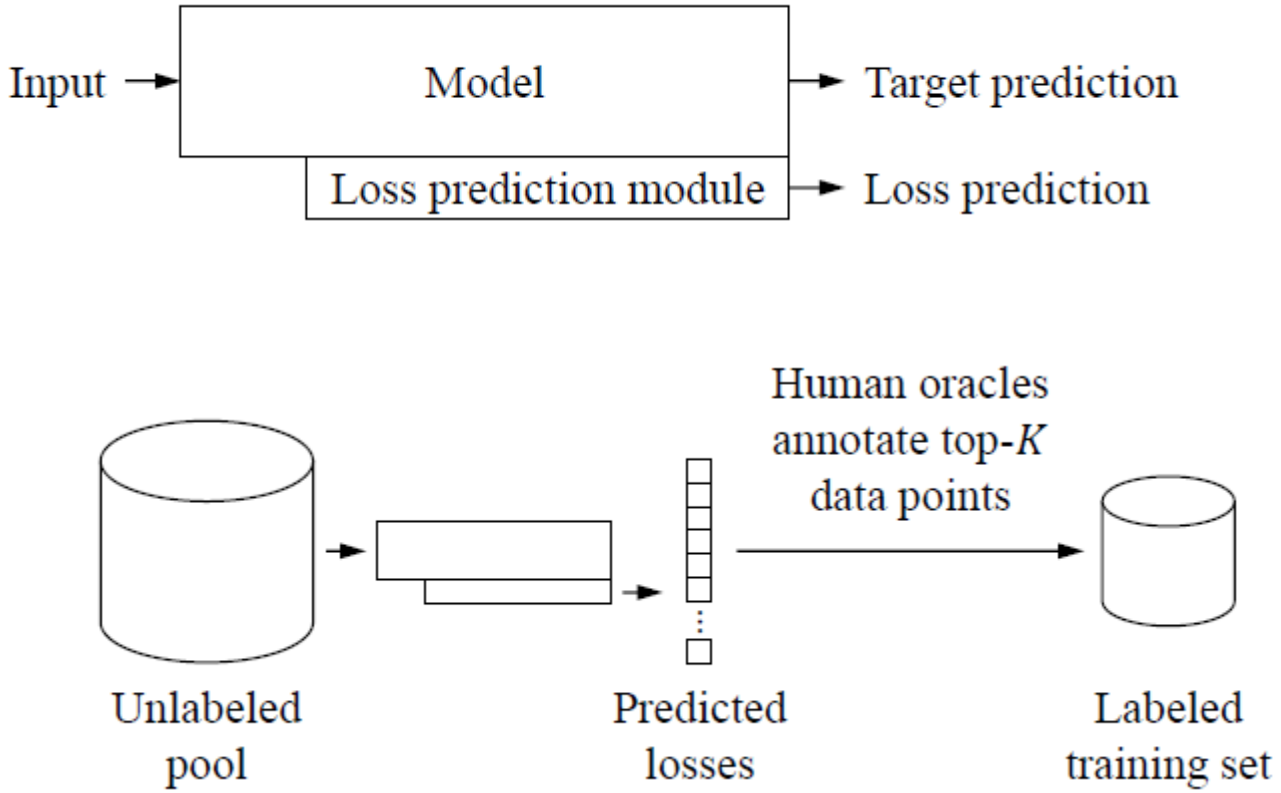
Most of active learning approach require task-specific design or are not efficient in the recent deep networks.

This paper aims to propose a novel active learning method that is simple but task-agnostic, and performs well on deep networks.

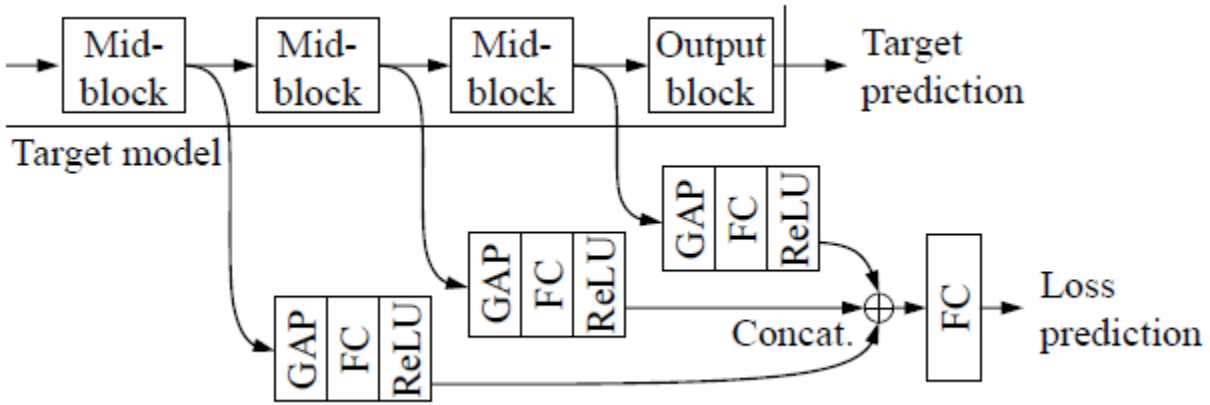
If we can predict the loss of a data point, it becomes possible to select data points that are expected to have high losses, i.e. more informative to the current model.

We attach a “loss prediction module” to a deep network and learn the module to predict the loss of an input data point.

Proposed Method

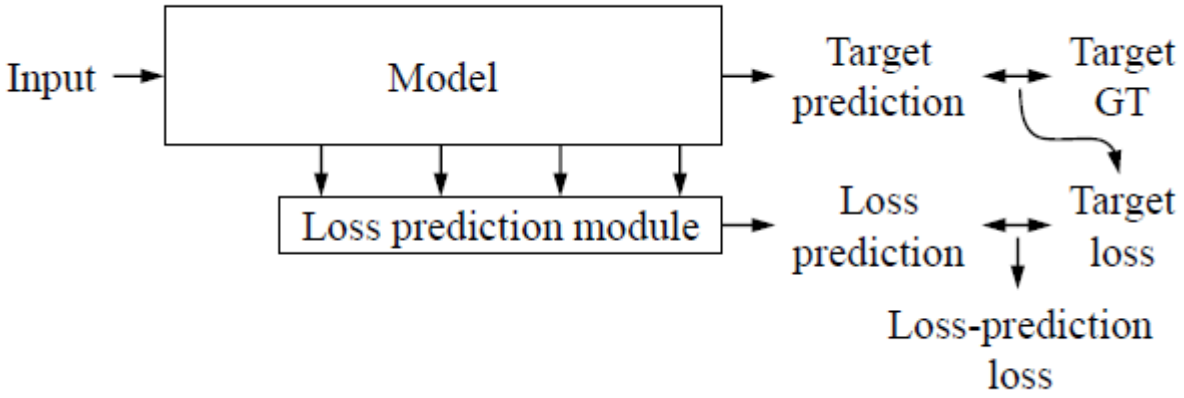


Loss Prediction Module



“Learning this two-story module requires much less memory and computation than the target model. We have tried to make this module deeper and wider, but the performance does not change much.”

Learning Loss



Classification $\hat{y} = \Theta_{\text{target}}(x)$

Loss prediction $\hat{l} = \Theta_{\text{loss}}(h)$

$$L_{\text{target}}(\hat{y}, y) + \lambda \cdot L_{\text{loss}}(\hat{l}, l)$$

Learning Loss

MSE is not a suitable choice for this problem since the scale of the real loss l changes (decreases in overall) as learning of the target model progresses.

- For mini-batch $\mathcal{B}^s \subset \mathcal{L}^s_{K\cdot(s+1)}$

  make $B/2$ data pairs such as $\{x^p = (x_i, x_j)\}$

- For every pair,

$$L_{\text{loss}}(\hat{l^p}, l^p) = \max\left(0, -\mathbb{1}(l_i, l_j)\cdot(\hat{l_i} - \hat{l_j}) + \xi\right)$$

$$\text{s.t.} \quad \mathbb{1}(l_i, l_j) = \begin{cases} +1, & \text{if} \quad l_i > l_j \\ -1, & \text{otherwise} \end{cases}$$

when $l_i > l_j$, this function states that no loss is given to the module only if $\hat{l}_i$ is larger than $\hat{l}_j + \xi$, but otherwise a loss is given to the module to force it to increase $\hat{l}_i$ and decrease $\hat{l}_j$.

- Total loss for batch $\mathcal{B}^s$

$$\frac{1}{B}\sum_{(x,y)\in\mathcal{B}^s} L_{\text{target}}(\hat{y}, y) + \lambda\frac{2}{B}\cdot\sum_{(x^p,y^p)\in\mathcal{B}^s} L_{\text{loss}}(\hat{l^p}, l^p)$$

$$\hat{y} = \Theta_{\text{target}}(x)$$

$$\text{s.t.} \quad \hat{l^p} = \Theta_{\text{loss}}(h^p)$$

$$l^p = L_{\text{target}}(\hat{y^p}, y^p).$$

Experiments

Baselines:

- Random sampling
- Entropy-based sampling
- Core-set sampling

Tasks:

- **image classification** as a **classification** task
- **object detection** as a hybrid task of **classification** and **regression**
- **human pose estimation** as a typical **regression** problem

Image Classification

Dataset: CIFAR-10

Target model: ResNet-18

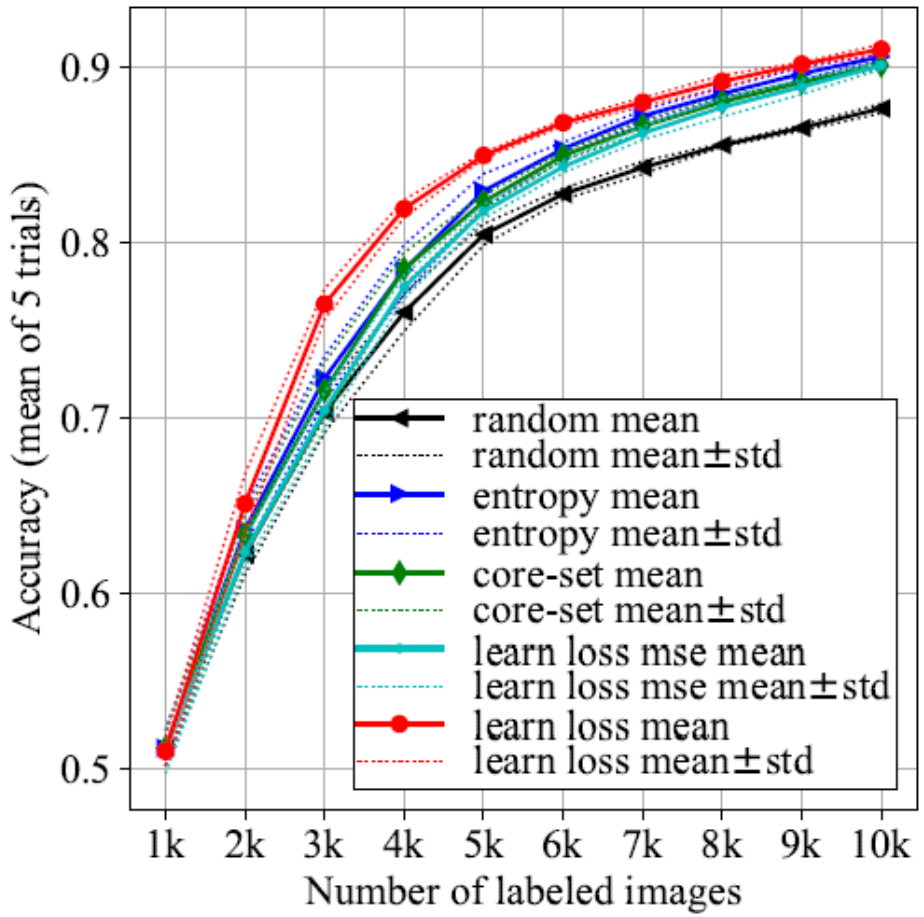


Figure 4. Active learning results of image classification over CIFAR-10.

Object Detection

Dataset: PASCAL VOC 2007+2012

Target model: SSD

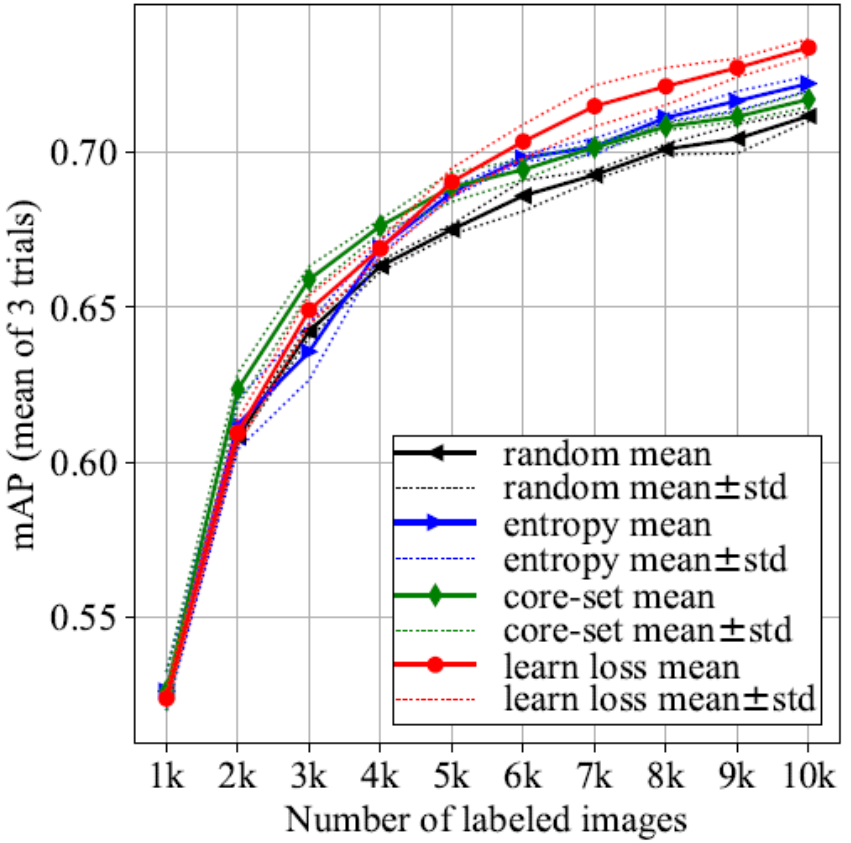


Figure 6. Active learning results of object detection over PASCAL VOC 2007+2012.

Human Pose Estimation

Dataset: MPII dataset

Target model: Stacked Hourglass Networks

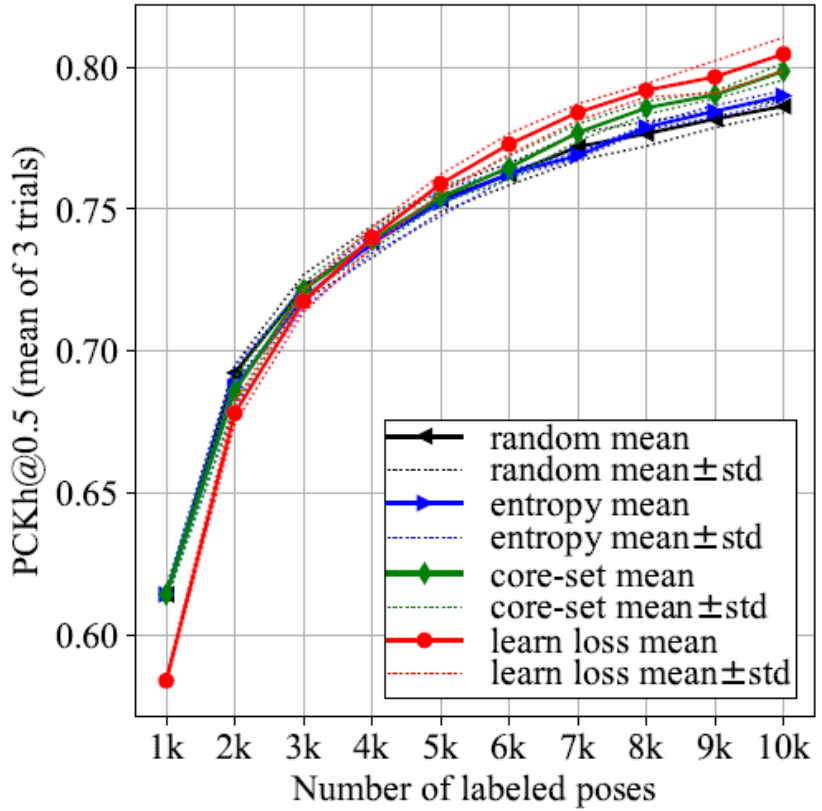


Figure 7. Active learning results of human pose estimation over MPII.

Loss Prediction

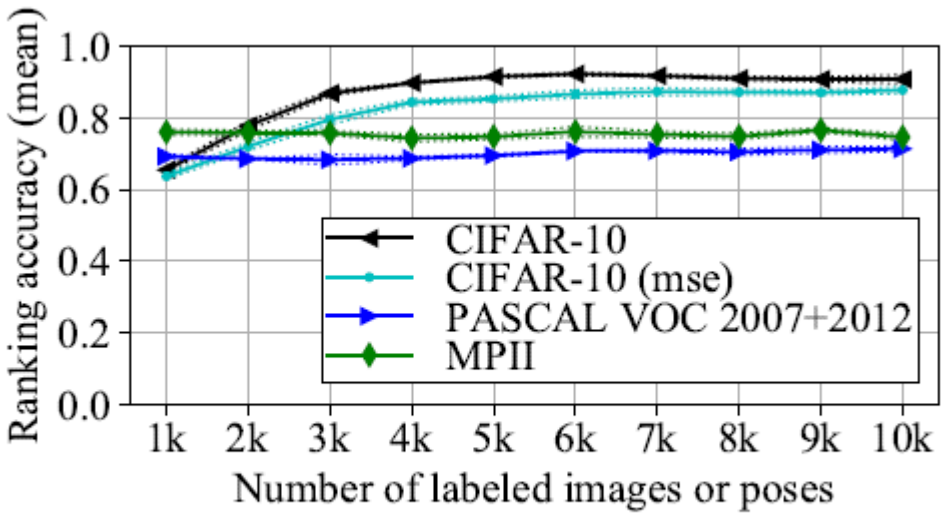
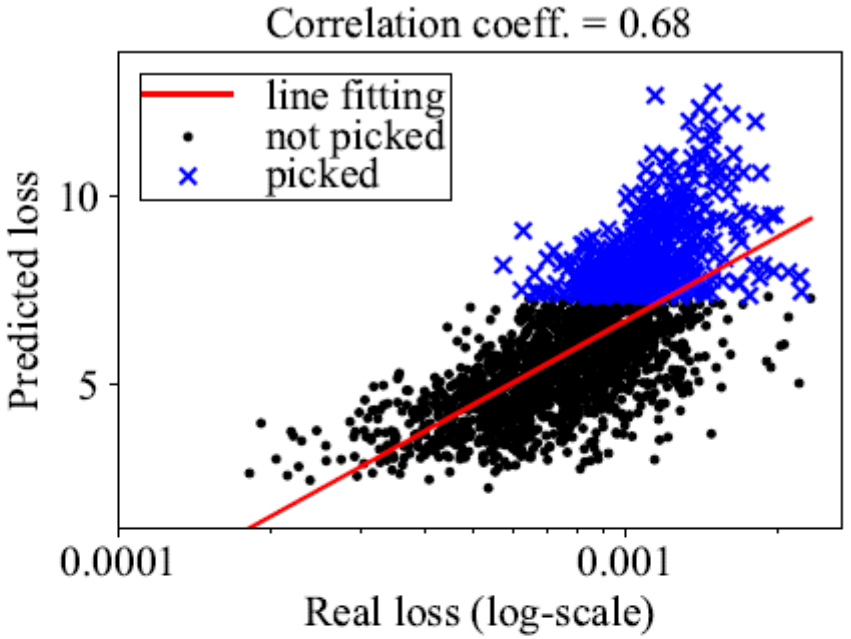
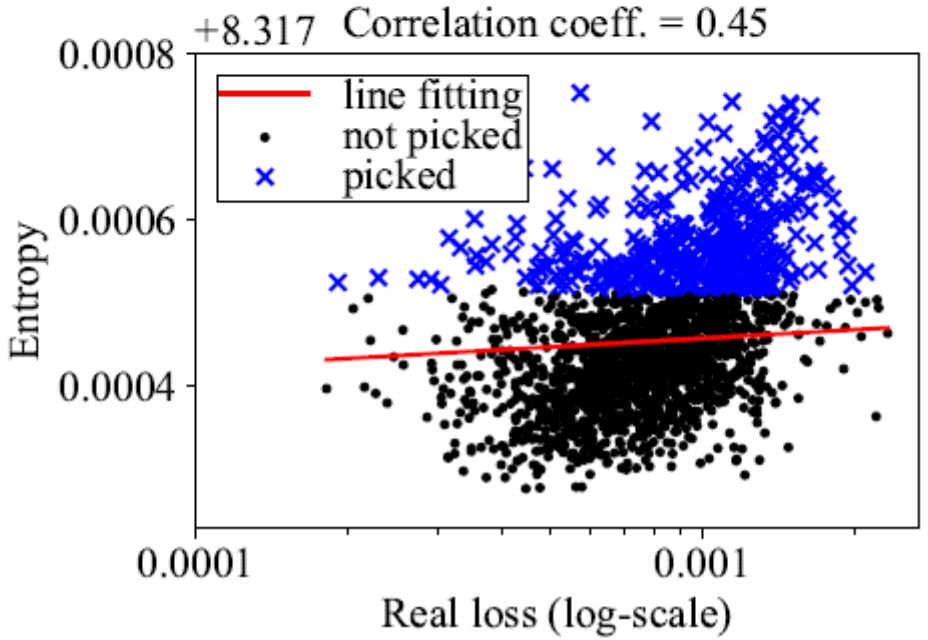


Figure 5. Loss-prediction accuracy of the loss prediction module.





Limitation

- the uncertainty score provided by this method has been effective, the diversity or density of data was not considered
- the loss prediction accuracy was relatively low in complex tasks such as object detection and human pose estimation