# Rapid Performance Gain through Active Model Reuse

Feng Shi
Yu-Feng Li

IJCAI 2019

# Introduction

- Learning models work well on the specific tasks, but have to be discarded once the target task changes.
- **Model reuse** tries to reduce the learning resources for a new target task with the exploitation of pre-trained models.
- Previous model reuse studies usually assume that the labeled data for the target task are **passively** collected.
- **Active Model Reuse** (AcMR) construct queries through pre-trained models to facilitate the active learner when labeled examples are insufficient.
- Different from Active Learning.

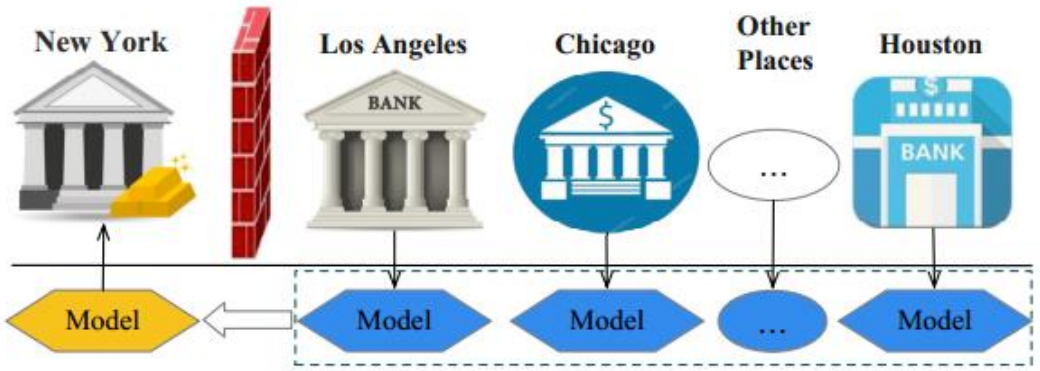



Figure 1: A practical example of model reuse. It is often challenging to share bank data due to the privacy issue. Instead we can reuse pre-trained models of banks to help improve the performance.

# Notation

Table 1: Summary of Notation

| Notation | Meaning |
|---|---|
| $N$ | number of training data |
| $y_t \in \{+1, -1\}$ | ture label of instance $\mathbf{x}_t$ |
| $\mathcal{L}$ | labeled data in the target task |
| $\mathcal{U}$ | unlabeled data in the target task |
| $\{f_1, f_2, \ldots, f_k\}$ | $k$ pre-trained models |
| $f_j^{(t)}$ | the prediction of $f_j$ on $\mathbf{x}_t$ |
| $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_k]$ | the weight vector of $k$ models |
| $l_j^{(t)} = (1 - y^{(t)} f_j^{(t)})_+$ | the hinge loss of $j$-th model for $\mathbf{x}_t$ |
| $L_j = \sum_{\mathbf{x}_t \in \mathcal{L}} l_j^{(t)}$ | the empirical loss of $j$-th model |
| $f_{\mathcal{L}}$ | the active learner built on $\mathcal{L}$ |
| $f_{\mathcal{L}}^{(t)}$ | the prediction of $f_{\mathcal{L}}$ on $\mathbf{x}_t$ |

# Deficiencies of Baseline Approaches

$$\hat{y}^{(t)} = \underset{c\in\{-1,+1\}}{\arg\max} \sum_{j=1}^{k} \eta_j \cdot \mathbb{I}(f_j^{(t)} = c)$$

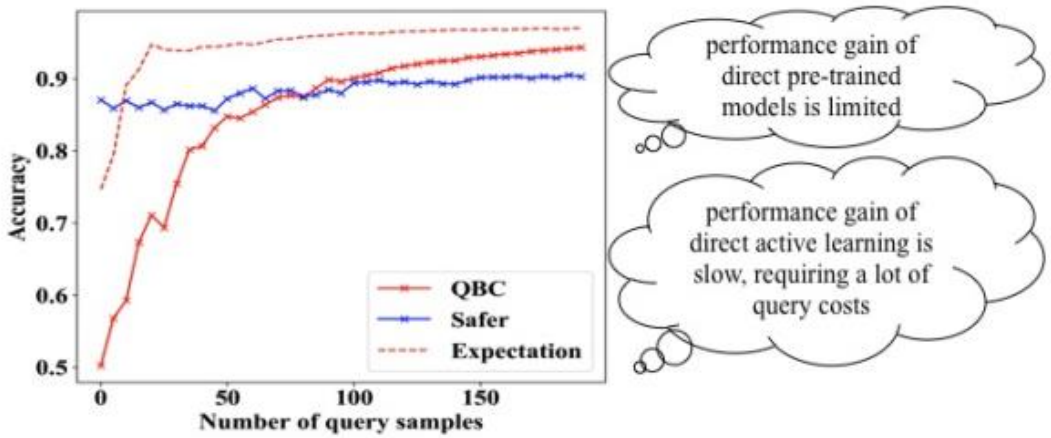



Figure 2: A true example of learning curves of two methods on a classification task. We expect that with the help of pre-trained models, the performance of active learner can get rapid improvement.

This prediction may be risky since we can not get accurate prior weights especially when there are very few labeled examples in target tasks.

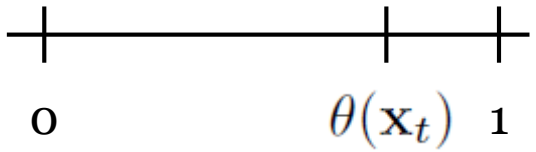
## Strategies:

- Rather than fully trusting pre-trained models, we consider trusting the pre-trained models partially and leverage pre-trained models to **filter out not very necessary queries** generated by direct active learning to facilitate an effective active learner.
- Moreover, once the label capacity is enhanced, we continually **update the confident weights of pre-trained models** to improve their confidence for target task.

# Actively Reuse Pre-trained Models

- Predicted probability of pre-trained model $f_j$ for instance $\mathbf{x}_t$ $\quad P(\hat{y}^{(t)}|\mathbf{x}_t) = \sum_{j=1}^{k} \eta_j P_j(\hat{y}^{(t)}|\mathbf{x}_t) \cdot \mathbb{I}(f_j^{(t)} = \hat{y}^{(t)})$
- Consistency of pre-trained model and active learner $\quad \alpha(\mathbf{x}_t) = \left(1 - \mathbb{I}(\hat{y}^{(t)} \neq f_{\mathcal{L}}^{(t)})\right) P(\hat{y}^{(t)}|\mathbf{x}_t)$
- Query necessity $\quad \theta(\mathbf{x}_t) = \left(1 + \alpha(\mathbf{x}_t)\right)^{-1}$
- Randomly generate a real number R within 0 to 1, and then the **decision function** is defined as

$$\mathbb{F}(\mathbf{x}_t) = \begin{cases} 0, & \text{if } R > \theta(\mathbf{x}_t) \\ 1, & \text{otherwise} \end{cases}$$




Label the instance $\mathbf{x}_t$ by the pre-trained models with probability $1 - \theta(\mathbf{x}_t)$

# Sampling Error

错误地相信了pre-trained models

Let $\varepsilon_p$ and $\varepsilon_a$ denote the expected error of the pre-trained models and the active learner $f_{\mathcal{L}}$ respectively, and let $\delta = \varepsilon_p + \varepsilon_a$.

**Theorem 1.** *In the algorithm* ACMR, *we assume that* $\varepsilon_a \leq \varepsilon_p$, *then the sampling error* $\varepsilon$ *for* ACMR *satisfies:*

$$\varepsilon \leq \frac{\varepsilon_p^2}{1 + (1 - \varepsilon_p)} \tag{7}$$

**Proof.** *According to the analysis of the decision function* $\mathbb{F}(\mathbf{x})$ *described above,* ACMR *makes wrong decision only when both the pre-trained models and the active learner* $f_{\mathcal{L}}$ *agree on the wrong label. In this case,* ACMR *has probability* $1 - \theta(\mathbf{x})$ *to trust the classification result given by the pre-trained models, where* $\theta(\mathbf{x})$ *is defined in Eq.*(5). *Thus, the sampling error of* ACMR *can be written as* $\varepsilon = \varepsilon_p \varepsilon_a (1 - \theta(\mathbf{x})) \leq \varepsilon_p^2 (1 - \theta(\mathbf{x}))$. *Moreover, in this situation,* $\theta(\mathbf{x}) = \frac{1}{1+(1-\varepsilon_p)}$. *Thus,*

$$\varepsilon \leq \frac{\varepsilon_p^2 \times (1 - \varepsilon_p)}{1 + (1 - \varepsilon_p)} \leq \frac{\varepsilon_p^2}{1 + (1 - \varepsilon_p)} \tag{8}$$

# Querying Rate

**Theorem 2.** *In the algorithm* ACMR, *for an unlabeled instance, the probability that* ACMR *queries the label from the experts (with cost) satisfies:*

$$P(Q) \leq \delta + \frac{1-\delta}{1+(1-\varepsilon_p)} \tag{9}$$

**Proof.** *According to the analysis of the decision function* $\mathbb{F}(\mathbf{x})$, ACMR *will query the experts to label the instance when the pre-trained models and the active learner hold different predictions on the classification result. And when the two classifiers agree on the result, it still has probability* $\theta(\mathbf{x})$ *to query the experts. Thus:*

$$\begin{aligned} P(Q) &= \varepsilon_a(1-\varepsilon_p) + \left[\varepsilon_p\varepsilon_a + (1-\varepsilon_p)(1-\varepsilon_a)\right]\theta(\mathbf{x}) \\ &\quad + (1-\varepsilon_a)\varepsilon_p \qquad (10) \\ &= \theta(\mathbf{x}) + (\varepsilon_p + \varepsilon_a - 2\varepsilon_p\varepsilon_a)\big(1-\theta(\mathbf{x})\big) \\ &\leq \delta + (1-\delta)\theta(\mathbf{x}) \\ &\leq \delta + \frac{1-\delta}{1+(1-\varepsilon_p)} \end{aligned}$$

- The more accurate the pre-trained models and active learner become, the less necessarily will we query the experts to label the instance.
- Reduce $\varepsilon_p$ to improve performance in the whole learning process.

# Update Weights for Pre-Trained Models

At time m, the target task receives a training instance $\mathbf{x}_{t(m)}$, the pre-trained models make a prediction and suffer a loss after $\mathbf{y}_{t(m)}$ is revealed.

$$\boldsymbol{\eta}^{(m+1)} = \arg\min_{\boldsymbol{\eta}\in\Theta} \sum_{j\in[k]} \eta_j l_j^{(t(m))} + \lambda \mathcal{D}_{KL}(\boldsymbol{\eta}||\boldsymbol{\eta}^{(m)})$$

$$\eta_j^{(m+1)} = \frac{\eta_j^{(m)} \exp(-l_j^{t(m)}/\lambda)}{\sum_{j'=1}^{k} \eta_{j'}^{(m)} \exp(-l_{j'}^{t(m)}/\lambda)}, \quad j \in [k]$$

**Proposition 1.** *(Weight Concentration). During the weight update procedure in the whole learning process, the weights will concentrate on those pre-trained models who suffer a small cumulative loss on the target task.*

**Proof.** *Through the update rule, we know that the weight associated with the $j$-th previous model is equal to $\eta_j = \frac{\exp(-L_j/\lambda)}{\sum_{j'=1}^{k} \exp(-L_{j'}/\lambda)}, j \in [k]$. The smaller the loss $L_j$ of the pre-trained model, the higher the weight $\eta_j$.*

# AcMR

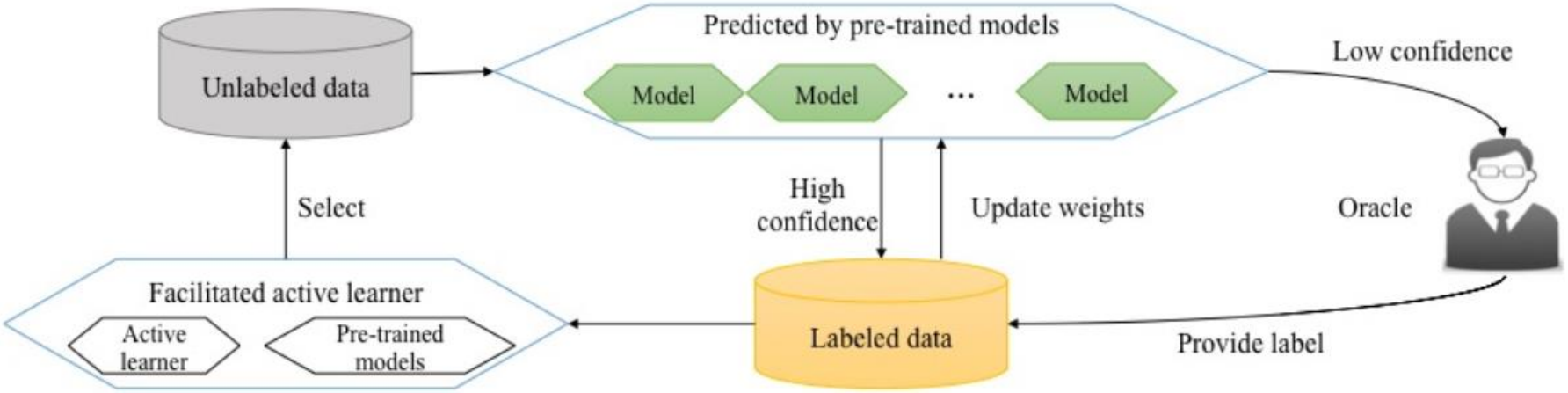



Figure 3: Model architecture of ACMR. In contrast to traditional active learning, we propose to facilitate the active learner with pre-trained models, and we also use pre-trained models to filter out not very necessary queries. During the learning process, the weights of $k$ models are continually updated such that they can predict unlabeled samples more accurately.

**Algorithm 1** The learning algorithm for ACMR

**Input**: labeled dataset $\mathcal{L}$, unlabeled dataset $\mathcal{U}$, sampling size $N$, $k$ pre-trained models $\{f_1, f_2, \ldots, f_k\}$ and $\lambda > 0$
**Output**: the model $f_{\mathcal{L}}$ for the target task

Initialize weight vector $\boldsymbol{\eta} = [1/k, ..., 1/k]$ and $f_{\mathcal{L}}$.
**for** $m = 1, 2, \cdots, N$ **do**
  Select an instance $\mathbf{x}_{t(m)}$ from $\mathcal{U}$ by traditional active learning
  Predict $\mathbf{x}_{t(m)}$ by pre-trained models, and calculate the decision function $\mathbb{F}(\mathbf{x}_{t(m)})$ via Eq.(6)
  **if** $\mathbb{F}(\mathbf{x}_{t(m)}) = 0$ **then**
    $y_{t(m)} \leftarrow \hat{y}^{(t(m))}$
  **else**
    $y_{t(m)} \leftarrow$ query from oracle
    **if** $y_{t(m)} \neq \hat{y}^{(t(m))}$ **then**
      update the weights via Eq.(12)
    **else**
      $\boldsymbol{\eta}^{(m+1)} \leftarrow \boldsymbol{\eta}^{(m)}$
    **end if**
  **end if**
  $\mathcal{L} = \mathcal{L} \cup (\mathbf{x}_{t(m)}, y_{t(m)}), \mathcal{U} = \mathcal{U}/(\mathbf{x}_{t(m)})$;
  Train the learner $f_{\mathcal{L}}$ with $\mathcal{L}$
  Facilitate active learner $f_{\mathcal{L}}$ via Eq.(1)
**end for**
**return** $f_{\mathcal{L}}$

# Experiments

Baselines:

1) An actively transfer learning method **AcTraK** with QBC and random sample selection criteria
2) **Safer**: safely exploits pre-trained models
3) **Logistic Regression** with QBC and random
4) **AcMR** with QBC and random

Compare these methods in different number of samples labeled by experts (30, 60, 90).

Three tasks:
1) Text Classification
2) Sentiment Analysis Task
3) Spam Detection Task

# Text Classification

Dataset: 20 Newsgroups

6 text classification tasks

- comp vs. talk
- comp vs. sci
- comp vs. rec
- rec vs. talk
- rec vs. sci
- sci vs. talk

Table 3: Classification accuracy of compared methods with different queries on 6 classification tasks in 20 Newsgroups dataset. The boldfaces denote the best and the second best methods in terms of the accuracy, and Queries mean that the number of samples labeled by experts.

| Task | Queries | Safer | QBC | AcTrak-QBC | AcMR-QBC | Random | AcTrak-Ran | AcMR-Ran |
|---|---|---|---|---|---|---|---|---|
| Task1 | 30 | .876 ± .033 | .792 ± .118 | .817 ± .107 | **.923 ± .016** | .806 ± .126 | .839 ± .079 | **.926 ± .014** |
| | 60 | .880 ± .029 | .916 ± .029 | .909 ± .052 | **.932 ± .013** | .911 ± .030 | .915 ± .034 | **.933 ± .008** |
| | 90 | .891 ± .040 | .930 ± .022 | .929 ± .026 | **.934 ± .011** | .928 ± .025 | .934 ± .019 | **.936 ± .011** |
| Task2 | 30 | .737 ± .027 | .739 ± .124 | .711 ± .143 | **.838 ± .039** | .689 ± .110 | .756 ± .102 | **.839 ± .049** |
| | 60 | .738 ± .036 | .821 ± .076 | .809 ± .091 | **.870 ± .044** | .833 ± .059 | .842 ± .071 | **.873 ± .038** |
| | 90 | .749 ± .038 | .856 ± .065 | .851 ± .050 | **.885 ± .032** | .871 ± .032 | .861 ± .056 | **.892 ± .027** |
| Task3 | 30 | .944 ± .010 | .788 ± .155 | .803 ± .128 | **.954 ± .012** | .748 ± .146 | .839 ± .095 | **.952 ± .012** |
| | 60 | .945 ± .011 | .903 ± .080 | .920 ± .049 | **.956 ± .011** | .868 ± .115 | .921 ± .050 | **.956 ± .011** |
| | 90 | .945 ± .010 | .937 ± .040 | .948 ± .015 | **.957 ± .012** | .928 ± .043 | .951 ± .020 | **.957 ± .009** |
| Task4 | 30 | .689 ± .059 | .705 ± .119 | .664 ± .133 | **.905 ± .027** | .786 ± .142 | .757 ± .131 | **.910 ± .027** |
| | 60 | .680 ± .049 | .853 ± .085 | .797 ± .126 | **.928 ± .012** | .875 ± .063 | .875 ± .083 | **.927 ± .018** |
| | 90 | .703 ± .040 | .907 ± .044 | .859 ± .086 | **.938 ± .011** | .914 ± .032 | .922 ± .023 | **.933 ± .017** |
| Task5 | 30 | .850 ± .061 | .765 ± .133 | .753 ± .142 | **.915 ± .059** | .765 ± .134 | .758 ± .177 | **.941 ± .015** |
| | 60 | .869 ± .053 | .888 ± .081 | .880 ± .065 | **.941 ± .029** | .864 ± .085 | .833 ± .128 | **.937 ± .027** |
| | 90 | .871 ± .057 | .925 ± .030 | .915 ± .055 | **.949 ± .015** | .909 ± .042 | .886 ± .087 | **.950 ± .011** |
| Task6 | 30 | .770 ± .042 | .645 ± .105 | .646 ± .093 | **.819 ± .029** | .643 ± .104 | .676 ± .113 | **.821 ± .040** |
| | 60 | .763 ± .042 | .679 ± .089 | .718 ± .094 | **.838 ± .032** | .743 ± .103 | .779 ± .090 | **.850 ± .038** |
| | 90 | .788 ± .034 | .747 ± .106 | .781 ± .081 | **.860 ± .031** | .841 ± .083 | .823 ± .075 | **.867 ± .031** |

# Sentiment Analysis

Product reviews from Amazon dataset containing reviews from 4 domains: Book, DVD, Electronics and Kitchen.
Consider each domain as a binary classification task: reviews with rating > 3 are labeled positive(+), those with rating < 3 are labeled negative(-).

Table 4: Classification accuracy of compared methods with different queries on 4 classification tasks in Sentiment dataset. The boldfaces denote the best and the second best methods in terms of the accuracy, and Queries mean that the number of samples labeled by experts.

| Task | Queries | Safer | QBC | AcTrak-QBC | AcMR-QBC | Random | AcTrak-Ran | AcMR-Ran |
|---|---|---|---|---|---|---|---|---|
| Task1 | 30 | **.612 ± .022** | .529 ± .033 | .578 ± .033 | **.607 ± .028** | .577 ± .036 | .582 ± .039 | .598 ± .032 |
| | 60 | .614 ± .018 | .579 ± .048 | .609 ± .035 | **.631 ± .023** | .607 ± .025 | .615 ± .037 | **.631 ± .026** |
| | 90 | .614 ± .018 | .615 ± .047 | .627 ± .038 | **.645 ± .023** | .625 ± .024 | .637 ± .034 | **.649 ± .020** |
| Task2 | 30 | **.648 ± .020** | .562 ± .048 | .582 ± .041 | .610 ± .033 | .594 ± .031 | .594 ± .036 | **.611 ± .046** |
| | 60 | **.648 ± .018** | .591 ± .049 | .626 ± .038 | .631 ± .030 | .623 ± .033 | .629 ± .034 | **.643 ± .032** |
| | 90 | .648 ± .020 | .621 ± .047 | .649 ± .033 | **.660 ± .031** | .650 ± .026 | .657 ± .028 | **.668 ± .029** |
| Task3 | 30 | **.651 ± .044** | .578 ± .057 | .629 ± .033 | **.645 ± .036** | .612 ± .036 | .629 ± .033 | .642 ± .037 |
| | 60 | .654 ± .041 | .640 ± .046 | .665 ± .024 | **.676 ± .028** | .643 ± .032 | .664 ± .032 | **.676 ± .026** |
| | 90 | .665 ± .033 | .672 ± .035 | .685 ± .023 | **.694 ± .025** | .671 ± .027 | .679 ± .029 | **.690 ± .025** |
| Task4 | 30 | .621 ± .049 | .628 ± .032 | .634 ± .033 | **.644 ± .026** | .624 ± .036 | .628 ± .045 | **.641 ± .035** |
| | 60 | .631 ± .051 | .657 ± .035 | .666 ± .024 | **.680 ± .018** | .652 ± .032 | .663 ± .035 | **.684 ± .029** |
| | 90 | .643 ± .054 | .684 ± .026 | .687 ± .026 | **.702 ± .022** | .678 ± .026 | .681 ± .033 | **.705 ± .024** |

# Spam Detection

Dataset: ECML PAKDD Discovery challenge

Table 5: Classification accuracy of compared methods with different queries on 6 classification tasks in Spam dataset. The boldfaces denote the best and the second best methods in terms of the accuracy, and Queries mean that the number of samples labeled by experts.

| Task | Queries | Safer | QBC | AcTrak-QBC | AcMR-QBC | Random | AcTrak-Ran | AcMR-Ran |
|---|---|---|---|---|---|---|---|---|
| Task1 | 30 | .925 ± .028 | .757 ± .152 | .734 ± .144 | **.956 ± .016** | .673 ± .148 | .681 ± .149 | **.951 ± .017** |
| | 60 | .935 ± .028 | .778 ± .142 | .750 ± .116 | **.954 ± .021** | .731 ± .139 | .755 ± .113 | **.957 ± .018** |
| | 90 | .939 ± .021 | .788 ± .117 | .761 ± .092 | **.954 ± .024** | .805 ± .143 | .806 ± .088 | **.957 ± .023** |
| Task2 | 30 | .906 ± .037 | .750 ± .163 | .770 ± .161 | **.952 ± .032** | .707 ± .170 | .751 ± .159 | **.949 ± .024** |
| | 60 | .908 ± .037 | .824 ± .118 | .791 ± .151 | **.965 ± .017** | .787 ± .136 | .849 ± .110 | **.962 ± .021** |
| | 90 | .901 ± .038 | .872 ± .097 | .838 ± .144 | **.968 ± .014** | .843 ± .099 | .910 ± .078 | **.960 ± .019** |
| Task3 | 30 | .897 ± .051 | .846 ± .116 | .875 ± .098 | **.970 ± .024** | .860 ± .064 | .879 ± .047 | **.965 ± .031** |
| | 60 | .914 ± .052 | .895 ± .052 | .916 ± .037 | **.984 ± .014** | .895 ± .055 | .895 ± .044 | **.981 ± .015** |
| | 90 | .913 ± .042 | .928 ± .039 | .908 ± .042 | **.986 ± .013** | .902 ± .046 | .888 ± .039 | **.983 ± .014** |
| Task4 | 30 | **.962 ± .023** | .928 ± .072 | .924 ± .081 | .959 ± .023 | .944 ± .030 | .942 ± .022 | **.961 ± .021** |
| | 60 | .964 ± .023 | .964 ± .023 | .968 ± .019 | **.970 ± .019** | .959 ± .020 | .967 ± .015 | **.967 ± .016** |
| | 90 | .964 ± .023 | .964 ± .023 | .968 ± .019 | **.970 ± .019** | .963 ± .021 | .969 ± .012 | **.973 ± .013** |
| Task5 | 30 | .653 ± .137 | .794 ± .076 | .789 ± .097 | **.896 ± .045** | .762 ± .093 | .667 ± .066 | **.882 ± .055** |
| | 60 | .664 ± .132 | .802 ± .070 | .811 ± .076 | **.911 ± .039** | .781 ± .082 | .702 ± .081 | **.902 ± .044** |
| | 90 | .703 ± .118 | .794 ± .058 | .822 ± .068 | **.924 ± .023** | .796 ± .069 | .719 ± .056 | **.906 ± .031** |
| Task6 | 30 | **.782 ± .061** | .669 ± .097 | .652 ± .091 | **.778 ± .093** | .639 ± .112 | .647 ± .107 | .763 ± .095 |
| | 60 | .793 ± .059 | .768 ± .102 | .711 ± .089 | **.828 ± .058** | .700 ± .106 | .682 ± .103 | **.807 ± .048** |
| | 90 | .794 ± .062 | .793 ± .090 | .793 ± .076 | **.854 ± .048** | .758 ± .099 | .710 ± .087 | **.820 ± .044** |

# Conclusion

- Reusable model design becomes a desire for the rapid expansion of machine learning applications. However, previous model reuse studies assume that the target task receives labeled data passively. This leads to a slow performance improvement to the target task.

- In this paper, we study a kind of new model reuse problem, where the goal is that the model performance for the target task can be quickly improved.

- We propose the ACMR method which constructs queries through pre-trained models when labeled examples are insufficient for the target task, and leverages pre-trained models to filter out not very necessary queries so that considerable queries could be saved compared with direct active learning.