華中科技大學
人工智能与自动化学院
SCHOOL OF ARTIFICIAL INTELLIGENCE AND AUTOMATION.HUST

脑机接口与机器学习实验室
BRAIN-COMPUTER INTERFACE AND MACHINE LEARNING LABORATORY

Active Learning

Dongrui Wu

School of Artificial Intelligence and Automation

Huazhong University of Science and Technology

drwu@hust.edu.cn

Outline

- Weakly Supervised Learning
- Active Learning
- Active Learning for Classification
- Active Learning for Regression
- Deep Active Learning
- Applications
- Conclusions

Weakly Supervised Learning

- **Incomplete supervision**, i.e. only a (usually small) subset of training data is labeled.

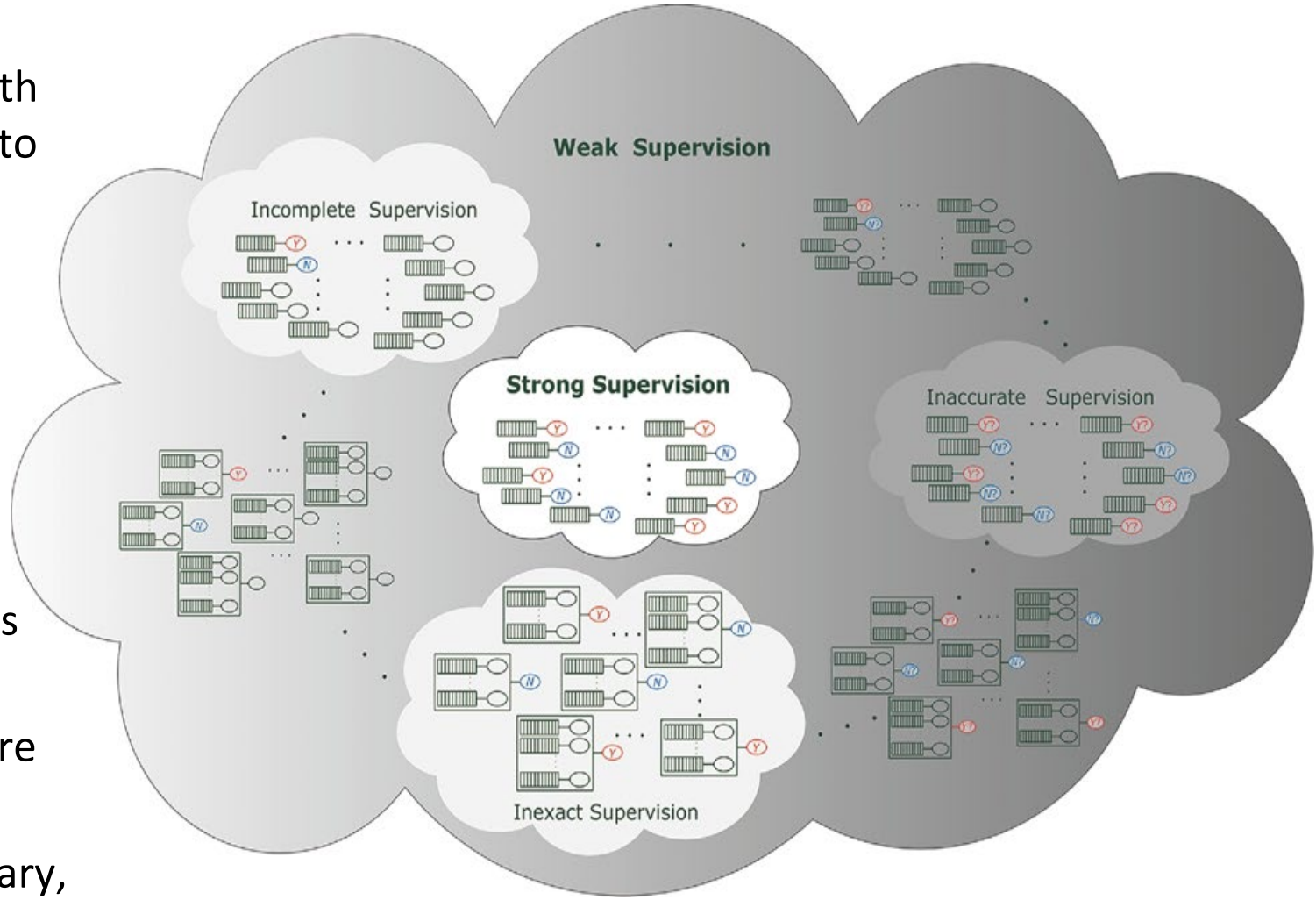
Example: In image categorization the groundtruth labels are given by human annotators; it is easy to get a huge number of images from the Internet, whereas only a small subset of images can be annotated due to the human cost.

- **Inexact supervision**, i.e. only coarse-grained labels are given.

Example: Image categorization. It is desirable to have every object in the images annotated; however, usually we only have image-level labels rather than object-level labels.

- **Inaccurate supervision**, i.e. the given labels are not always groundtruth.

Example: The image annotator is careless or weary, or some images are difficult to categorize.

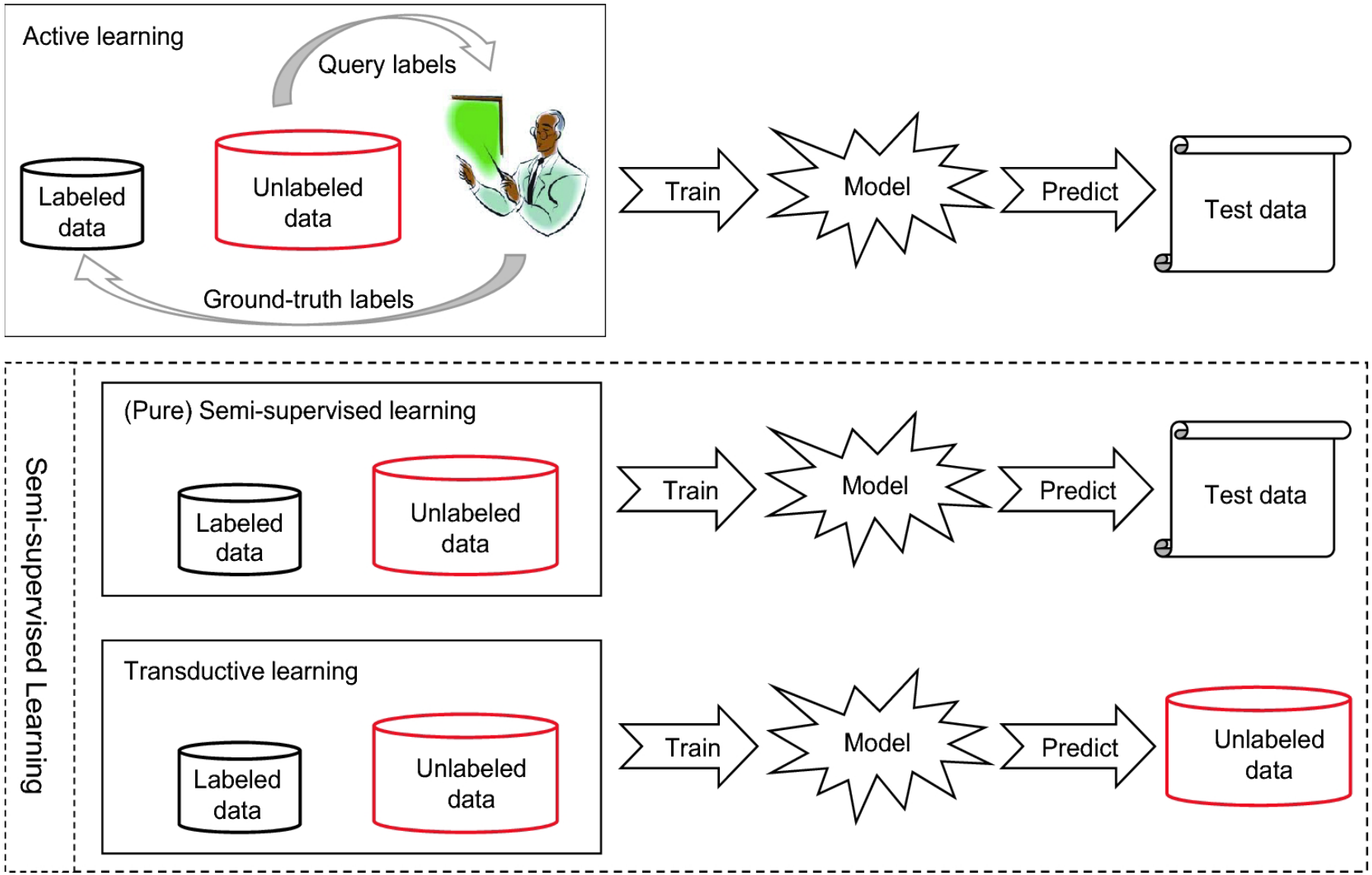


Zhi-Hua Zhou, A brief introduction to weakly supervised learning, *National Science Review*, 5(1): 44-53, 2018. https://doi.org/10.1093/nsr/nwx106

Incomplete Supervision

- **Settings**:
  - ✓ A small amount of labeled data, which is insufficient to train a good learner.
  - ✓ Abundant unlabeled data are available.
- To learn $f: X \mapsto Y$ from a training data set $D=\{(x_1,y_1),\ldots,(x_l,y_l), x_{l+1},\ldots, x_m,$ where there are l labeled training examples (i.e. those given with y_i) and $u = m - l$ unlabeled instances.
- Two major techniques:
  - ✓ **Active learning**: An 'oracle', such as a human expert, can be queried to get groundtruth labels for selected unlabeled instances.
  - ✓ **Semi-supervised learning**: Automatically exploit unlabeled data in addition to labeled data to improve learning performance; no human intervention is assumed.

Incomplete Supervision

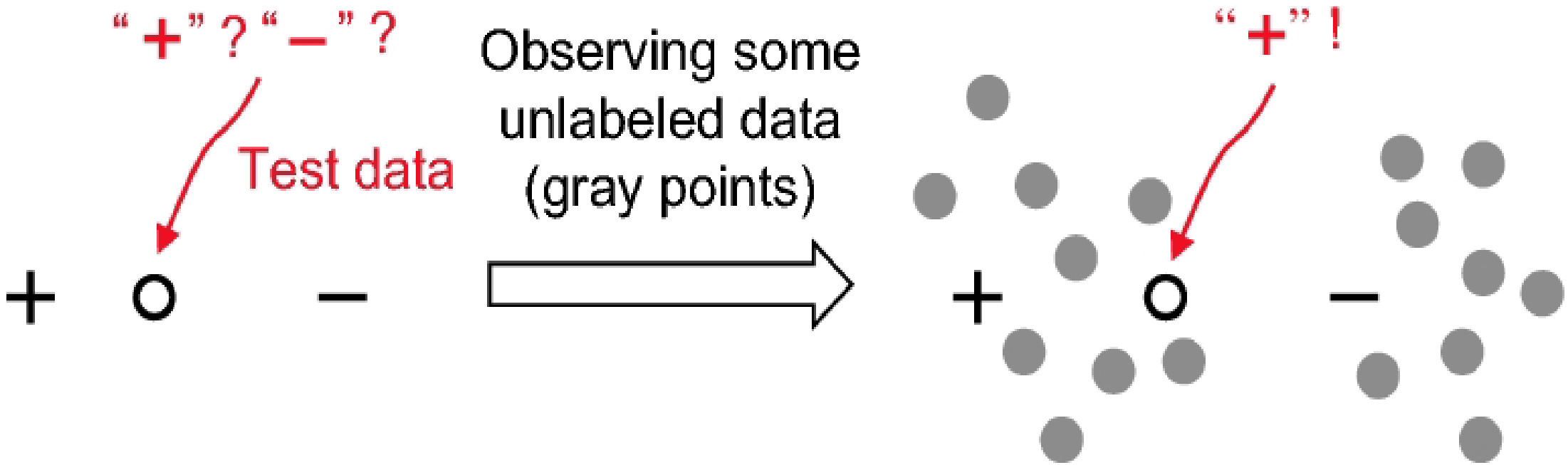


Active Learning

- **Goal:** Minimize the number of queries such that the labeling cost for training a good model can be minimized.
- Two widely used selection criteria:
  - **Informativeness**: How well an unlabeled instance helps reduce the uncertainty of a statistical model.
    - ✓ **Uncertainty sampling**: Train a single learner and then queries the unlabeled instance on which the learner has the least confidence.
    - ✓ **Query-by-committee (QBC):** Generate multiple learners and then query the unlabeled instance on which the learners disagree the most.
  - **Representativeness**: How well an instance helps represent the structure of input patterns, usually by **clustering**.

Semi-Supervised Learning

Exploit unlabeled data without querying human experts



Semi-Supervised Learning

- Two basic assumptions:
  - ✓**Cluster assumption**: Data have inherent cluster structure
    - ➔ Instances in the same cluster have the same class label.
  - ✓**Manifold assumption**: Data lie on a manifold ➔ Nearby instances have similar predictions.
- **Essence**: Similar data points should have similar outputs, whereas unlabeled data can be helpful to disclose which data points are similar.

Semi-Supervised Learning

Four major categories:

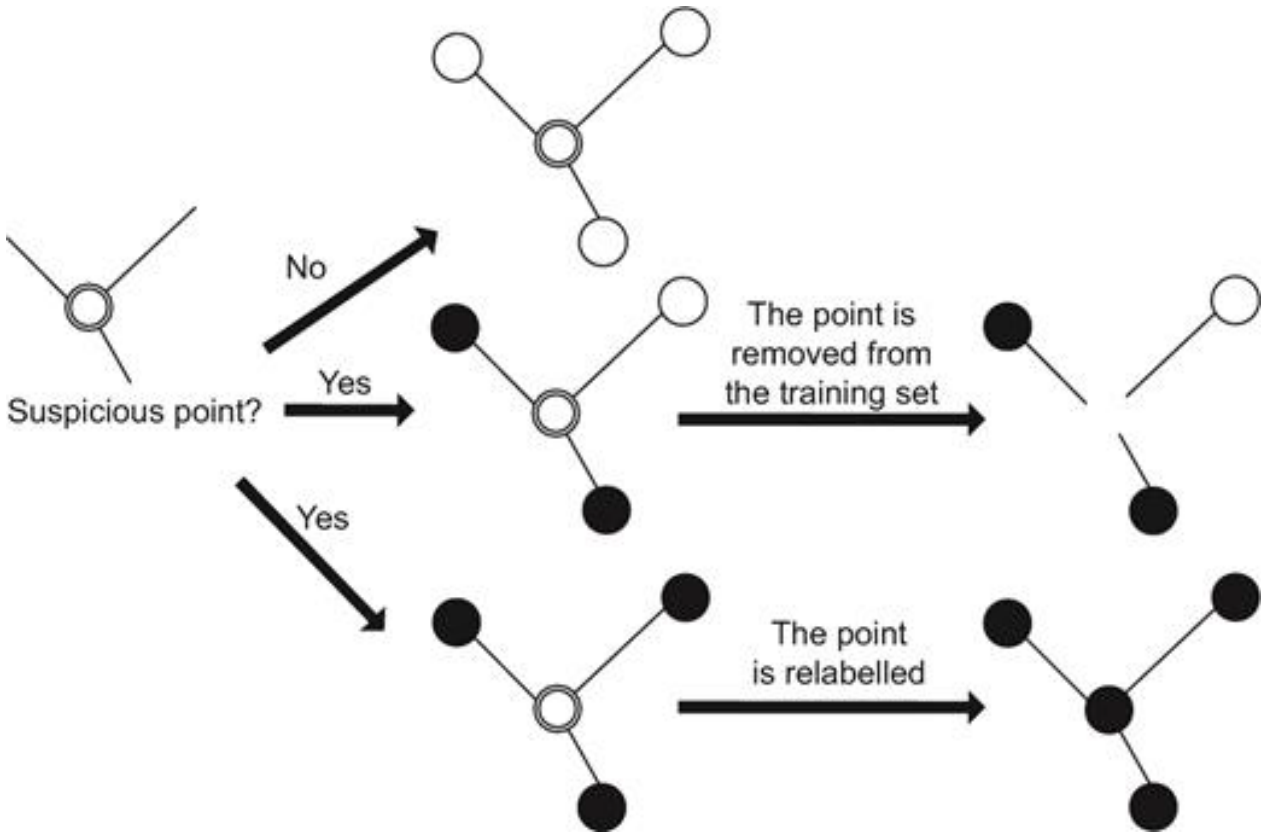
- ✓ **Generative methods**: Both labeled and unlabeled data are generated from the same inherent model, estimated by, e.g., EM.
- ✓ **Graph-based methods**: Construct a graph, where the nodes correspond to training instances and the edges to the relation (similarity) between them; then propagate label information on the graph according to some criteria.
- ✓ **Low-density separation methods**: The classification boundary goes across the less-dense regions in input space, e.g., **S3VM**.
- ✓ **Disagreement-based methods**: Generate multiple learners and let them collaborate to exploit unlabeled data, e.g., **co-training**.

Inexact Supervision

- **Setting**: Some supervision information is given, but not as exact as desired.
- **Typical scenario**: Only coarse-grained label information is available.
- **Multi-instance learning**: Learn $f: X \mapsto Y$ from a training data set $D = \{(X_1, y_1), \ldots, (X_m, y_m)$, where $X_i = \{x_{i1}, \ldots, x_{i,mi}\} \subseteq X$ is called a bag, $x_{ij} \in X$ $(j \in \{1, \ldots, m_i\})$ is an instance, m_i is the number of instances in X_i, and $y_i \in Y = \{Y, N\}$. X_i is a positive bag, i.e. $y_i = Y$, if there exists x_{ip} that is positive, while $p \in \{1, \ldots, m_i\}$ is unknown. The goal is to predict labels for unseen bags.
- Almost all supervised learning algorithms have their multi-instance peers
- Most algorithms attempt to adapt single-instance supervised learning algorithms to the multi-instance representation, mainly by shifting their focus from the discrimination on instances to the discrimination on bags.

Inaccurate Supervision

- **Setting**: The supervision information is not always groundtruth, i.e., some label information may suffer from errors.
- **Typical scenario**: Learning with label noise.
- **Basic idea**: Identify the potentially mislabeled examples, and then try to make some correction, e.g., data-editing.



Outline

- Weakly Supervised Learning
- **Active Learning**
- Active Learning for Classification
- Active Learning for Regression
- Deep Active Learning
- Applications
- Conclusions

Motivation

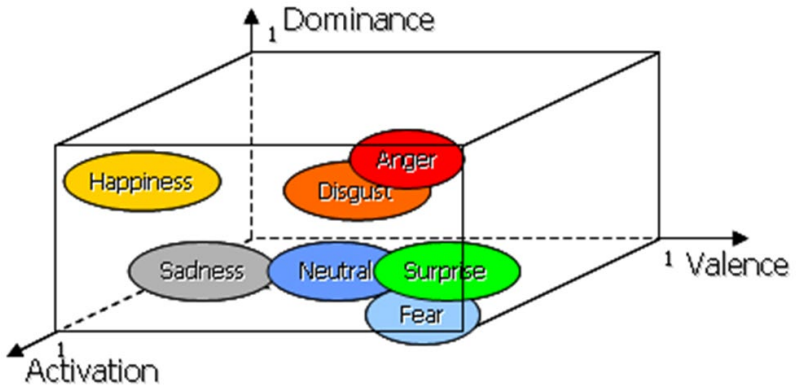
- Better data is often more useful than simply more data:
  Quality over **quantity**

- Data collection may be **expensive**
  - Cost of time and materials for an experiment
  - Cheap vs. expensive data: Raw images vs. annotated images

- Want to collect **best data** at **minimal cost**

Example 1: Affective Computing

- Emotions can be represented in the 2D space of arousal and valence, or in the 3D space of arousal, valence, and dominance.
- Emotions are very subjective, subtle, and uncertain.

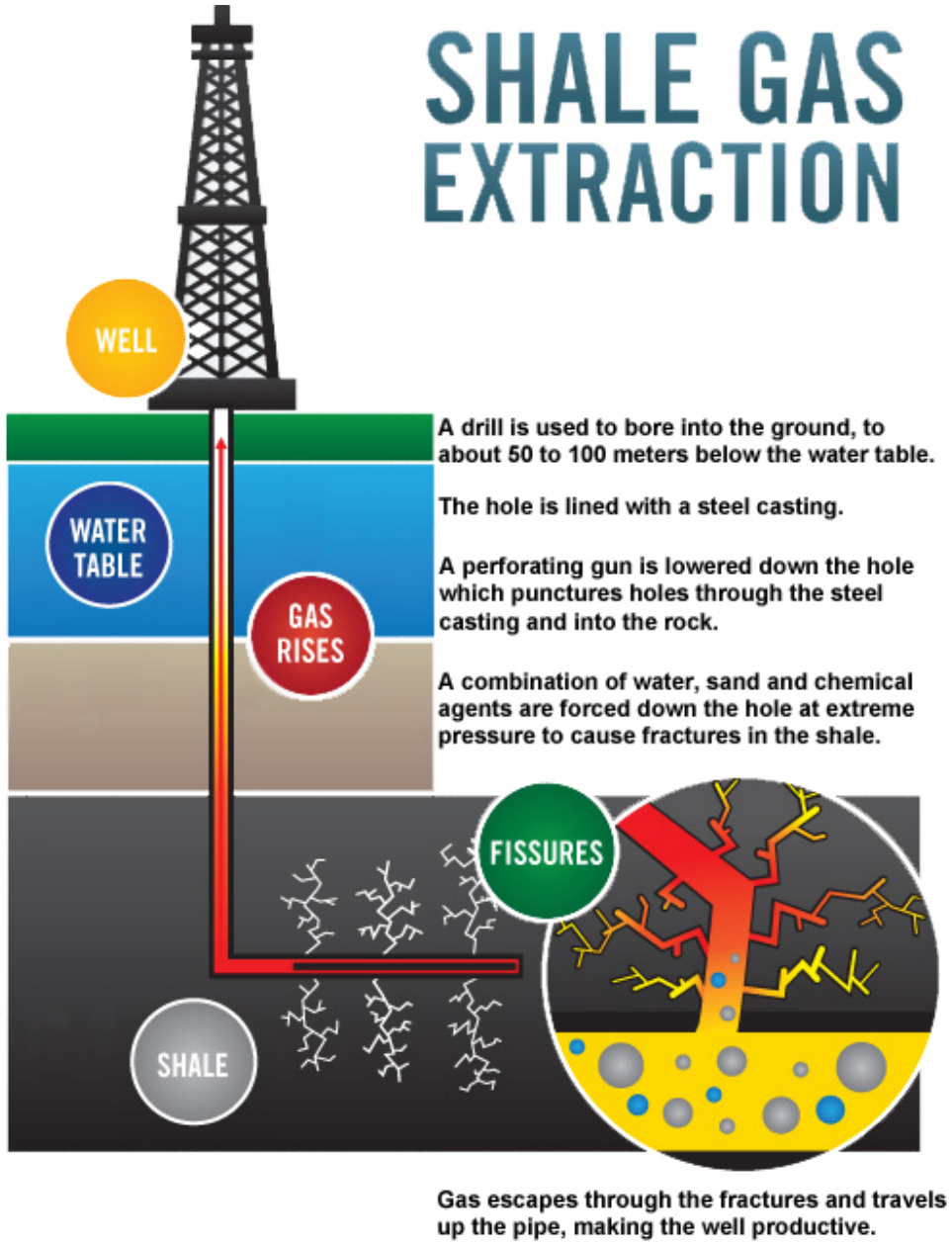


- Multiple assessors are needed to obtain the groundtruth emotion values for each affective sample (video, audio, image, physiological signal, etc).
  - ✓ 14-16 assessors were used to evaluate each video clip in the DEAP dataset
  - ✓ 6-17 assessors for each utterance in the VAM spontaneous speech corpus
  - ✓ 110+ assessors for each sound in the IADS-2 dataset
- Very time-consuming and labor-intensive.
- **Challenge**: How to optimally select the affective samples to label so that an accurate regression model can be built with the minimum cost?

Example 2: Oilfield Fracturing Optimization

180-day post-fracturing cumulative oil production prediction in enhanced oil recovery in the oil and gas industry:

- The inputs (fracturing parameters of an oil well, such as its location, length of perforations, number of zones/holes, volumes of injected slurry/water/sand, etc.) can be easily recorded during the fracturing operation.
- To get the groundtruth output (180-day post-fracturing cumulative oil production), one has to wait for at least 180 days.



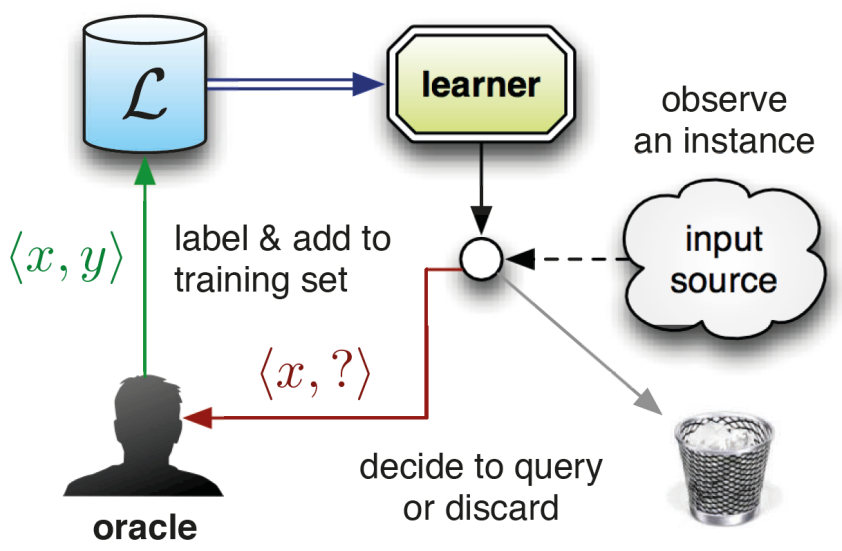
Active Learning (AL)

- **Setup**: Given existing knowledge, want to choose where to collect more data
  - ✓ Access to cheap unlabeled points
  - ✓ Make a query to obtain expensive label
  - ✓ Want to find labels that are most useful

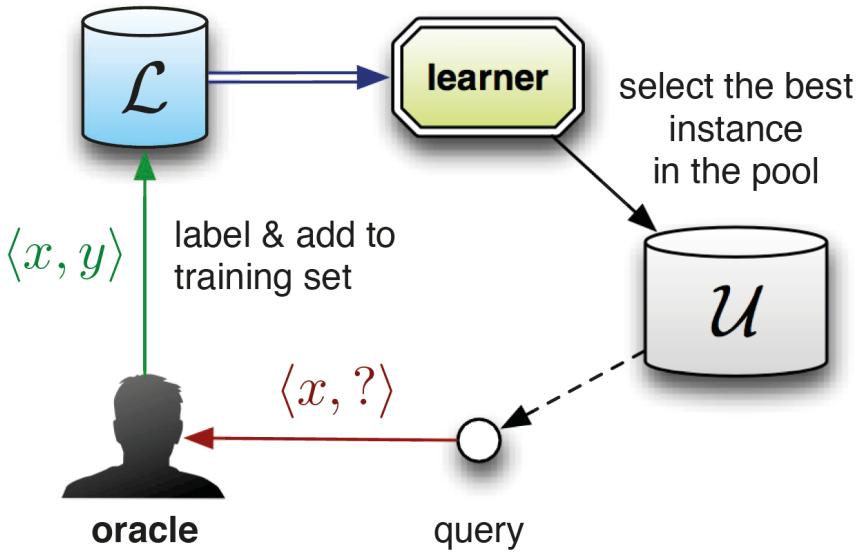
- **Output**: Classifier/regression model trained on less labeled data

Online and Offline AL

- **Online AL (streaming AL, selective sampling):** The learner decides whether to query or discard items from a stream
- **Offline AL (pool-based AL):** Queries the most useful instances from a large pool of available unlabeled data U.



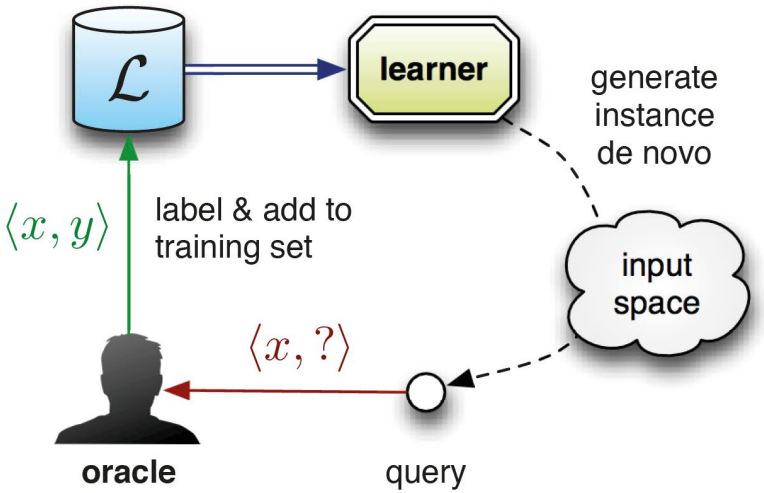
(a) selective sampling



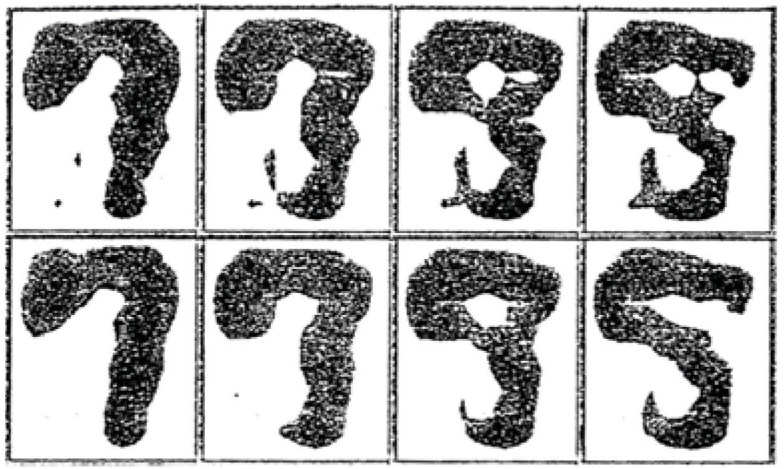
(b) pool-based sampling

Query Synthesis (Membership Queries)

- **Setup:** The test input distribution is known; Training input samples at any desired locations can be queried.
- **Goal:** Find the optimal training input density to generate the training input samples.



(a) query synthesis



(b) an example from handwriting recognition

Can result in awkward and uninterpretable queries, e.g., images generated by a neural network attempting to learn how to recognize handwritten digits.

Outline

- Weakly Supervised Learning
- Active Learning
- **Active Learning for Classification**
- Active Learning for Regression
- Deep Active Learning
- Applications
- Conclusions

Uncertainty Sampling

- Perhaps the simplest and most commonly used
- Query the most uncertain instances to label
- Focuses on the **informativeness**

Binary classification (maximum uncertain): $x^* = \arg\min_x |P(\hat{y}|x) - 0.5|$

Multi-class classification:

- ✓ Least confident: $x^* = \arg\max_x 1 - P(\hat{y}|x)$
- ✓ Margin sampling: $x^* = \arg\min_x P(\hat{y}_1|x) - P(\hat{y}_2|x)$
- ✓ Maximum entropy: $x^* = \arg\max_x -\sum_i P(y_i|x) \log P(y_i|x)$

Query-by-Committee (QBC)

- Popular for both classification and regression
- Focuses on the **informativeness**

- **Basic idea:**
  1. Build a committee of learners from existing labeled data
  2. Select the unlabeled sample on which the committee disagree the most to label

- **Disagreement measures:**
- Vote entropy:

$$x^* = \arg\max_x -\sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}$$

- Kullback-Leibler (KL) divergence:

$$x^* = \arg\max_x \frac{1}{C} \sum_{c=1}^{C} \sum_i P_c(y_i|c) \log \frac{P_c(y_i|x)}{\bar{P}(y_i|x)}$$

Expected Model Change

- Popular for classification, regression, and ranking
- Focuses on the **informativeness**
- **Basic idea:** Select the instance that would make the greatest (expected) change to the current model

$$x^* = \arg\max_x \sum_i P(y_i|x)\|\nabla_\theta(\mathbf{L} \cup \langle x, y_i\rangle)\| \approx \arg\max_x \sum_i P(y_i|x)\|\nabla_\theta(\langle x, y_i\rangle)\|$$

where $\|\cdot\|$ is the Euclidean norm of each resulting gradient vector. At query time, $\nabla\ell_\theta(\mathbf{L})$ should be nearly zero since ℓ converged at the previous round of training. Thus, we can approximate $\nabla_\theta(\mathbf{L} \cup \langle x, y_i\rangle) \approx \nabla_\theta(\langle x, y_i\rangle)$ for computational efficiency, because training instances are usually assumed to be independent.

Expected Error Reduction

- Focuses on the **informativeness**
- **Basic idea:** Select the instance that reduces the (expected) generalization error the most, or, minimize the expected 0/1 loss:

$$x^* = \arg\min_x \sum_i P_\theta(y_i|x) \left(\sum_{u=1}^{U} 1 - P_{\theta^{+\langle x, y_i \rangle}}(\hat{y}|x_u) \right)$$

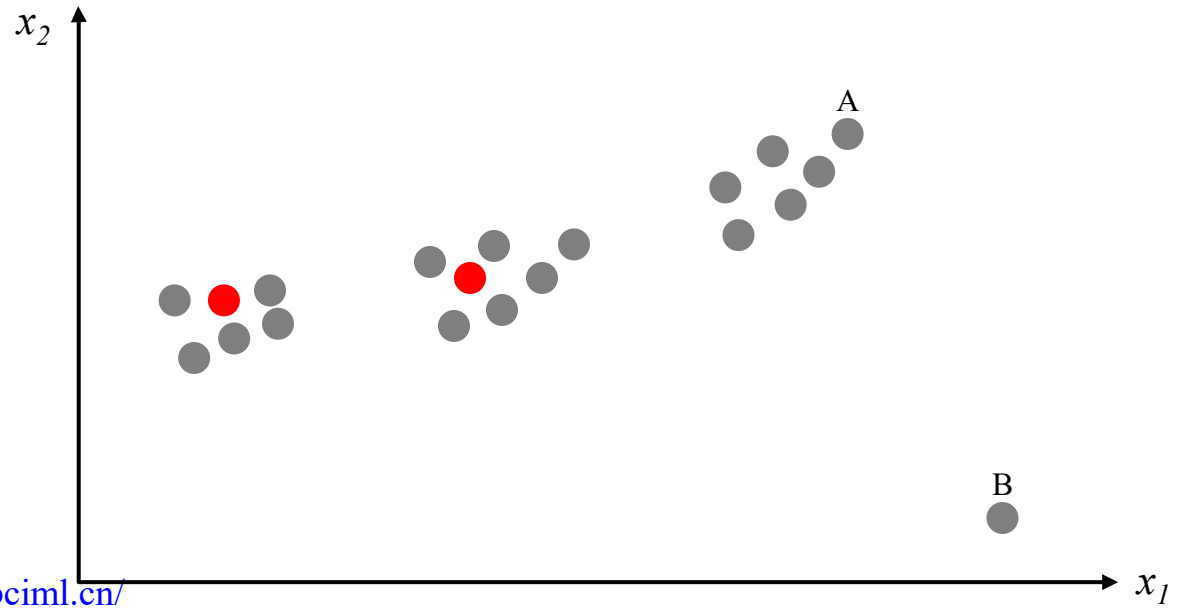
where $\theta^{+\langle x, y_i \rangle}$ refers to the the new model after it has been re-trained with the training tuple $\langle x, y_i \rangle$ added to $\mathbf{L}$. The computational cost is high.

Outline

- Weakly Supervised Learning
- Active Learning
- Active Learning for Classification
- **Active Learning for Regression**
- Deep Active Learning
- Applications
- Conclusions

Criteria for Pool-based ALR

- **Informativeness**: Measured by uncertainty (entropy, distance to the decision boundary, confidence of the prediction, etc.), expected model change, expected error reduction, etc.
- **Representativeness**: Evaluated by the number of samples that are similar or close to a target sample (or its density)
- **Diversity**: The selected sample should scatter across the full feature space, instead of concentrating in a small local region



Query-by-Committee (QBC)

- Popular for both classification and regression
- Focuses on the **informativeness**
- **Basic idea:**
  1. Build a committee of learners from existing labeled data
  2. Select the unlabeled sample on which the committee disagree the most to label
- **For regression:** Select the sample which has the maximal variance to label

$$\sigma_n = \frac{1}{P}\sum_{p=1}^{P}\left(y_n^p - \bar{y}_n\right)^2, \quad n = 1, ..., N$$

Expected Model Change Maximization (EMCM)

- Popular for classification, regression, and ranking
- Focuses on the **informativeness**
- **Basic idea:**
  1. Build a committee of learners from existing labeled data
  2. Select the unlabeled sample with the maximal expected model change to label
- **For linear regression:**

$$g(\mathbf{x}_n) = \frac{1}{P}\sum_{p=1}^{P} \left\| (y_n^p - \bar{y}_n)\mathbf{x}_n \right\|, \quad n = 1, ..., N$$

W. Cai, Y. Zhang, and J. Zhou, "Maximizing expected model change for active learning in regression," IEEE 13th Int. Conf. Data Mining, Dallas, TX, Dec. 2013, pp. 51–60.

Transductive Experimental Design (TED)

- Popular for classification and regression
- Focuses on the **informativeness and representative**
- **Basic idea:**
  Select a batch of unlabeled samples with the minimal estimation confidence, which also maximally reconstruct the whole dataset, to label.
- **For regularized linear regression:**

$$\max_{\mathbf{X}} \quad Tr\left[\mathbf{V}\mathbf{X}^{\top}(\mathbf{X}\mathbf{X}^{\top}+\mu\mathbf{I})^{-1}\mathbf{X}\mathbf{V}^{\top}\right]$$
$$subject\ to \quad \mathbf{X}\subset\mathbf{V}, |\mathbf{X}|=m$$

K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in Proc. Int'l Conf. on Machine learning, Pittsburgh, PA, Jun. 2006, pp. 1081–1088.

Clustering-based AL

- **Passive** sampling approach
- Focuses on the **representative**
- **Basic idea:**
  1. Partition the dataset into k clusters using clustering approaches, e.g., k -means clustering, k -medoid clustering.
  2. Select the samples closest to the cluster centroids to label.

[1] J. Kang, K. R. Ryu, and H.-C. Kwon, "Using cluster-based sampling to select initial training set for active learning in text classification," in Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining, Sydney, Australia, May 2004, pp. 384–388.
[2] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in Proc. Int'l Conf. on Machine Learning, Banff, Canada, Jul. 2004, p. 79.

Greedy Sampling (GS)

- **Passive** sampling approach
- Focuses on the **diversity**

- **Basic idea:**

1. For each of the N unlabeled samples, compute its distances to all M labeled samples, $d_{nm}, n = 1, ..., N; m = 1, ..., M$
2. Compute $d_n = \min_m d_{nm}, n = 1, ..., N$
3. Select the sample with the maximal d_n to label

H. Yu and S. Kim, "Passive sampling for regression," in Proc. IEEE Int. Conf. Syst., Man, Cybern., Sydney, NSW, Australia, Dec. 2010, pp. 1151–1156.

Limitations of Previous ALR Approaches

- Only consider informativeness, representative, or diversity, but not informativeness, diversity and representativeness **simultaneously**
- The first a few samples are usually initialized **randomly**

Representativeness-Diversity (RD)

1. **Representativeness** and **diversity** in **initialization**:

   i. Perform k-means clustering on all samples, where k equals the size of the initial labeled samples.

   ii. For each cluster, select the sample closest to its centroid for labeling.

2. **Representativeness** and **diversity** in the n^{th} **iteration** of the sequential AL:

   i. Perform k-means clustering on all samples, where $k=n$.

   ii. Identify the largest cluster that does not contain a labeled sample, and select the sample closest to its centroid to label.

D. Wu, "Pool-based sequential active learning for regression," *IEEE Trans. on Neural Networks and Learning Systems*, 30(5):1348-1359, 2019.

Integrate RD with an Existing ALR Approach

1. Representativeness and diversity in **initialization**:
   1) Perform k-means clustering on all samples, where k equals the size of the initial labeled samples.
   2) For each cluster, select the sample closest to its centroid for labeling.

2. Representativeness, diversity, and/or **informativeness** in the n^{th} **iteration** of sequential AL:
   1) Perform k-means clustering on all samples, where $k=n$.
   2) Identify the largest cluster that does not contain a labeled sample
   3) Use **QBC** or **EMCM** or **GS** to select a sample from the above cluster for labeling.

Pseudo-Code

Algorithm 2: The proposed RD ALR algorithm, and its variations.

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^d$; M, the maximum number of labeled samples to query

Output: The regression model $f(\mathbf{x})$.

// Initialize d labeled samples

Perform k-means clustering on $\{\mathbf{x}_n\}_{n=1}^N$, where $k = d$;

Select from each cluster the sample closest to its centroid, and query for its label;

// End initialization

for $m = d + 1, ..., M$ **do**

  Perform k-means clustering on $\{\mathbf{x}_n\}_{n=1}^N$, where $k = m$;

  Identify the largest cluster that does not already contain a labeled sample;

  Option 1: Select the sample closest to the cluster centroid for labeling;

  Option 2: Use QBC (Section II-B) to select a sample from the cluster for labeling;

  Option 3: Use EMCM (Section II-C) to select a sample from the cluster for labeling;

  Option 4: Use GS (Section II-D) to select a sample from the cluster for labeling;

end

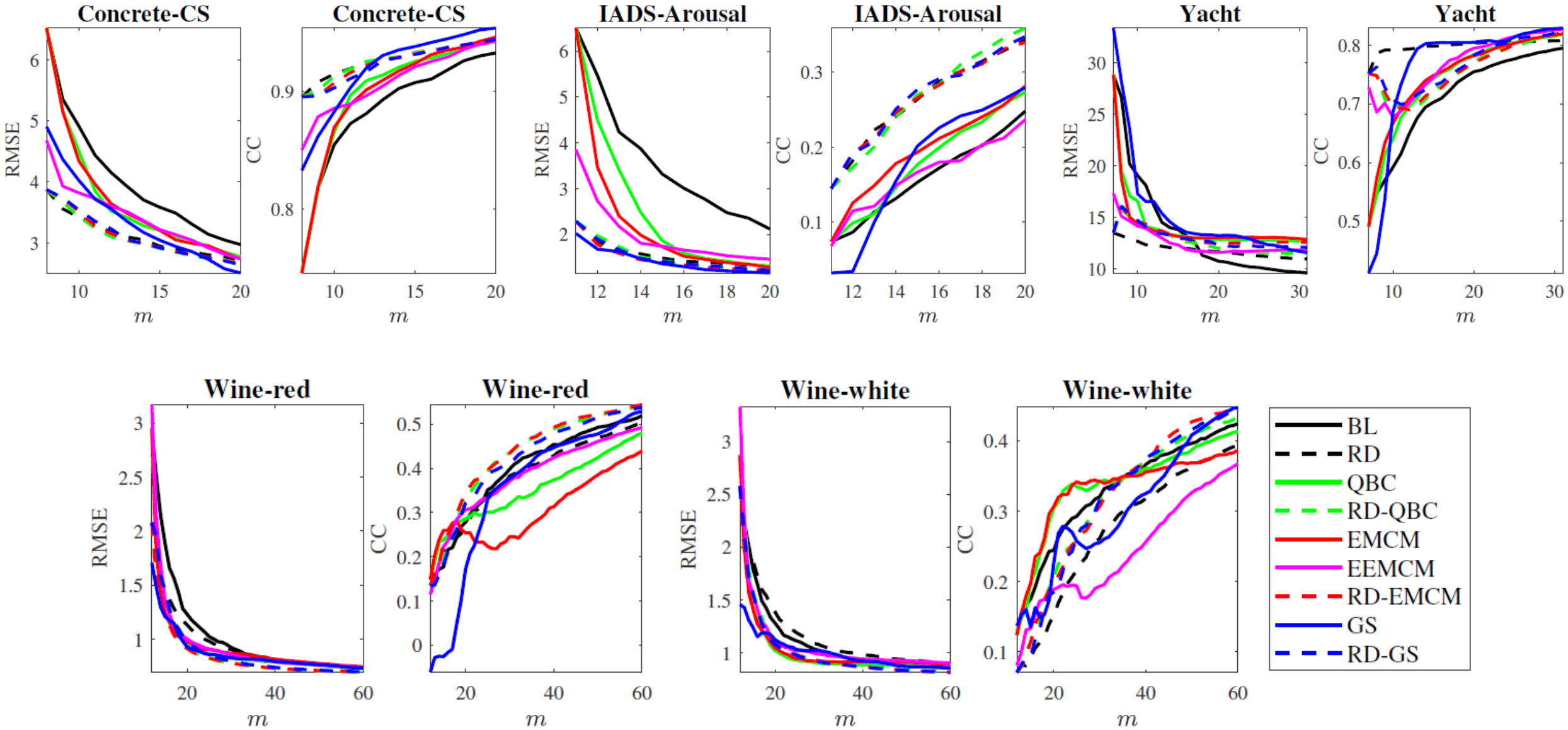
Construct the regression model $f(\mathbf{x})$ from the M labeled samples.

Datasets

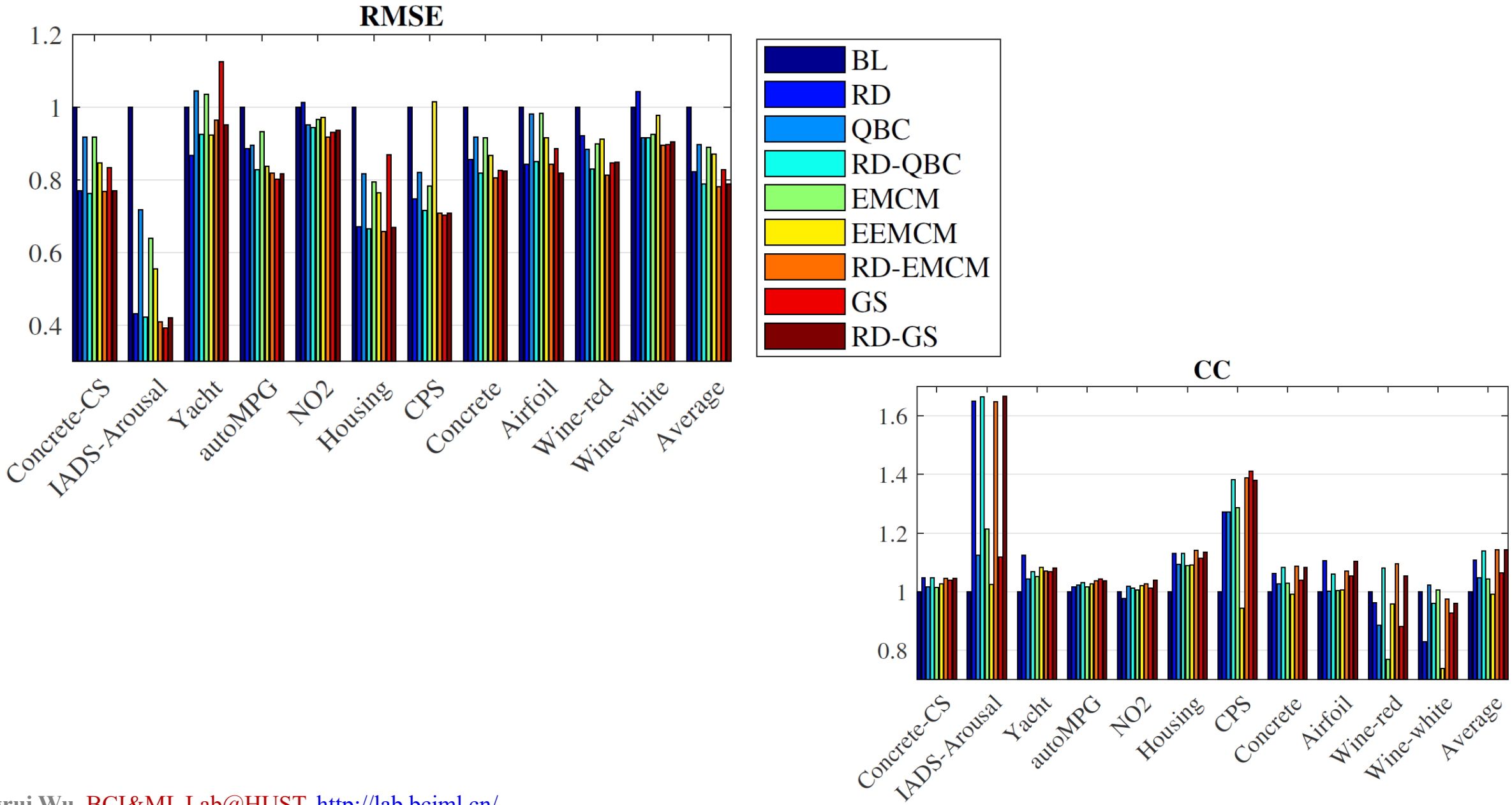
Dataset	Source	No. of samples	No. of raw features	No. of numerical features	No. of categorical features	No. of total features
Concrete-CS[1]	UCI	103	7	7	0	7
IADS-Arousal[2]	UFL	167	10	10	0	10
Yacht[3]	UCI	308	6	6	0	6
autoMPG[4]	UCI	392	7	6	1	9
NO2[5]	StatLib	500	7	7	0	7
Housing[6]	UCI	506	13	13	0	13
CPS[7]	StatLib	534	11	8	3	19
Concrete[8]	UCI	1030	8	8	0	8
Airfoil[9]	UCI	1503	5	5	0	5
Wine-red[10]	UCI	1599	11	11	0	11
Wine-white[10]	UCI	4898	11	11	0	11

[1] https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test
[2] http://csea.phhp.ufl.edu/media.html#midmedia
[3] https://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics
[4] https://archive.ics.uci.edu/ml/datasets/auto+mpg
[5] http://lib.stat.cmu.edu/datasets/
[6] https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
[7] http://lib.stat.cmu.edu/datasets/CPS_85_Wages
[8] https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength
[9] https://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise
[10] https://archive.ics.uci.edu/ml/datasets/Wine+Quality

Experimental Results



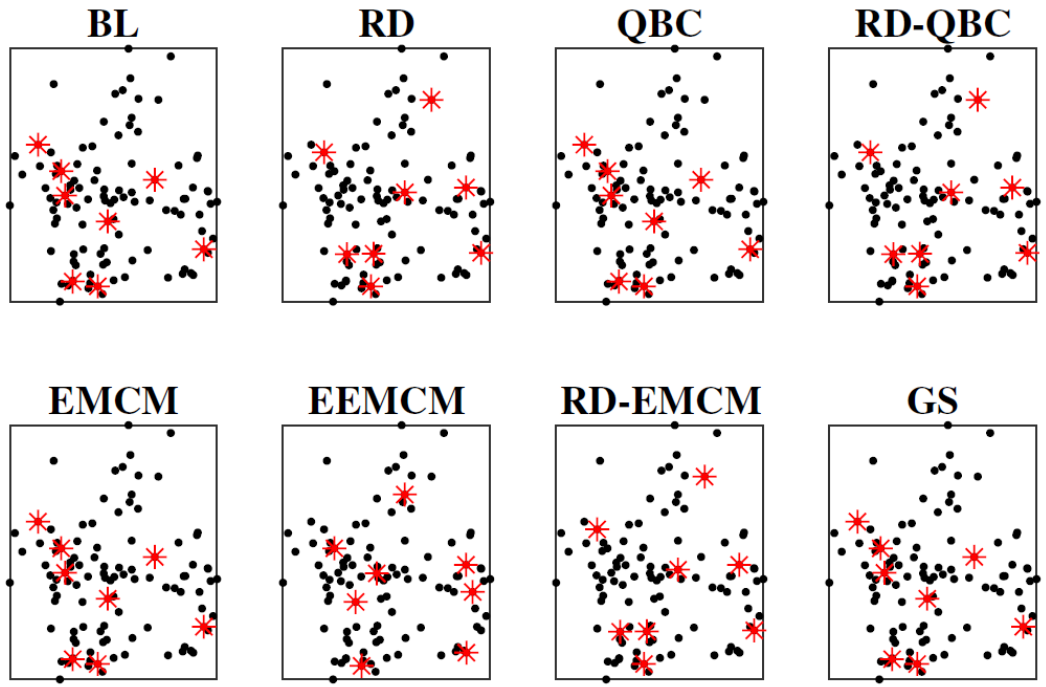
Area Under the Curve (AUC)



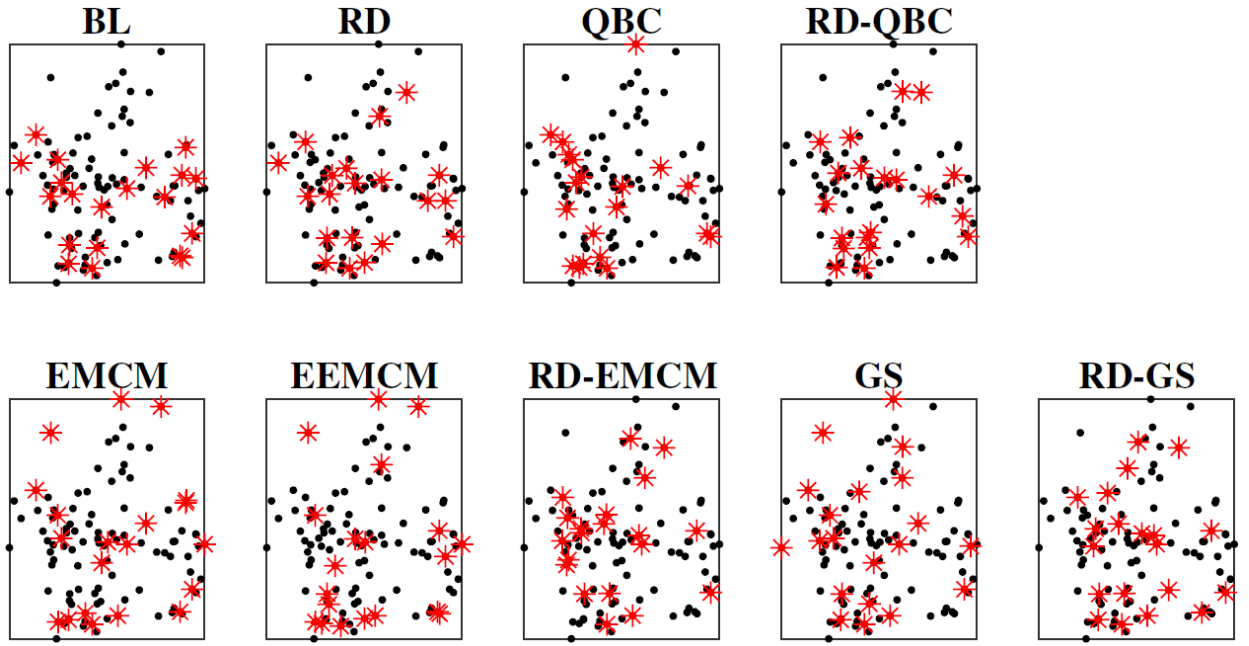
Ranks of the Algorithms

	Dataset	BL	QBC	EMCM	EEMCM	GS	RD	RD-QBC	RD-EMCM	RD-GS
RMSE	Concrete-CS	9	8	7	6	5	1	4	2	3
	IADS-Arousal	9	8	7	6	1	5	4	2	3
	Yacht	5	8	7	3	9	1	2	6	4
	autoMPG	9	7	8	5	1	6	4	2	3
	NO2	9	5	7	6	3	8	4	1	2
	Housing	9	7	6	5	8	3	2	1	4
	CPS	8	7	6	9	1	5	3	4	2
	Concrete	9	8	7	6	4	5	3	1	2
	Airfoil	9	3	7	6	8	1	2	4	5
	Wine-red	9	4	6	7	5	8	2	1	3
	Wine-white	8	4	3	7	1	9	6	2	5
	Average	**9**	**7**	**8**	**6**	**4**	**5**	**2**	**1**	**2**
CC	Concrete-CS	9	8	7	6	5	1	4	2	3
	IAPS-Arousal	9	5	6	7	8	4	3	1	2
	Yacht	9	8	7	3	6	1	5	4	2
	autoMPG	9	6	7	5	1	8	4	2	3
	NO2	8	5	7	4	6	9	3	2	1
	Housing	9	8	7	6	5	4	3	1	2
	CPS	8	6	7	9	1	5	2	3	4
	Concrete	9	6	7	8	5	4	3	2	1
	Airfoil	9	5	7	6	8	1	2	3	4
	Wine-red	4	7	9	6	8	5	2	1	3
	Wine-white	2	1	3	9	6	8	7	5	4
	Average	**9**	**6**	**8**	**7**	**5**	**4**	**3**	**1**	**2**

Visualization



Initial 7 samples



Final 20 samples

iRDM: Further Improves RD

- RD considers representativeness and diversity simultaneously, by choosing from each cluster a point closest to its centroid for labeling.
- This does not guarantee the global Representativeness-Diversity is maximized.
- Iterative Representativeness-Diversity Maximization (iRDM):

  1. Use RD to select M samples as the initial candidate set.
  2. Iteratively update each candidate to maximize the Representativeness-Diversity, until convergence.

Z. Liu, X. Jiang, H. Luo, W. Fang, J. Liu and D. Wu*, "Pool-Based Unsupervised Active Learning for Regression Using Iterative Representativeness-Diversity Maximization (iRDM)," Pattern Recognition Letters, 142:11-19, 2021.

iRDM: Iterative Update

Let the current candidate set be $\{\bar{\mathbf{x}}_m\}_{m=1}^{M}$ ($\bar{\mathbf{x}}_m$ is the $m$the sample in the candidate set, instead of the mth sample in the pool).

Let the candidate sample to be optimized be $\bar{\mathbf{x}}_m$, and its corresponding cluster be C_m.

Assume there are N_m samples in C_m. iRDM selects a better sample in C_m to replace $\bar{\mathbf{x}}_m$.

The representativeness of a sample $\mathbf{x}_n$ in C_m is the average distance between $\mathbf{x}_n$ and all remaining samples in C_m:

$$R(\mathbf{x}_n) = \frac{1}{N_m - 1} \times \sum_{\mathbf{x}_i \in C_m} ||\mathbf{x}_n - \mathbf{x}_i|| \tag{1}$$

The diversity of a sample $\mathbf{x}_n$ in C_m is computed similar to GSx:

$$D(\mathbf{x}_n) = \min_{\substack{i \in [1, M] \\ i \neq m}} ||\mathbf{x}_n - \mathbf{x}_i|| \tag{2}$$

$\bar{\mathbf{x}}_m$ in the candidate set is then replaced by:

$$\mathbf{x}_n^* = \arg \max_{n \in [1, N_C]} [D(\mathbf{x}_n) - R(\mathbf{x}_n)] \tag{3}$$

iRDM: Illustration and Pseudo-code

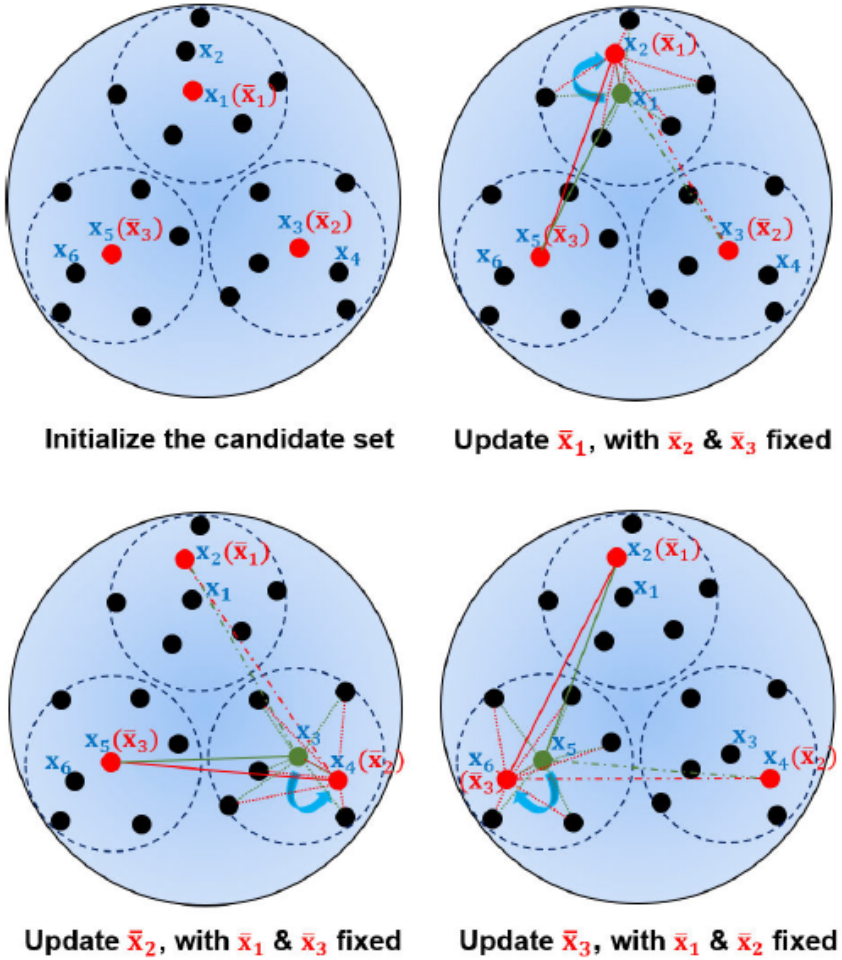


Fig. 2. Optimization of the candidate set in iRDM ($M = 3$, one iteration). The blue dashed circles represent the cluster boundaries. In each subplot, the dashed lines represent the distances of the sample under consideration to other samples in the same cluster, whose average is R in (1). The solid and dotted lines represent the distances of the sample under consideration to the $M-1$ fixed samples in the candidate set, among which the solid line is the shortest and is D in (2). The green and red dots are the samples before and after optimization, respectively.

Algorithm 1: The proposed iRDM algorithm.

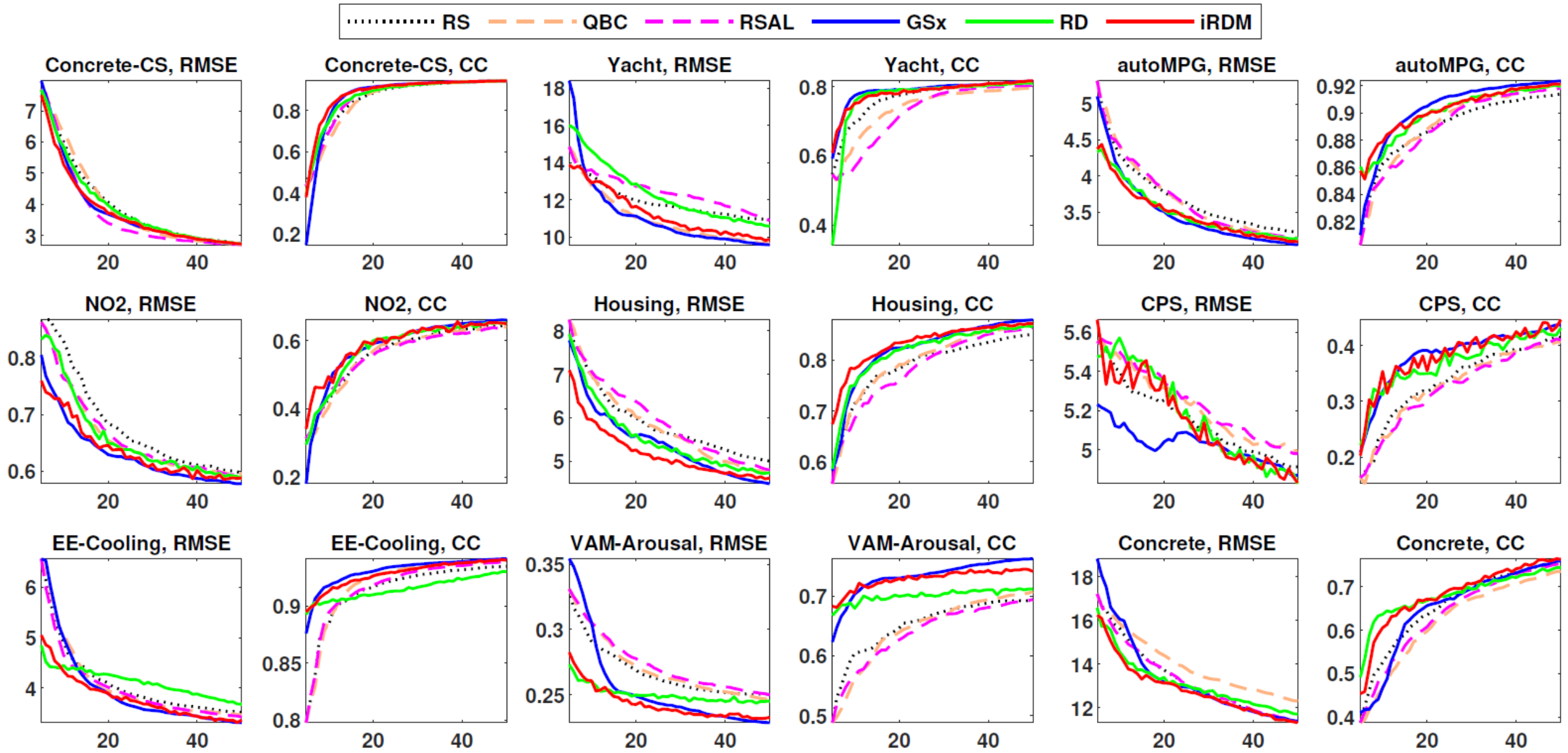
Input: A pool of N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
$c_{\max}$, the maximum number of iterations.
Output: $\{\bar{\mathbf{x}}_m\}_{m=1}^M$, the set of M samples to label.
Perform k-means clustering ($k = M$) on $\{\mathbf{x}_n\}_{n=1}^N$, and denote the clusters as $\{C_m\}_{m=1}^M$;
Select $\mathbf{x}_m$ as the sample closest to the centroid of C_m, $m = 1, ..., M$;
Sort the indices of the M samples in the candidate set and save them to the first row of matrix P;
Compute $R(\mathbf{x}_n)$ in (1) for $n = 1, ..., N$ and save them;
$c = 0$;
while $c < c_{\max}$ **do**
  Denote the M selected samples as $\{\bar{\mathbf{x}}_m\}_{m=1}^M$;
  for $m = 1, ..., M$ **do**
    Fix $\{\mathbf{x}_1, \ldots, \mathbf{x}_{m-1}, \mathbf{x}_{m+1}, \ldots, \mathbf{x}_M\}$;
    Compute $D(\mathbf{x}_n)$ in (2) for each sample in C_m;
    Identify $\mathbf{x}_n^*$ in (3);
    Set $\bar{\mathbf{x}}_m$ to $\mathbf{x}_n^*$;
  end
  Sort the indices of the M samples in the candidate set;
  if *the sorted indices of the M samples match any row in P* **then**
    Break;
  else
    Save the sorted indices of the M samples to the next row of P;
  end
  $c = c + 1$;
end

iRDM: Datasets in Experiments

Table 1. Summary of the 12 regression datasets.

Dataset	No. of samples	No. of raw features	No. of numerical features	No. of categorical features	No. of total features
Concrete-CS	103	7	7	0	7
Yacht	308	6	6	0	6
autoMPG	392	7	6	1	9
NO2	500	7	7	0	7
Housing	506	13	13	0	13
CPS	534	10	7	3	19
EE-Cooling	768	7	7	0	7
VAM-Arousal	947	46	46	0	46
Concrete	1,030	8	8	0	8
Airfoil	1,503	5	5	0	5
Wine-Red	1,599	11	11	0	11
Wine-White	4,898	11	11	0	11

iRDM: Experimental Results (RBF-SVR)



iRDM: Datasets in Experiments

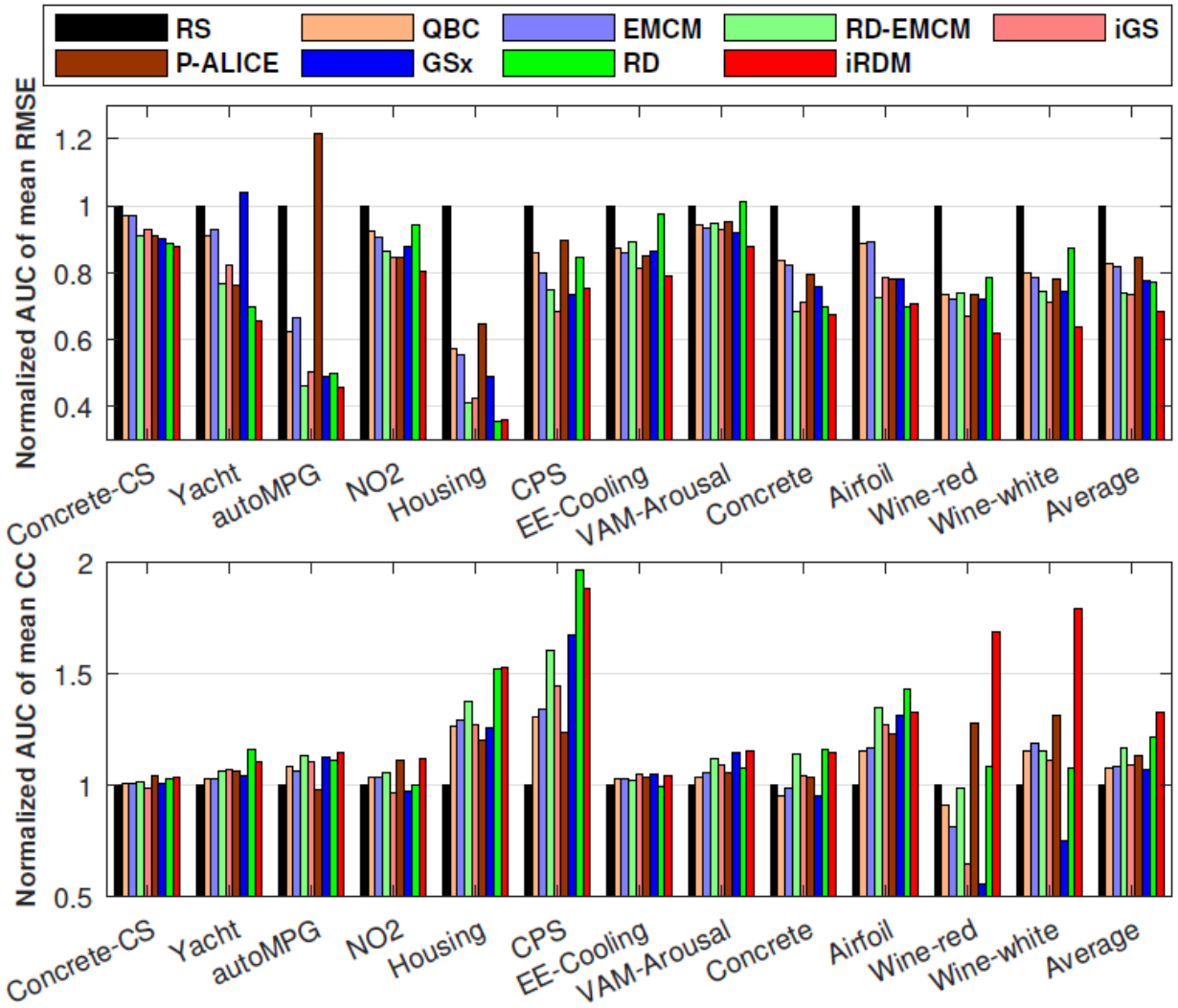


Fig. 4. Normalized AUCs ($M \in [5, 20]$) of the mean RMSEs and the mean CCs on the 12 datasets. RR ($r = 0.1$) was used.

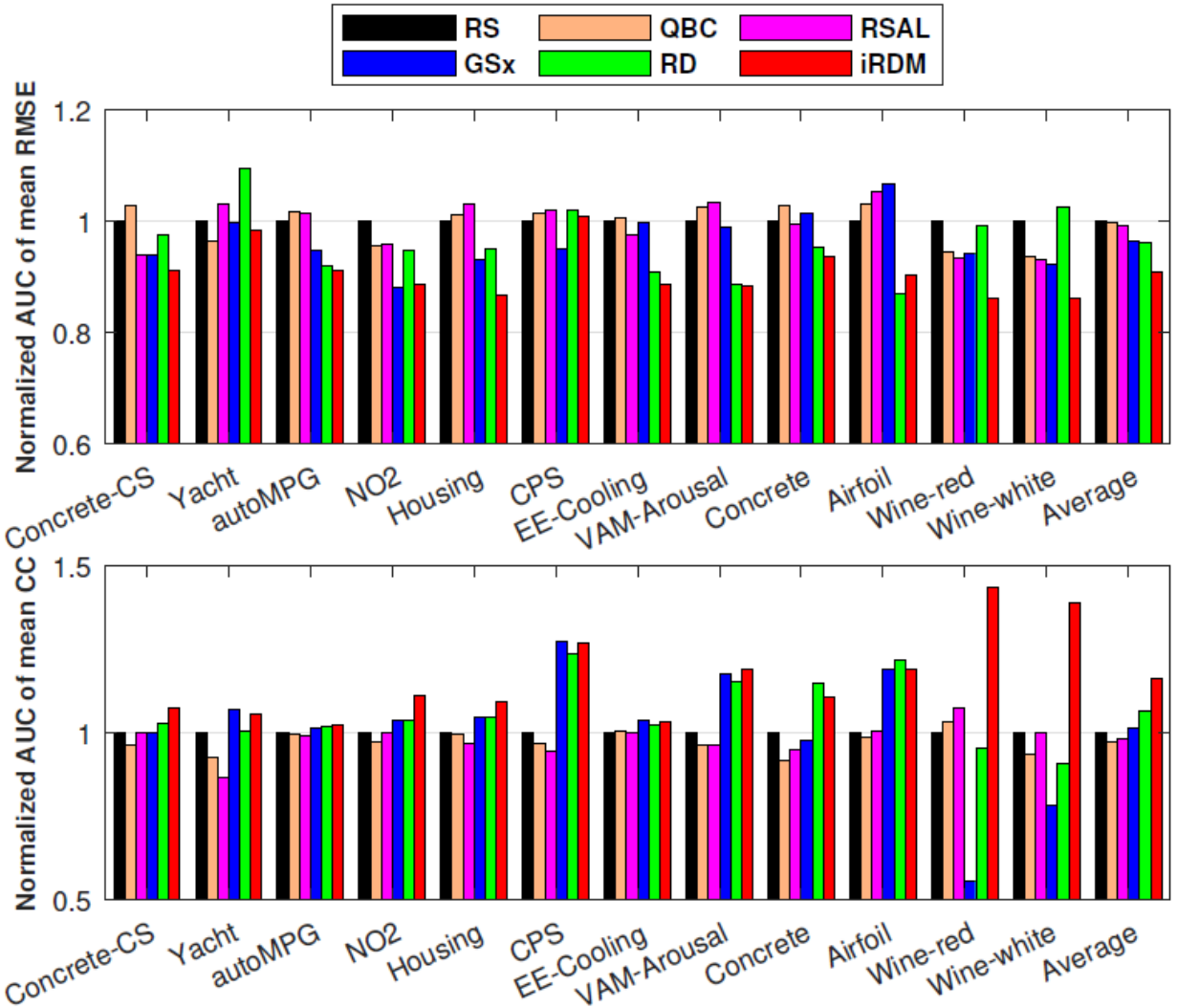


Fig. 5. Normalized AUCs ($M \in [5, 20]$) of the mean RMSEs and the mean CCs on the 12 datasets. RBF-SVR ($C = 50$, $\lambda = 0.01$) was used.

Greedy Sampling on the Input (GSx)

- **Passive** sampling approach
- Focuses on the **diversity**

- **Basic idea:**

1. For each of the N unlabeled samples, compute its distances to all M labeled samples, $d_{nm}, n = 1, ..., N; m = 1, ..., M$
2. Compute $d_n = \min_m d_{nm}, n = 1, ..., N$
3. Select the sample with the maximal d_n to label

Greedy Sampling on the Output (GSy)

- GSx achieves diversity in the **input** space.
- GSy achieves diversity in the **output** space:
  1. Selects the first a few samples using GSx to build an initial regression model
  2. In each subsequent iteration, a new sample located farthest away from all previously selected samples in the **output** space is selected
- GSx is **passive**; GSy is **NOT**!

D. Wu, C-T Lin and J. Huang, "Active Learning for Regression Using Greedy Sampling," *Information Sciences*, 474: 90-105, 2019.

GSy

Assume the first k ($k \geq K_0$) samples have already been labeled with outputs $\{y_m\}_{m=1}^k$, and a regression model $f(\mathbf{x})$ has been constructed.

For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, GSy computes first its distance to each of the k outputs:

$$d_{nm}^y = |f(\mathbf{x}_n) - y_m|, \quad m = 1, ..., k; n = k+1, ..., N$$

and d_n^y, the shortest distance from $f(\mathbf{x}_n)$ to $\{y_m\}_{m=1}^k$:

$$d_n^y = \min_m d_{nm}^y, \quad n = k+1, ..., N$$

and then selects the sample with the maximum d_n^y to label.

Improved Greedy Sampling (iGS) on Both the Inputs and Output

- **GSx** considers only the diversity in the **input** (feature) space, without taking feature selection/weighting into consideration.
- **GSy** considers only the diversity in the **output** (label) space, which implicitly considers feature selection /weighting. However, it may not be reliable, as the model is constructed from a very small number of samples.
- **iGS** combines GSx and GSy to ensure that we take feature selection/weighting into consideration, but can also avoid catastrophic failure if feature selection/weighting is misleading.

iGS

iGS selects the first a few samples using GSx to build an initial regression model, and then in each subsequent iteration a new sample located farthest away from all previously selected samples in both input and output spaces to achieve balanced diversity among the selected samples.

iGS uses GSx to select the first K_0 samples to label.

Assume the first k samples have already been labeled with labels $\{y_n\}_{n=1}^{k}$.

For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^{N}$, iGS computes first $d_{nm}^{\mathbf{x}}$ in GSx and d_{nm}^{y} in GSy, and $d_n^{\mathbf{x}y}$:

$$d_n^{\mathbf{x}y} = \min_m d_{nm}^{\mathbf{x}} d_{nm}^{y}, \quad n = k+1, ..., N$$

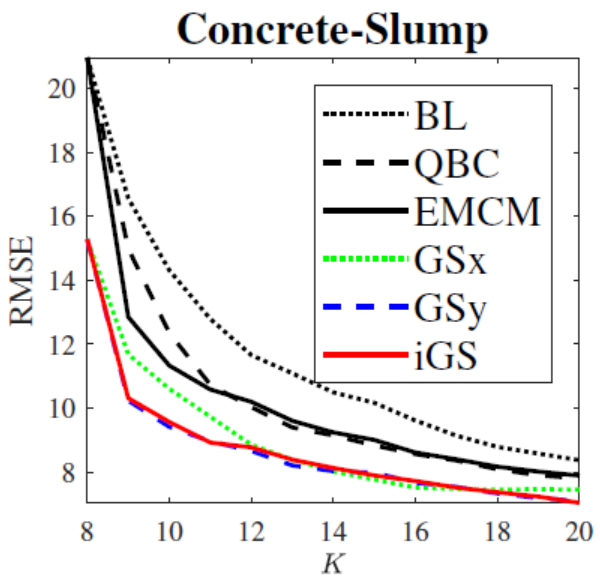
and then selects the sample with the maximum $d_n^{\mathbf{x}y}$ to label.

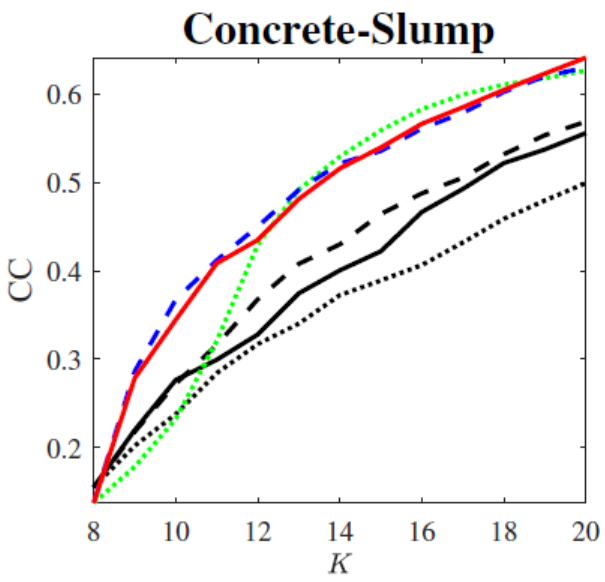
Datasets

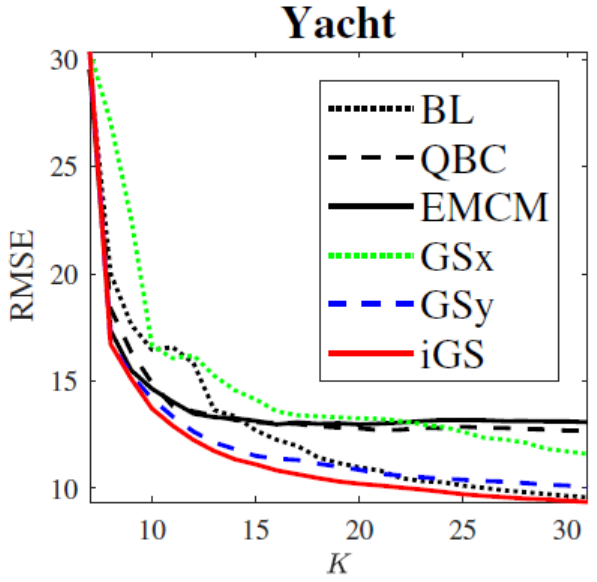
Table 1: Summary of the 10 UCI and CMU StatLib datasets.

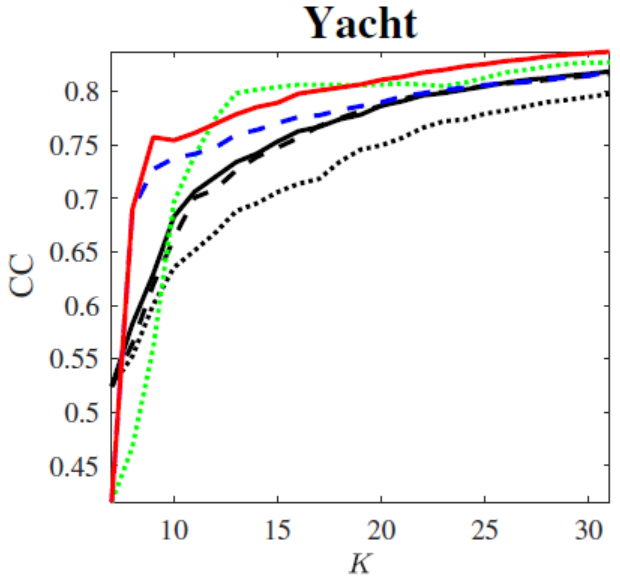
Dataset	Source	No. of samples	No. of raw features	No. of numerical features	No. of categorical features	No. of total features
Concrete-Slump[1]	UCI	103	7	7	0	7
Yacht[2]	UCI	308	6	6	0	6
autoMPG[3]	UCI	392	7	6	1	9
NO2[4]	StatLib	500	7	7	0	7
PM10[4]	StatLib	500	7	7	0	7
Housing[5]	UCI	506	13	13	0	13
CPS[6]	StatLib	534	11	8	3	19
Concrete[7]	UCI	1030	8	8	0	8
Wine-red[8]	UCI	1599	11	11	0	11
Wine-white[8]	UCI	4898	11	11	0	11

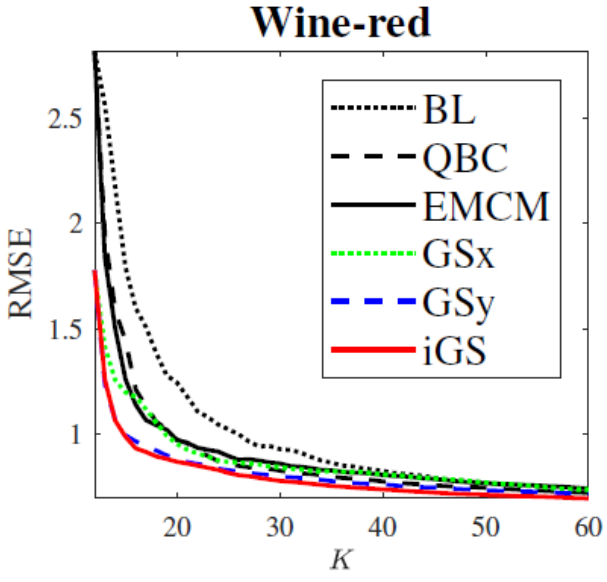
Experimental Results

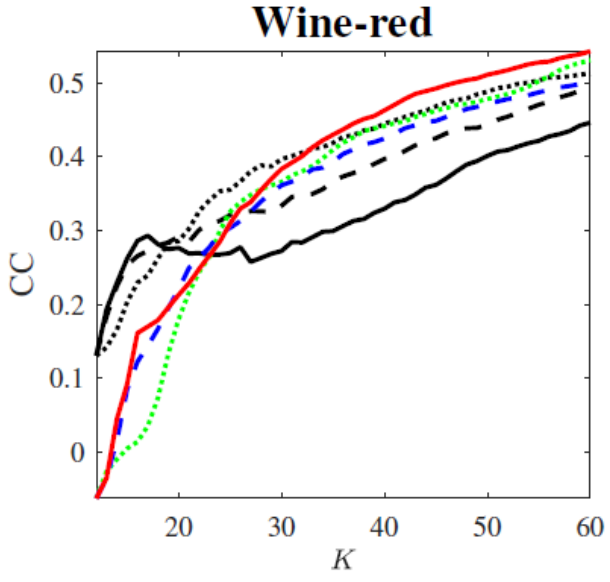


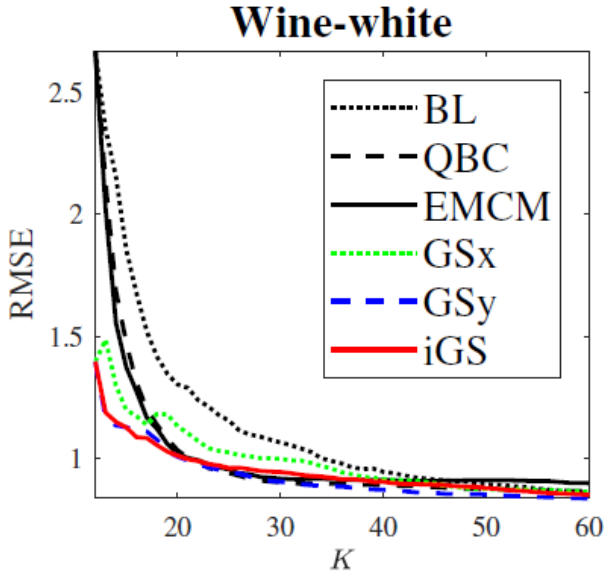


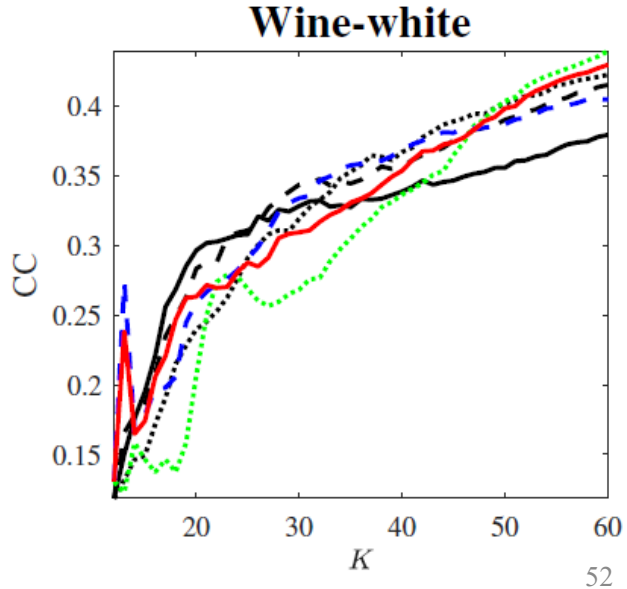




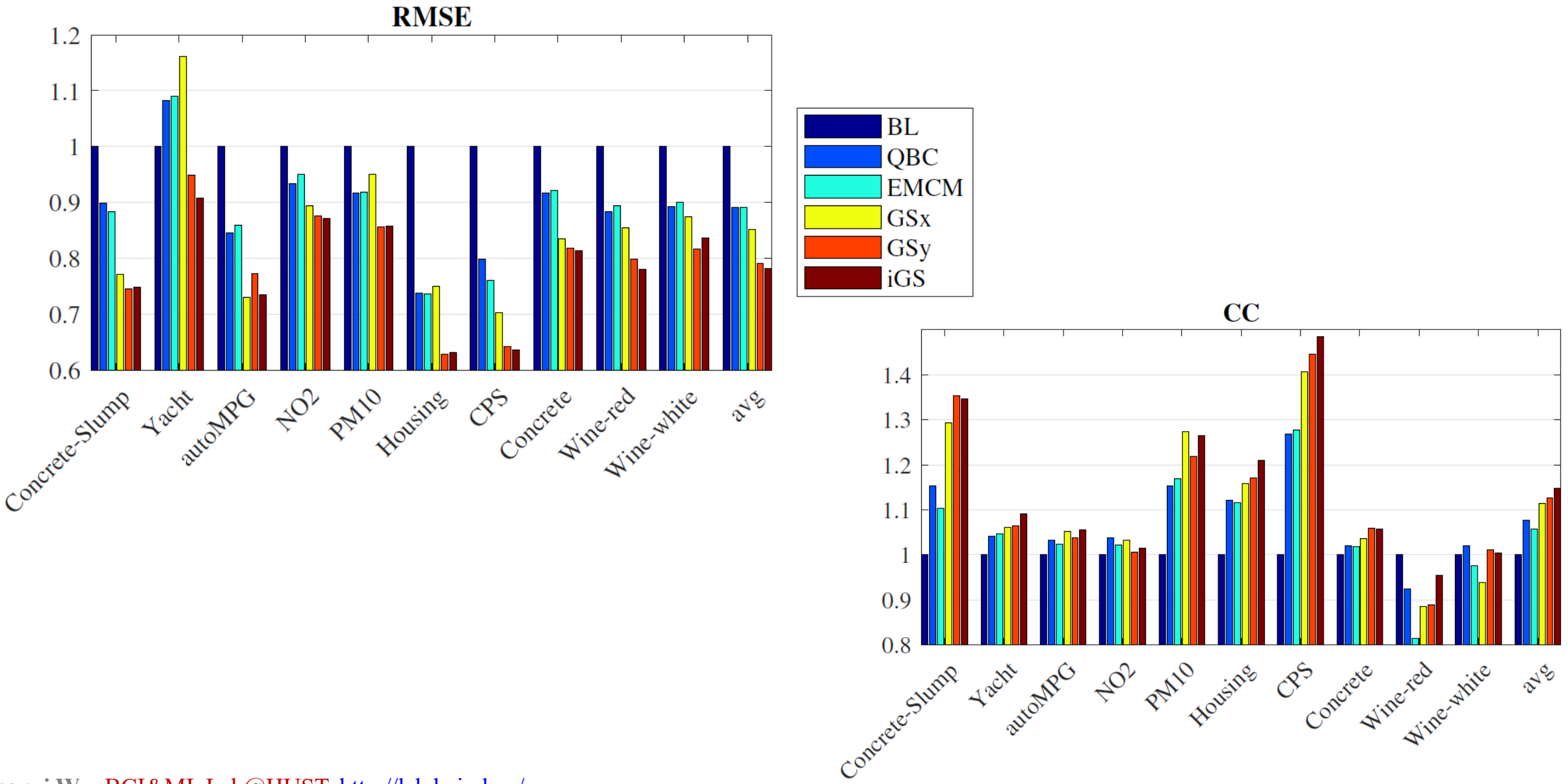








Area Under the Curve (AUC)



Normalized RMSEs and CCs

Table 2: Normalized RMSEs and CCs of the six approaches on the 10 datasets.

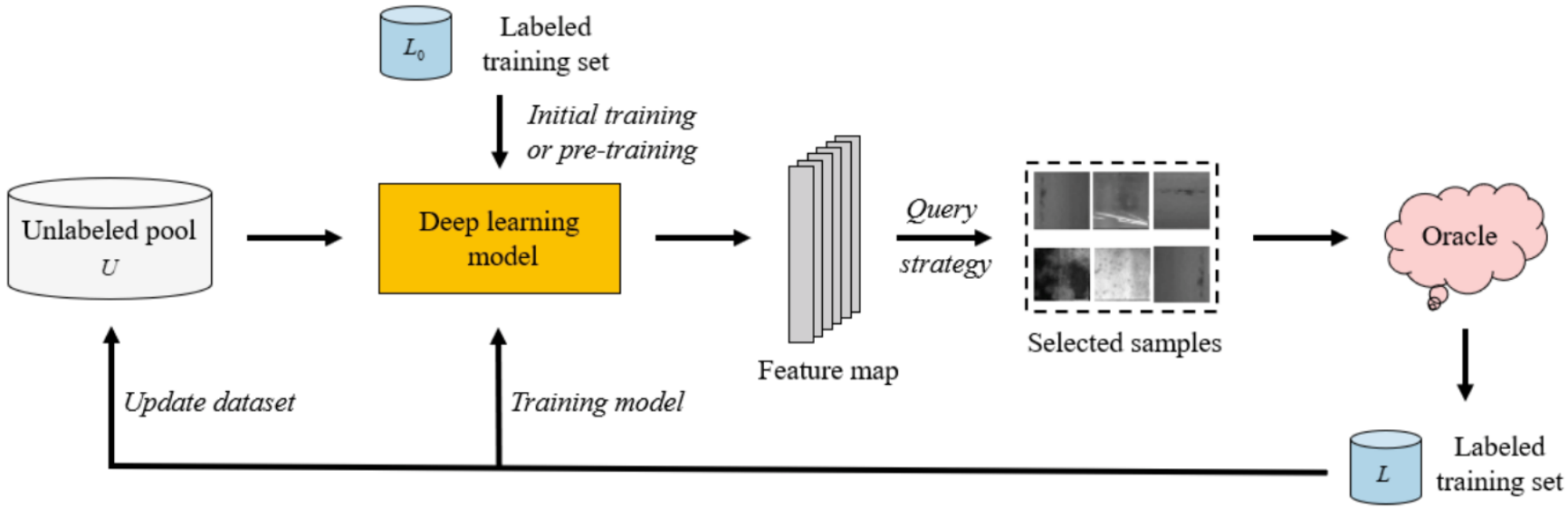
	Dataset	BL	QBC	EMCM	GSx	GSy	iGS
RMSE	Concrete-Slump	1.00	0.90	0.88	0.77	0.74	0.75
	Yacht	1.00	1.08	1.09	1.16	0.95	0.91
	autoMPG	1.00	0.85	0.86	0.73	0.77	0.73
	NO2	1.00	0.93	0.95	0.89	0.88	0.87
	PM10	1.00	0.92	0.92	0.95	0.86	0.86
	Housing	1.00	0.74	0.74	0.75	0.63	0.63
	CPS	1.00	0.80	0.76	0.70	0.64	0.64
	Concrete	1.00	0.92	0.92	0.83	0.82	0.81
	Wine-red	1.00	0.88	0.89	0.85	0.80	0.78
	Wine-white	1.00	0.89	0.90	0.87	0.82	0.84
	Average	**1.00**	**0.89**	**0.89**	**0.85**	**0.79**	**0.78**
CC	Concrete-Slump	1.00	1.15	1.10	1.29	1.35	1.35
	Yacht	1.00	1.04	1.05	1.06	1.07	1.09
	autoMPG	1.00	1.03	1.02	1.05	1.04	1.06
	NO2	1.00	1.04	1.02	1.03	1.01	1.02
	PM10	1.00	1.15	1.17	1.27	1.22	1.26
	Housing	1.00	1.12	1.12	1.16	1.17	1.21
	CPS	1.00	1.27	1.28	1.41	1.45	1.48
	Concrete	1.00	1.02	1.02	1.04	1.06	1.06
	Wine-red	1.00	0.92	0.82	0.89	0.89	0.95
	Wine-white	1.00	1.02	0.98	0.94	1.01	1.00
	Average	**1.00**	**1.08**	**1.06**	**1.11**	**1.13**	**1.15**

Outline

- Weakly Supervised Learning
- Active Learning
- Active Learning for Classification
- Active Learning for Regression
- **Deep Active Learning**
- Applications
- Conclusions

Necessity and Challenges of Combining DL & AL

- **Model uncertainty in Deep Learning.** The output confidence of the softmax layer is usually too confident and unreliable for estimating the uncertainty
- **Insufficient data for labeled samples.** The one-by-one sample query approach in classic AL is inefficient and not applicable in the DL context, where large amounts of labeled data are required
- **Processing pipeline inconsistency.** Traditional AL algorithms use fixed feature representations and focus primarily on the training of classifiers, whereas DL optimizes the feature extractor and the classifier jointly.



Bayesian Active Learning by Disagreement (BALD)

- The batch-based query strategy is the foundation of Deep AL.
- **Basic Idea**

  Select top-k samples with the highest mutual information between model parameters and predictions, i.e., maximal prediction disagreement using different model parameters

$$a_{\mathrm{BALD}}\left(\{x_1,\ldots,x_b\},\mathcal{P}\left(\omega \mid D_{train}\right)\right)=\sum_{i=1}^{b}\mathbb{I}\left(y_i;\omega \mid x_i,D_{train}\right),$$

$$\mathbb{I}\left(y;\boldsymbol{\omega} \mid x,D_{train}\right)=\mathbb{H}\left(y \mid x,D_{train}\right)-\mathbb{E}_{\mathcal{P}(\omega \mid D_{train})}\left[\mathbb{H}\left(y \mid x,\boldsymbol{\omega},D_{train}\right)\right],$$

  Drawbacks: BALD considers each sample independently and ignores the correlation between samples, which is likely to lead to local decisions

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian Active Learning for Classification and Preference Learning. *CoRR abs/1112.5745 (2011).*

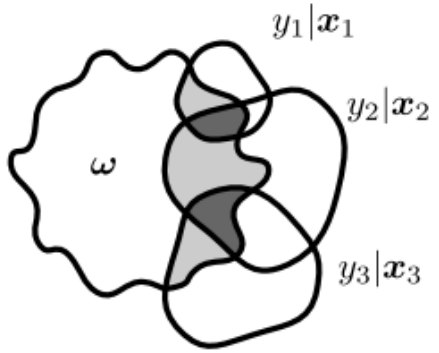
BatchBALD

- Considers the correlation between data points by estimating the joint mutual information between multiple data points and model parameters.
- **Basic Idea**

  Jointly score points by estimating the mutual information between a joint of multiple data points and the model parameters

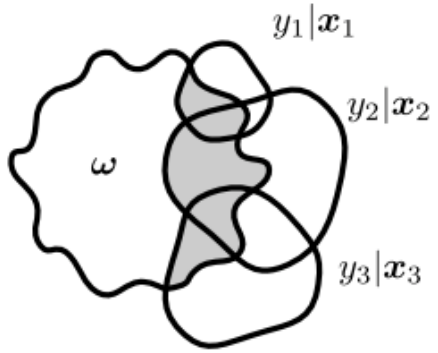
$$a_{\text{BatchBALD}}\left(\{x_1,\ldots,x_b\},\mathcal{P}\left(\omega \mid D_{train}\right)\right)=\mathbb{I}\left(y_1,\ldots,y_b;\omega \mid x_1,\ldots,x_b,D_{train}\right),$$

$$\mathbb{I}\left(y_{1:b};\boldsymbol{\omega} \mid x_{1:b},D_{train}\right)=\mathbb{H}\left(y_{1:b} \mid x_{1:b},D_{train}\right)-\mathbb{E}_{\mathcal{P}(\boldsymbol{\omega}|D_{train})}\mathbb{H}\left(y_{1:b} \mid x_{1:b},\boldsymbol{\omega},D_{train}\right),$$



$$\sum_i \mathbb{I}(y_i;\boldsymbol{\omega}\,|\,\boldsymbol{x}_i,\mathcal{D}_{\text{train}})=\sum_i \mu^*(y_i\cap\boldsymbol{\omega})$$

(a) **BALD**



$$\mathbb{I}(y_1,\ldots,y_b;\boldsymbol{\omega}\,|\,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_b,\mathcal{D}_{\text{train}})=\mu^*\left(\bigcup_i y_i\cap\boldsymbol{\omega}\right)$$

(b) **BatchBALD**

Kirsch, Andreas, Joost van Amersfoort, and Yarin Gal. "BatchBALD: efficient and diverse batch acquisition for deep Bayesian active learning." in *Proc. Int'l Conf. on Neural Information Processing Systems*. 2019.

Exploration-P

- Select the sample set S that has the largest uncertainty and the smallest redundancy
- **Basic Idea**

1. Compute the information entropy as the measure of the uncertainty

$$\mathrm{E}(x) = -\sum_{0<<i<<|Y|} h_i(x) log(h_i(x))$$

2. Calculate the similarity between an unlabeled sample and the selected sample set S

$$\mathrm{Sim}(i, S) = \max_{j \in S}(\mathrm{Sim}(i, j)) \quad \mathrm{Sim}(i, j) = f_i M f_j, \quad M \text{ denotes a similarity matrix}$$

3. Select the unlabeled sample having maximum score

$$\mathrm{I}(i) = \mathrm{E}(x_i) - \alpha \mathrm{Sim}(i, S)$$

i.e., sample with maximal uncertainty and least redundancy with the selected samples

4. Select the sample $\boldsymbol{x}_i$ that furthest from labeled and selected unlabeled sample in a greedy manner $i = \min_i \mathrm{Sim}(i, L \cup S)$

C. Yin, B. Qian, S. Cao, X. Li, J. Wei, Q. Zheng, and I. Davidson, "Deep Similarity-Based Batch Mode Active Learning with Exploration-Exploitation," in *Proc. IEEE Int'l Conf. on Data Mining*, 2017.

Batch Active Learning by Diverse Gradient Embeddings (BADGE)

- Considering both informativeness and diversity, BADGE selects a set of samples with diverse gradients
- **Basic Idea**

1. Compute the cross-entropy loss $\ell_{\mathrm{CE}}(f(x;\theta),y) = \ln\left(\sum_{j=1}^{K} e^{W_j \cdot z(x;V)}\right) - W_y \cdot z(x;V).$

2. Obtain the gradients corresponding to the predicted label

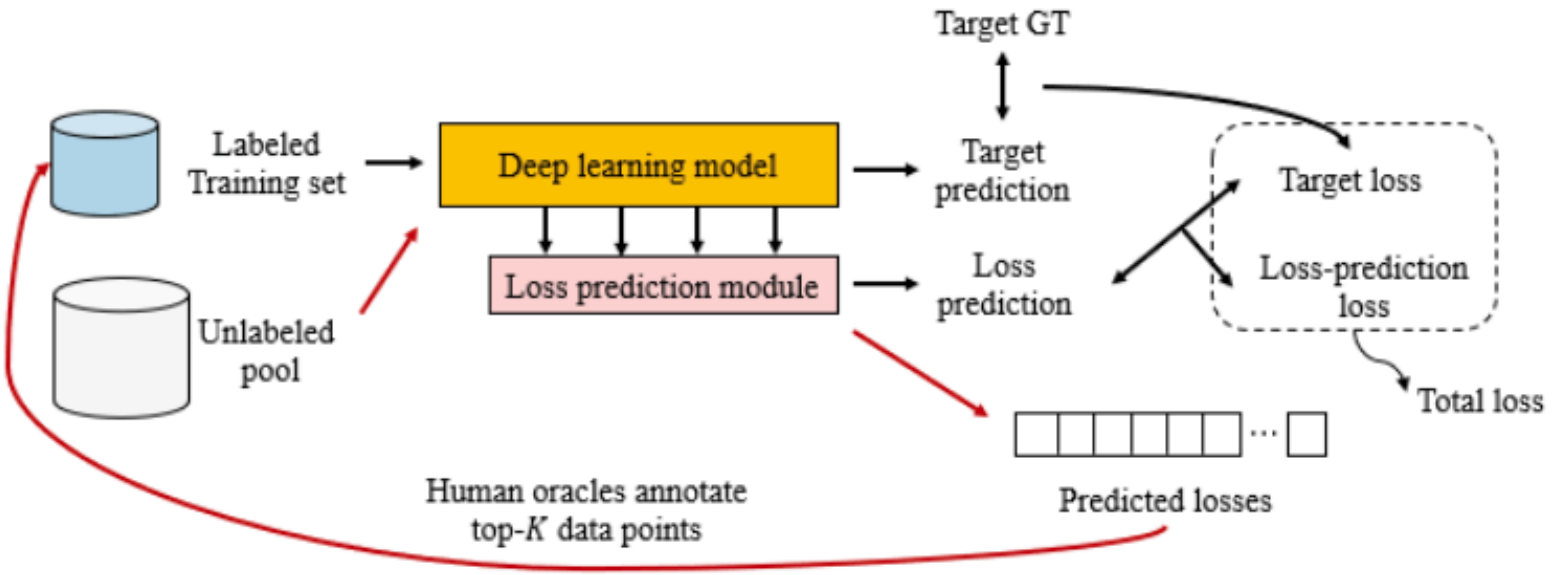
$$(g_x)_i = \frac{\partial}{\partial W_i}\ell_{\mathrm{CE}}(f(x;\theta),\hat{y}) = (p_i - I(\hat{y}=i))z(x;V).$$

3. Use the k-means++ seeding algorithm on $\{g_x : x \in U \setminus S\}$ and query for their labels.

Ash, Jordan T., Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal, "Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds." in Proc. *Int'l Conf. on Learning Representations,* Addis Ababa, Ethiopia, April. 2017.

Learning Loss for Active Learning

- Use a loss prediction module to select unlabeled samples with large losses.
- **Basic Idea**
  1. Train the task model using target loss
  2. Train the loss prediction module using ranking loss
  3. Select the top-K samples with maximal predicted losses and query for labels

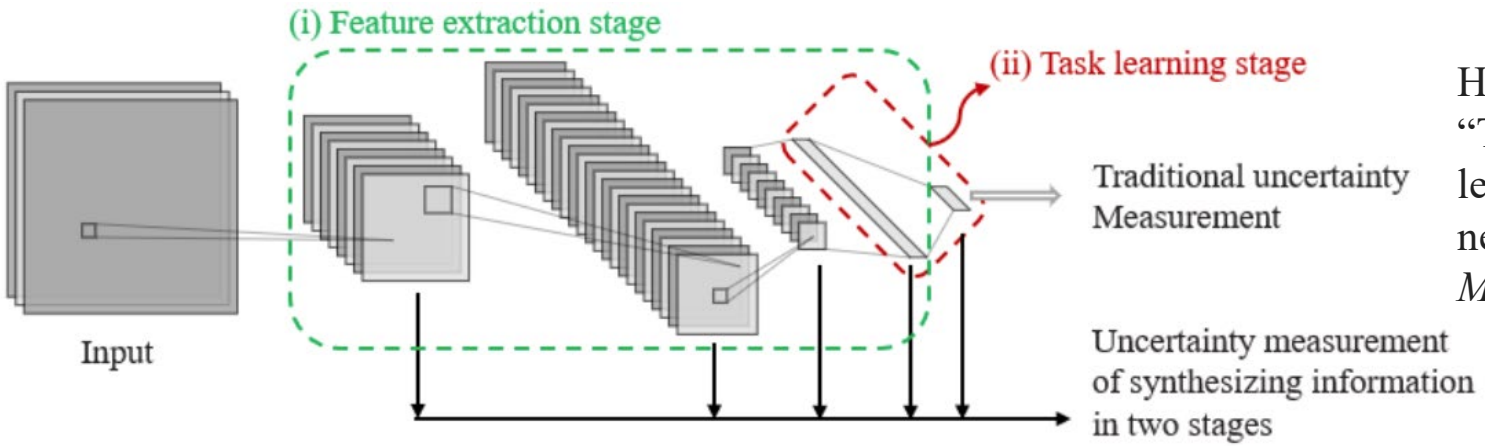


Donggeun Yoo, and In So Kweon, "Learning loss for active learning." In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition,* Long Beach, CA, Jun. pp. 93-102. 2019.

Active Learning with Multiple Views

- Estimate the uncertainty from the outputs from multiple hidden layers of the model
- **Basic Idea**

1. Train the CNN model using the training set and compute loss between the predictions and ground-truth labels on the validation set l_n
2. Add a softmax layer on top of each hidden layer to form (n-1) new classifiers. Train these models on the training set and compute the loss l_i on the validation set, $i = 1, \ldots, n-1$
3. Obtain the weight of each model by applying softmax to their validation loss $w_i = \frac{e^{-l_i}}{\sum_{k=1}^{n} e^{-l_k}}$
4. Compute the weighted average of the uncertainty in the classifiers following each hidden layer $u_j = \sum_{i=1}^{n} w_i \times f_{uncertainty}(O_i(x^j))$
5. Select the top-*k* samples with maximal uncertainty to query for labels



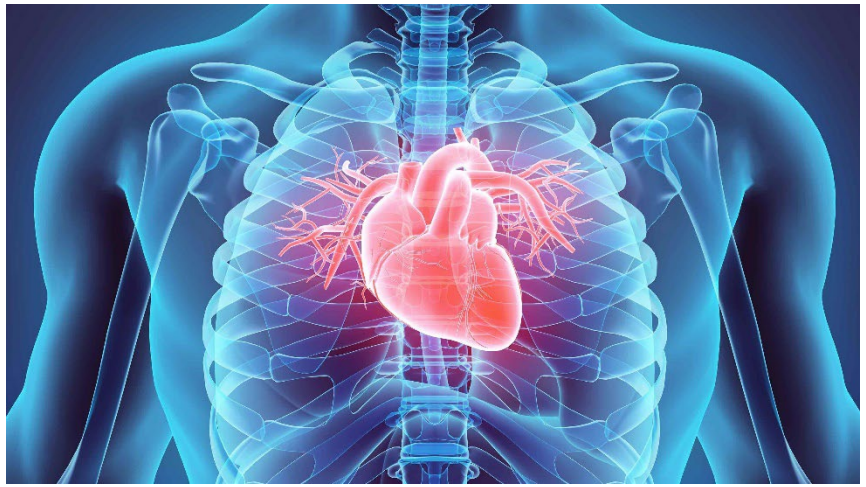
He, T., Jin, X., Ding, G., Yi, L. and Yan, C, "Towards better uncertainty sampling: Active learning with multiple views for deep convolutional neural network." in *Proc. IEEE Int'l Conf. on Multimedia and Expo,* Shanghai, China, Jul. 2019.

Outline

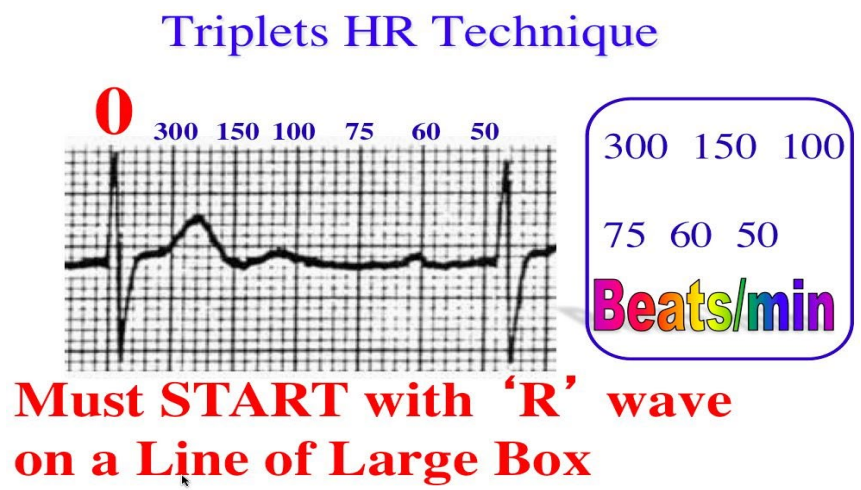
- Weakly Supervised Learning
- Active Learning
- Active Learning for Classification
- Active Learning for Regression
- Deep Active Learning
- **Applications**
- Conclusions

Application 1: Cardiovascular Diseases

- Cardiovascular diseases (CVDs) are the **leading** cause of human death: Take **17.9 million** lives every year, **31%** of all global deaths (WHO).



- Real-time accurate heart rate estimation from wearable ECG system is critical to cardiovascular disease detection and treatment.



D. Wu, C. Guo, F. Liu and C. Liu, "Active Stacking for Heart Rate Estimation," *Int'l Joint Conf. on Neural Networks* (IJCNN), Glasgow, UK, July 2020.

Heart Rate Estimation

- ECG from wearable systems generally has **poor quality** due to bad electrode contact, wrong electrode positioning, body movements, and various noise.

- Traditional heart rate estimation algorithms, which mainly considered clinic quality ECG signals, **cannot** be used.

Ensemble Regression (ER)

- ER can improve the estimation performance, by integrating multiple base estimators.
- In heart rate estimation, different QRS detectors can be viewed as base estimators.
- **Unsupervised** ER (no labeled ECG trials are available): Average, median.
- **Supervised** ER (some labeled ECG trials are available): Bagging, Boosting, stacking.
- Supervised ER usually outperforms unsupervised ER.
- How to minimize the number of labeled ECG trials in supervised ER?

Stacking

Assume among the N ECG trials, K have been labeled, i.e., their reference heart rates $\{y_k\}_{k=1}^{K}$ are known.

Stacking trains a regression model $\hat{y}_n = f(\mathbf{x}_n)$ from these K trials.

Ridge regression (RR) ($\lambda = 0.01$):

$$\hat{y}_n = \mathbf{w}^T \mathbf{x}_n + b, \quad n = 1, ..., N$$

where b and $\mathbf{w} = [w_1, ..., w_M]^T$ are obtained from minimizing the following objective function:

$$g(b, \mathbf{w}) = \sum_{k=1}^{K} (y_k - \hat{y}_k)^2 + \lambda \mathbf{w}^T \mathbf{w}$$

Linear SVR ($C = 1$):

$$g(b, \mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{k=1}^{K} \epsilon_k$$

$$s.t. \quad |y_k - \hat{y}_k| \leq \epsilon_k, \quad \epsilon_k \geq 0$$

Active Stacking Using GSx (AS-GSx)

AS-GSx integrates stacking and GSx:

1. Use GSx to select K trials to query for their reference heart rates
2. Check if any base estimator has the same heart rate estimates as the reference for all K selected trials:
   - ✓ **Yes**: For each of the remaining N-K trials, the median of these base estimators is taken as its final estimate.
   - ✓ **No**: Train a linear SVR model from the K labeled trials as the final stacking model.

Active Stacking Using RD (AS-RD)

AS-RD integrates stacking and RD:

1. Use RD to select K trials to query for their reference heart rates
2. Check if any base estimator has the same heart rate estimates as the reference for all K selected trials:
   - ✓ **Yes**: For each of the remaining N-K trials, the median of these base estimators is taken as its final estimate.
   - ✓ **No**: Train a linear SVR model from the K labeled trials as the final stacking model.

AS-RD-EMCM

AS-RD-EMCM integrates stacking and RD-EMCM:

1. Use RD-EMCM to select $K_0=2$ trials to query for their reference heart rates.
2. Train a linear SVR stacking model from them.
3. Use the SVR model in RD-EMCM to select the next trial to label, and update the linear SVR stacking model.
4. Iterate until K trials have been selected and labeled.
5. Check if any base estimator has the same heart rate estimates as the reference for all K selected trials:
   - ✓ **Yes**: For each of the remaining N-K trials, the median of these base estimators is taken as its final estimate.
   - ✓ **No**: Train a linear SVR model from the K labeled trials as the final stacking model.

AS-iGS

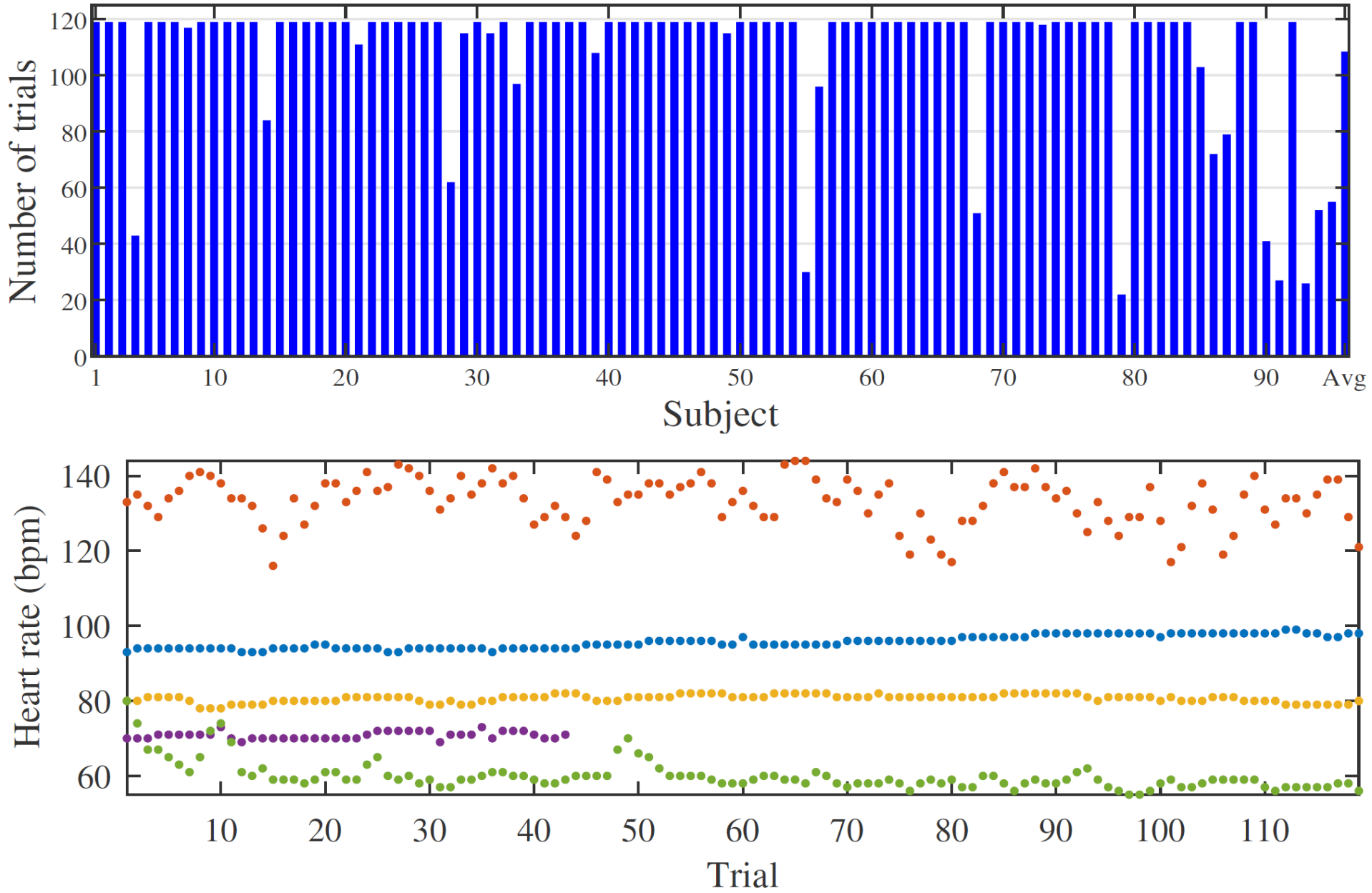
AS-iGS integrates stacking and iGS:

1. Use iGS to select $K_0=2$ trials to query for their reference heart rates.
2. Train a linear SVR stacking model from them.
3. Use the SVR model in iGS to select the next trial to label, and update the linear SVR stacking model.
4. Iterate until K trials have been selected and labeled.
5. Check if any base estimator has the same heart rate estimates as the reference for all K selected trials:
   - ✓ **Yes**: For each of the remaining N-K trials, the median of these base estimators is taken as its final estimate.
   - ✓ **No**: Train a linear SVR model from the K labeled trials as the final stacking model.

Datasets

- 100 ECG recordings in the augmented training set of 2014 PhysioNet/CinC Challenge.
- Patients with a wide range of problems, and healthy volunteers.
- Each recording was $\leq$ 10 minutes, 360 Hz, 16-bit resolution.
- Four recordings (2041, 2728, 41024, 41778) shorter than 2 minutes, and one consisting of pure Gaussian noise (42878), were excluded.
- The remaining 95 ECG recordings had manually annotated QRS complex locations.

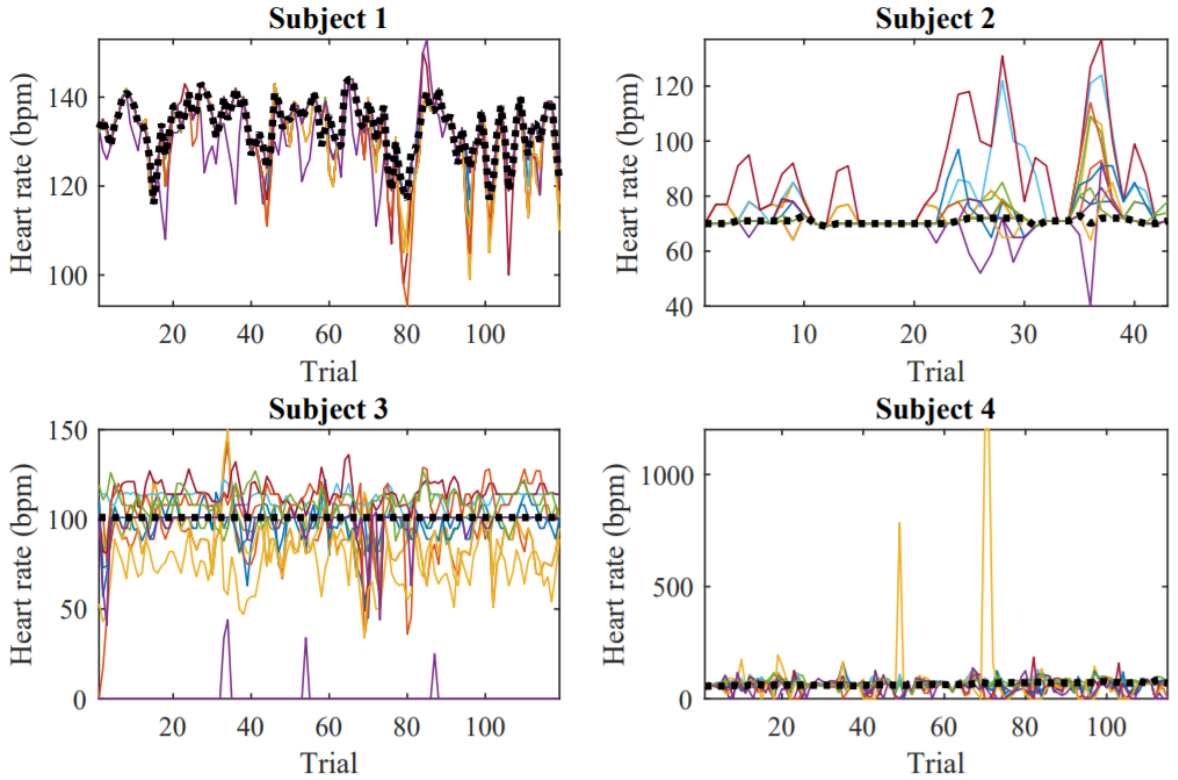
Datasets

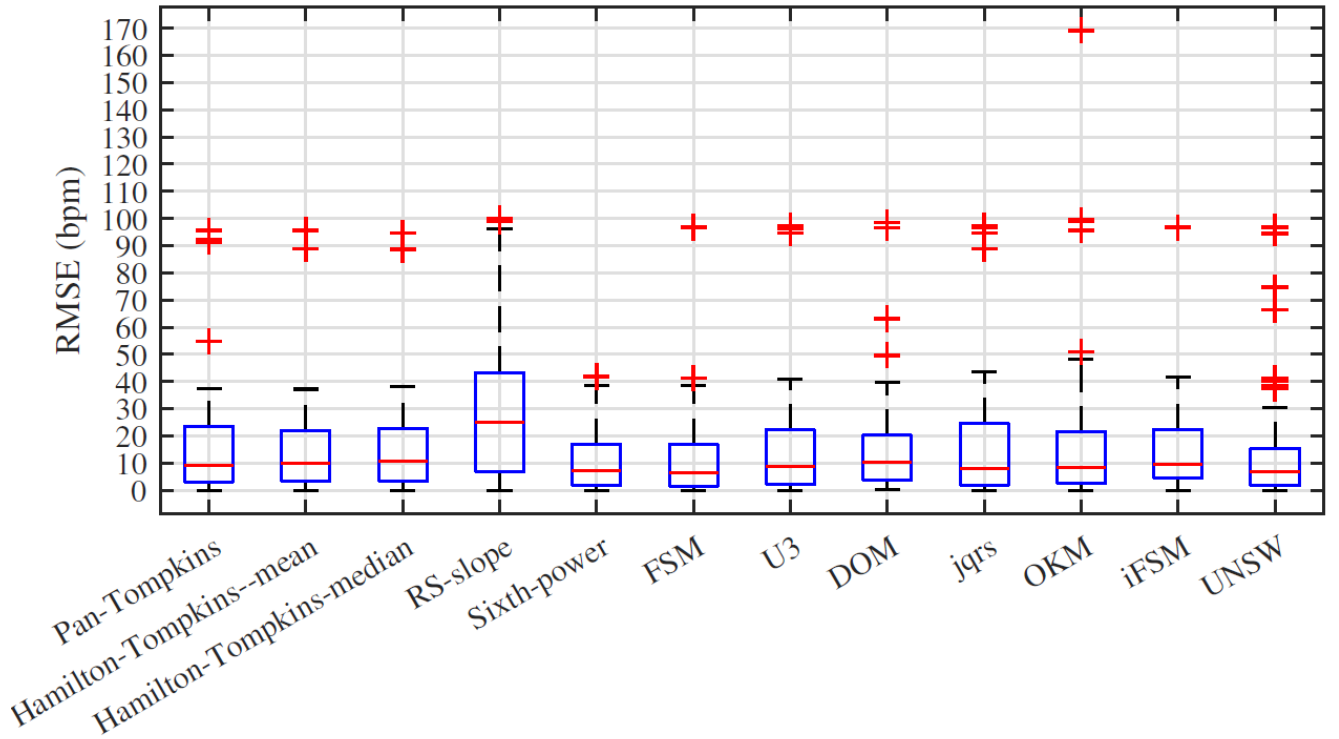

Number of trials
120
100
80
60
40
20
0
1
10
20
30
40
50
60
70
80
90
Avg
Subject
Heart rate (bpm)
140
120
100
80
60
10
20
30
40
50
60
70
80
90
100
110
Trial

12 Base Estimators (QRS Detectors)

1. Pan-Tompkins
2. Hamilton-Tompkins-mean
3. Hamilton-Tompkins-median
4. RS-slope
5. Sixth-power
6. Finite state machine (FSM)
7. Improved FSM (iFSM)
8. U3
9. Difference operation algorithm (DOM)
10. jqrs
11. Optimized knowledge-based method (OKM)
12. UNSW

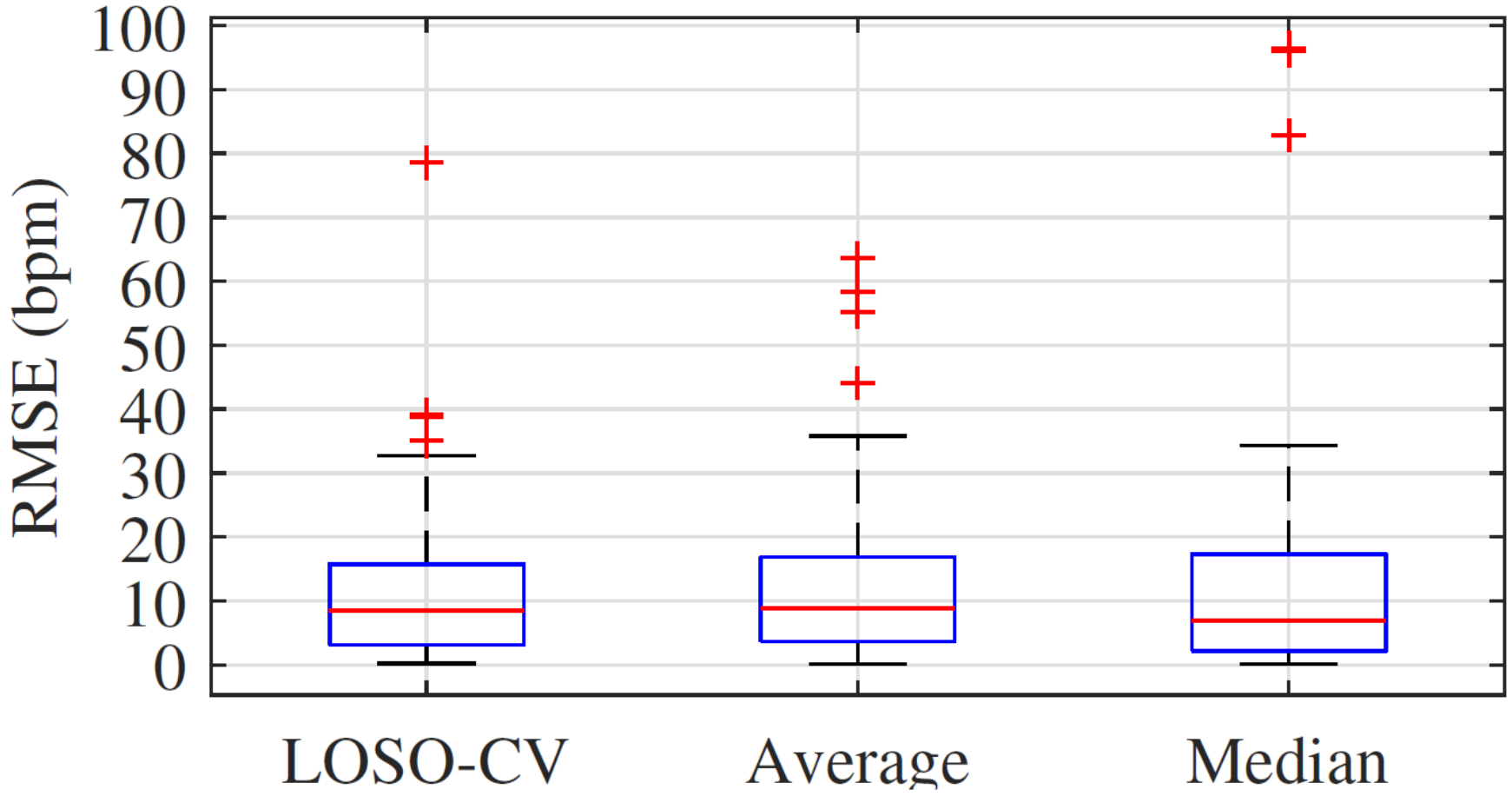
RMSEs of the Base Estimators





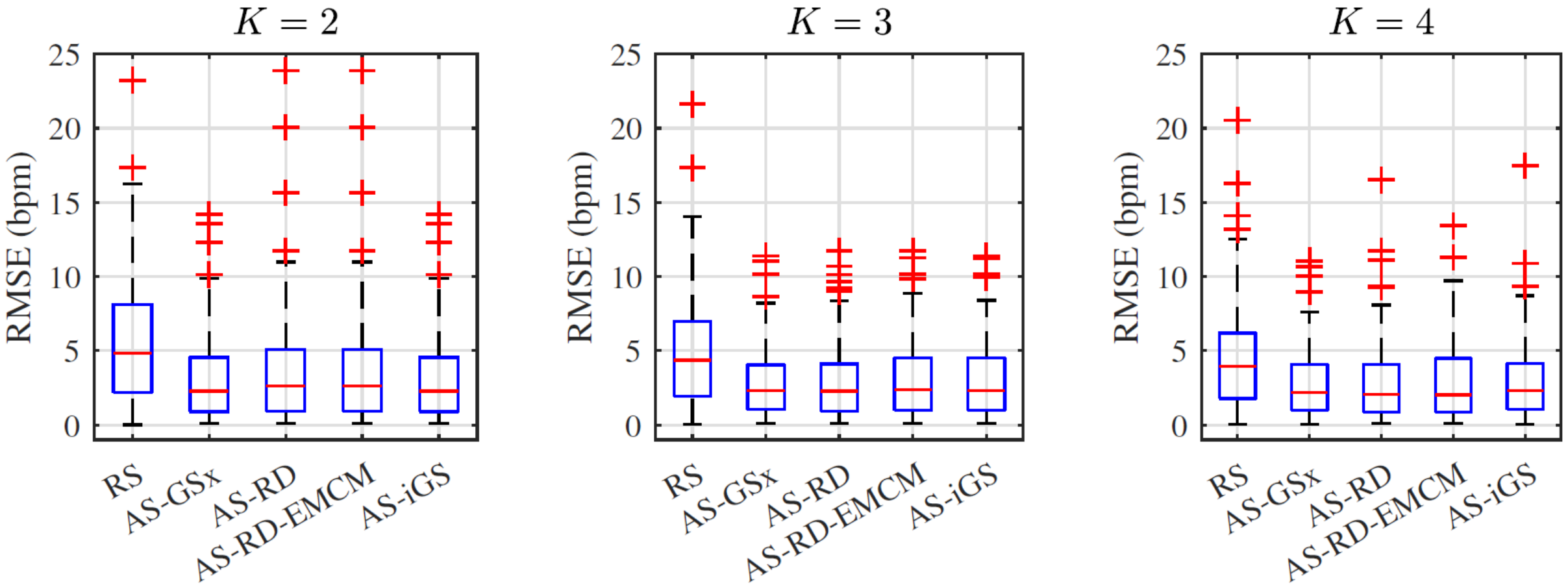
- Sixth-power achieved the smallest RMSE (10.55 bpm), and RS-slope the largest (29.57 bpm).
- These RMSEs represented **11.99-33.61%** relative error.
- Requirement: ≤**10%,** or **5 bpm**, whichever is greater.

RMSEs of Unsupervised ER



Given that the mean heart rate across the 95 subjects was 87.99 bpm, these RMSEs represented **12.92-13.75%** relative error, which should not be acceptable in practice.

RMSEs of Active Stacking



RMSEs much smaller than those of the 12 base estimators, and also much smaller than those of the three unsupervised ensemble regression approaches.

Active Stacking vs. Random Stacking

- RMSEs of the proposed AS approaches converge at ***K*=3 or 4**, i.e., only 3 or 4 labeled trials are needed for them to achieve a low RMSE, very favorable in practice.
- Compared with RS, AS can reduce the RMSE by **35-40%**, suggesting the effectiveness of using ALR in heart rate estimation.

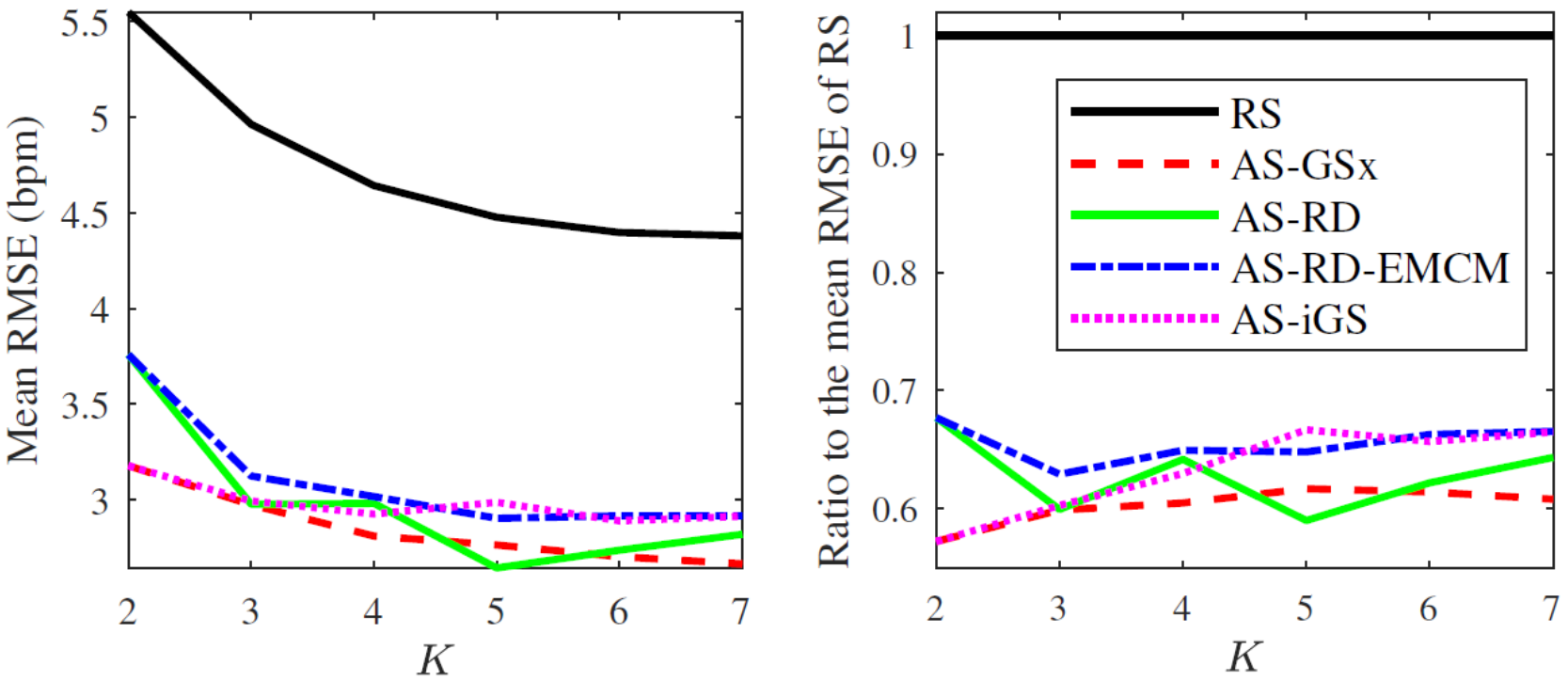


Fig. 4. Mean RMSEs (left) of the five supervised stacking approaches across the 95 subjects, and the ratio (right) to the mean RMSE of RS.

RMSEs and CCs

Table 1: The mean and standard deviation (std) of the RMSEs of different approaches.

Category	Approach		RMSE mean (bpm)	RMSE std (bpm)
Base Estimator	Pan-Tompkins		15.49	18.06
	Hamilton-Tompkins-mean		14.69	15.78
	Hamilton-Tompkins-median		14.87	15.73
	RS-slope		29.57	26.92
	Sixth-power		**10.55**	**10.95**
	FSM		12.14	16.16
	iFSM		15.26	16.54
	U3		15.68	20.47
	DOM		15.67	19.03
	jqrs		16.33	20.83
	OKM		17.09	25.30
	UNSW		14.22	21.88
Unsupervised Ensemble Regression	LOSO-CV		**11.37**	**11.65**
	Average		11.97	12.14
	Median		12.10	16.86
Supervised	$K = 2$	RS	5.55	4.45
		AS-GSx	**3.18**	**3.07**
		AS-RD	3.76	4.02
		AS-RD-EMCM	3.76	4.02
		AS-iGS	**3.18**	**3.07**
	$K = 3$	RS	4.96	4.15
		AS-GSx	**2.97**	2.68
		AS-RD	2.98	**2.65**
		AS-RD-EMCM	3.12	2.67
		AS-iGS	2.99	2.66
	$K = 4$	RS	4.64	3.97
		AS-GSx	**2.81**	**2.45**
		AS-RD	2.98	2.95
		AS-RD-EMCM	3.02	2.75
		AS-iGS	2.92	2.78

Requirement:
$\leq$**10%,** or **5 bpm**, whichever is greater.

A Subtle Detail: Why Take the Median

AS-GSx integrates stacking & GSx:

1. Use GSx to select K trials to query for their reference HRs
2. Check if any base estimator has the same HR estimates as the reference for all K selected trials:
   - ✓ **Yes**: For each of the remaining N-K trials, the median of these base estimators is taken as its final estimate.
   - ✓ **No**: Train a linear SVR model from the K labeled trials as the final stacking model.

- **Median**, which takes the median of the selected base estimators.
- **Subset**, which performs a linear SVR on the selected base estimators.
- **All**, which performs a linear SVR on all 12 base estimators.

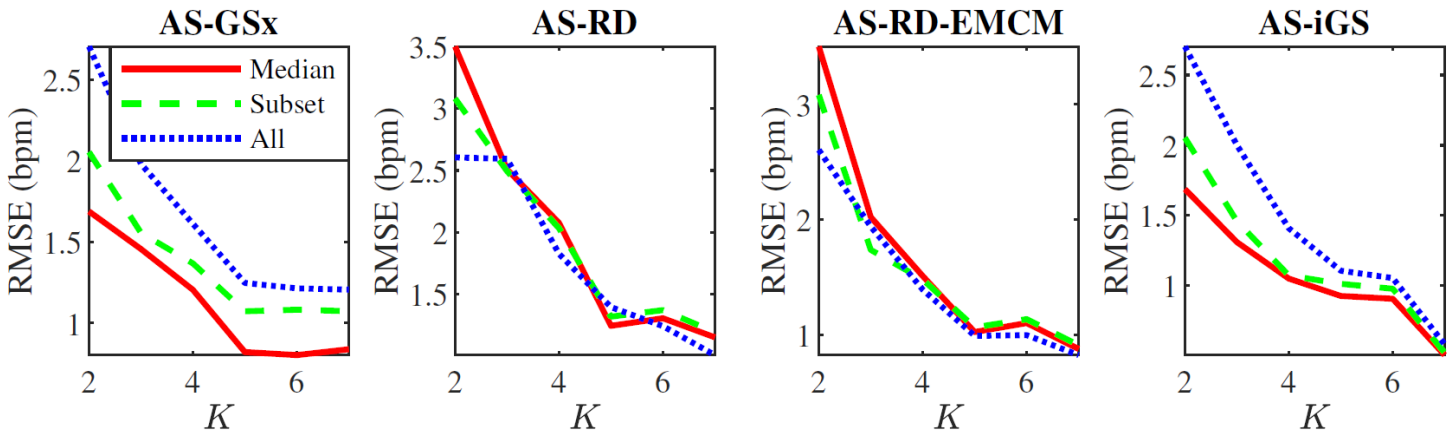
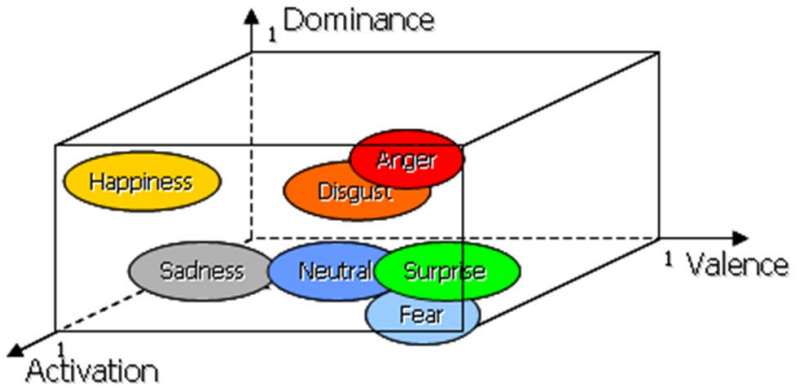


Fig. 5. Average RMSEs of three variants of the algorithm, when there exist some base estimators whose outputs are identical to the reference heart rates on all selected trials.

Application 2: Affective Computing

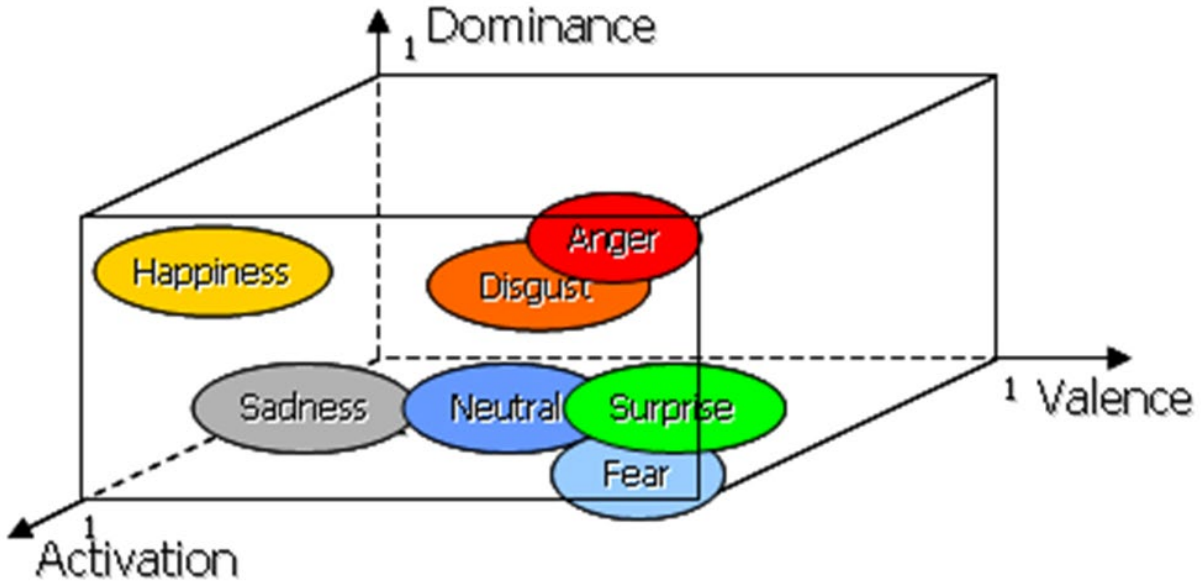
- Emotions can be represented in the 2D space of arousal and valence, or in the 3D space of arousal, valence, and dominance.
- Emotions are very subjective, subtle, and uncertain.



- Multiple assessors are needed to obtain the groundtruth emotion values for each affective sample (video, audio, image, physiological signal, etc).
  - ✓ 14-16 assessors were used to evaluate each video clip in the DEAP dataset
  - ✓ 6-17 assessors for each utterance in the VAM spontaneous speech corpus
  - ✓ 110+ assessors for each sound in the IADS-2 dataset
- Very time-consuming and labor-intensive.
- **Challenge**: How to optimally select the affective samples to label so that an accurate regression model can be built with the minimum cost?

Multi-Task AL

- Traditional **(single-task) AL**: Optimally query the unlabeled samples to predict only one output.
- **Multi-task AL**: Optimally query the unlabeled samples, so that the 3 dimensions of emotion can be predicted simultaneously.



D. Wu and J. Huang, "Affect Estimation in 3D Space Using Multi-Task Active Learning for Regression," *IEEE Trans. on Affective Computing*, 2022.

Single-Task Gsy vs. Multi-Task GSy

Single-task GSy:

Single-task GSy first uses GSx to select the first K samples to label, and build a regression model $f(\mathbf{x})$. For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^{N}$, GSy computes first its distance to each of the k outputs:

$$d_{nm}^{y} = ||f(\mathbf{x}_n) - y_m||, \quad \begin{matrix} m = 1, ..., k \\ n = k+1, ..., N \end{matrix}$$

and d_n^y, the shortest distance from $f(\mathbf{x}_n)$ to $\{y_m\}_{m=1}^{k}$:

$$d_n^y = \min_m d_{nm}^y, \quad n = k+1, ..., N$$

and then selects the sample with the maximum d_n^y to label.

Multi-task GSy:

MT-GSy uses GSx to select the first k samples, and trains P regression models $f_p(\mathbf{x})$ $(p = 1, ..., P)$. For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^{N}$, MT-GSy computes first its distance to each of the k outputs, for each of the P tasks:

$$d_{nm,p}^{\mathbf{y}} = ||f_p(\mathbf{x}_n) - y_m||$$

where $m = 1, ..., k$, $n = k+1, ..., N$, and $p = 1, ..., P$. MT-GSy then computes:

$$d_n^{\mathbf{y}} = \min_m \prod_{p=1}^{P} d_{nm,p}^{\mathbf{y}}, \quad n = k+1, ..., N$$

and selects the sample with the maximum $d_n^{\mathbf{y}}$ to label.

Single-Task iGS vs. Multi-Task iGS

Single-task iGS:

Assume the first k samples have already been labeled with labels $\{y_n\}_{n=1}^{k}$. For each of the remaining $N-k$ unlabeled sample $\{\mathbf{x}_n\}_{n=k+1}^{N}$, single-task iGS computes first its distance to each of the k labeled samples in the input space:

$$d_{nm}^{\mathbf{x}} = ||\mathbf{x}_n - \mathbf{x}_m||, \quad m = 1, ..., k \qquad n = k+1, ..., N$$

and d_{nm}^{y} in GSy, and then $d_n^{\mathbf{x}y}$:

$$d_n^{\mathbf{x}y} = \min_m d_{nm}^{\mathbf{x}} d_{nm}^{y}, \quad n = k+1, ..., N$$

Next, single-task iGS selects the sample with the maximum $d_n^{\mathbf{x}y}$ to label.

Multi-task iGS:

MT-iGS first uses iGS to select and label K samples. It then builds P regression models $\{f_p(\mathbf{x})\}_{p=1}^{P}$ for the P tasks. For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^{N}$, MT-iGS computes $d_{nm}^{\mathbf{x}}$ and $d_{nm,p}^{\mathbf{y}}$, and then $d_n^{\mathbf{xy}}$:

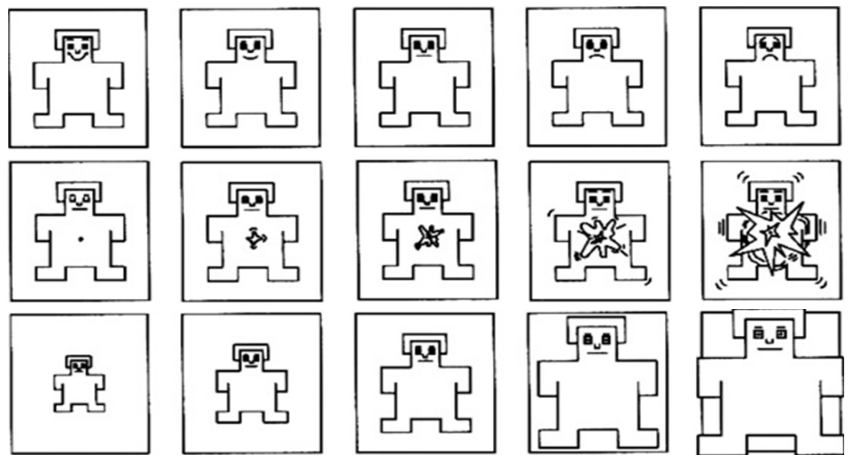
$$d_n^{\mathbf{xy}} = \min_m d_{nm}^{\mathbf{x}} \prod_{p=1}^{P} d_{nm,p}^{\mathbf{y}}, \quad n = k+1, ..., N$$

and selects the sample with the maximum $d_n^{\mathbf{xy}}$ to label.

Data Description

- **VAM Corpus**: Spontaneous speech with authentic emotions in a German TV talk-show *Vera am Mittag* (*Vera at Noon* in English).
- 947 emotional utterances from 47 speakers (11m/36f).
- Main characteristics:
  - ✓ Authentic & real life conversations
  - ✓ Emotion evaluated in 3D space

http://www.sat1.de/comedy_show/vera/



Feature Extraction

46 acoustic features:

- **Pitch features (9)**: f0 mean, std, median, min, max, range, 25% & 75% quantiles, and the inter-quantile distance.
- **Duration features (5)**: mean/std of the duration of voiced/unvoiced segments, ratio between the duration of unvoiced and voiced segments.
- **Energy features (6)**: energy mean, std, max, 25% & 75% quantiles, and the inter-quantile distance.
- **MFCC features (26)**: mean/std of 13 Mel-frequency cepstral coefficients.

Results: RMSE & CC vs. K

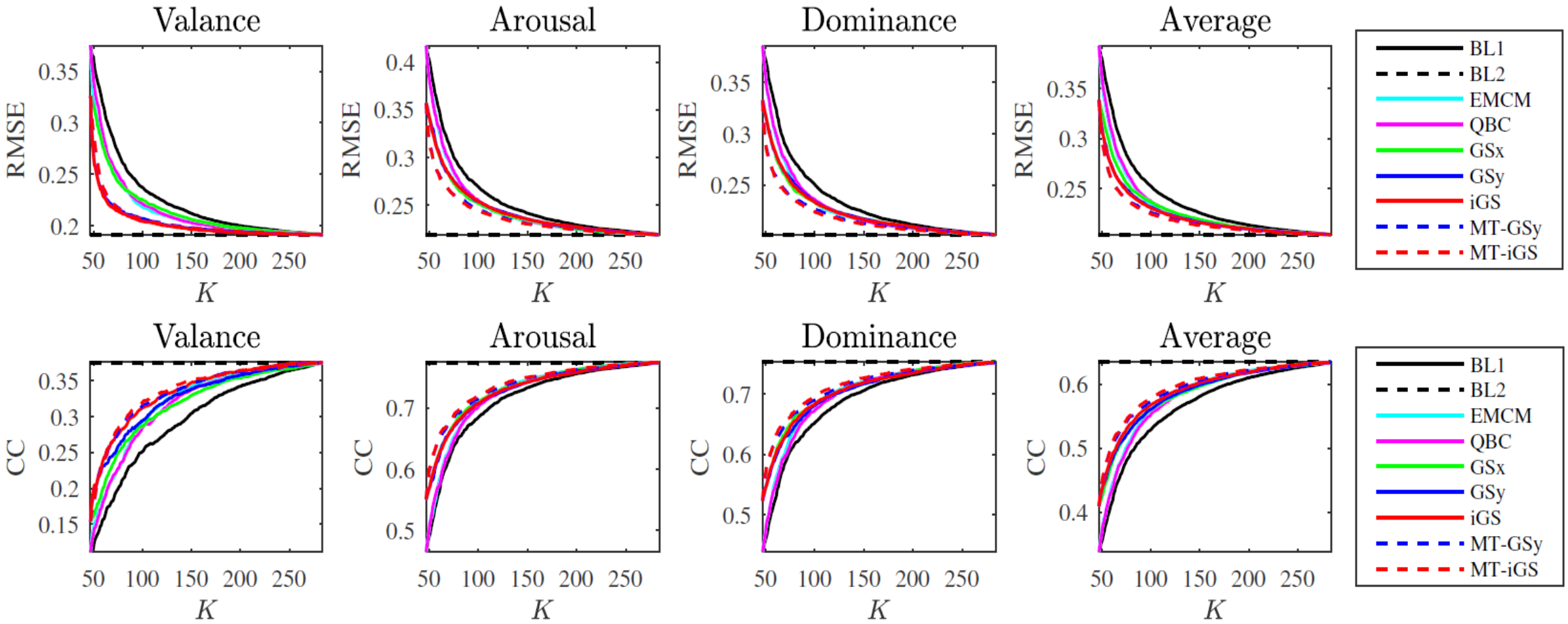


Fig. 1. Performances of the sample selection algorithms, averaged over 100 runs, when the single-task ALR approaches focused on *Valence* estimation. The last column shows the average RMSE and CC across the three tasks. RR was used as the regression model.

Results: RMSE & CC Percentage Improvement

Performances and percentages of improvement (in the parentheses) when different ALR approaches are compared with BL1. The best two in each row are marked in bold.

Emotion Primitive	Performance Measure	K	BL1	Performance and percentage improvement over BL1						
				EMCM	QBC	GSx	GSy	iGS	MT-GSy	MT-iGS
Valence	RMSE	50	0.380	0.356 (6%)	0.361 (5%)	0.326 (14%)	0.311 (18%)	0.310 (18%)	**0.300 (21%)**	**0.299 (21%)**
		100	0.252	0.235 (7%)	0.237 (6%)	0.237 (6%)	0.232 (8%)	0.230 (9%)	**0.226 (10%)**	**0.225 (11%)**
		150	0.226	0.217 (4%)	0.217 (4%)	0.219 (3%)	0.216 (4%)	0.216 (4%)	**0.214 (5%)**	**0.213 (6%)**
		200	0.213	0.210 (2%)	0.210 (2%)	0.210 (1%)	0.210 (2%)	0.210 (2%)	**0.209 (2%)**	**0.208 (2%)**
		250	0.207	0.206 (1%)	0.206 (1%)	0.206 (1%)	0.206 (1%)	0.206 (1%)	**0.206 (1%)**	**0.205 (1%)**
	CC	50	0.354	0.371 (5%)	0.367 (4%)	0.424 (20%)	0.434 (23%)	0.437 (23%)	**0.446 (26%)**	**0.448 (26%)**
		100	0.529	0.560 (6%)	0.553 (5%)	0.560 (6%)	0.561 (6%)	0.568 (7%)	**0.574 (8%)**	**0.579 (9%)**
		150	0.581	0.604 (4%)	0.600 (3%)	0.597 (3%)	0.599 (3%)	0.603 (4%)	**0.606 (4%)**	**0.609 (5%)**
		200	0.610	0.621 (2%)	0.619 (1%)	0.618 (1%)	0.618 (1%)	0.619 (1%)	**0.622 (2%)**	**0.623 (2%)**
		250	0.626	0.630 (1%)	0.630 (1%)	0.629 (1%)	0.630 (1%)	**0.631 (1%)**	0.630 (1%)	**0.631 (1%)**
Arousal	RMSE	50	0.374	0.350 (6%)	0.357 (4%)	0.330 (12%)	0.311 (17%)	0.308 (18%)	**0.300 (20%)**	**0.298 (20%)**
		100	0.253	0.235 (7%)	0.236 (7%)	0.234 (7%)	0.235 (7%)	0.232 (8%)	**0.226 (11%)**	**0.225 (11%)**
		150	0.224	0.217 (3%)	0.216 (4%)	0.216 (4%)	0.219 (2%)	0.217 (3%)	**0.213 (5%)**	**0.213 (5%)**
		200	0.213	0.209 (2%)	0.209 (2%)	0.209 (2%)	0.210 (1%)	0.209 (2%)	**0.208 (3%)**	**0.208 (3%)**
		250	0.207	0.205 (1%)	0.205 (1%)	0.205 (1%)	0.206 (1%)	0.205 (1%)	**0.205 (1%)**	**0.205 (1%)**
	CC	50	0.368	0.393 (7%)	0.379 (3%)	0.419 (14%)	0.436 (18%)	0.442 (20%)	**0.447 (21%)**	**0.449 (22%)**
		100	0.529	0.559 (6%)	0.554 (5%)	0.567 (7%)	0.557 (5%)	0.564 (7%)	**0.573 (8%)**	**0.576 (9%)**
		150	0.584	0.603 (3%)	0.599 (3%)	0.604 (3%)	0.593 (1%)	0.600 (3%)	**0.606 (4%)**	**0.608 (4%)**
		200	0.609	0.620 (2%)	0.620 (2%)	0.621 (2%)	0.615 (1%)	0.619 (2%)	**0.622 (2%)**	**0.622 (2%)**
		250	0.626	0.630 (1%)	0.630 (1%)	0.630 (1%)	0.628 (0%)	0.630 (1%)	**0.631 (1%)**	**0.631 (1%)**
Dominance	RMSE	50	0.370	0.354 (4%)	0.359 (3%)	0.321 (13%)	0.304 (18%)	0.303 (18%)	**0.296 (20%)**	**0.296 (20%)**
		100	0.251	0.236 (6%)	0.235 (6%)	0.235 (7%)	0.233 (7%)	0.231 (8%)	**0.224 (11%)**	**0.224 (11%)**
		150	0.224	0.217 (3%)	0.217 (3%)	0.217 (3%)	0.217 (3%)	0.216 (4%)	**0.213 (5%)**	**0.213 (5%)**
		200	0.213	0.209 (2%)	0.209 (2%)	0.209 (2%)	0.210 (2%)	0.210 (1%)	**0.208 (2%)**	**0.208 (2%)**
		250	0.207	0.205 (1%)	0.205 (1%)	0.205 (1%)	0.205 (1%)	0.206 (1%)	**0.205 (1%)**	**0.205 (1%)**
	CC	50	0.377	0.388 (3%)	0.384 (2%)	0.433 (15%)	0.445 (18%)	0.446 (18%)	**0.453 (20%)**	**0.454 (21%)**
		100	0.536	0.560 (5%)	0.559 (4%)	0.566 (6%)	0.558 (4%)	0.566 (6%)	**0.579 (8%)**	**0.579 (8%)**
		150	0.586	0.602 (3%)	0.600 (2%)	0.601 (3%)	0.597 (2%)	0.601 (3%)	**0.607 (4%)**	**0.608 (4%)**
		200	0.611	0.621 (2%)	0.619 (1%)	0.620 (2%)	0.616 (1%)	0.617 (1%)	**0.621 (2%)**	**0.623 (2%)**
		250	0.626	0.630 (1%)	0.630 (1%)	0.630 (1%)	0.629 (1%)	0.629 (1%)	**0.630 (1%)**	**0.631 (1%)**

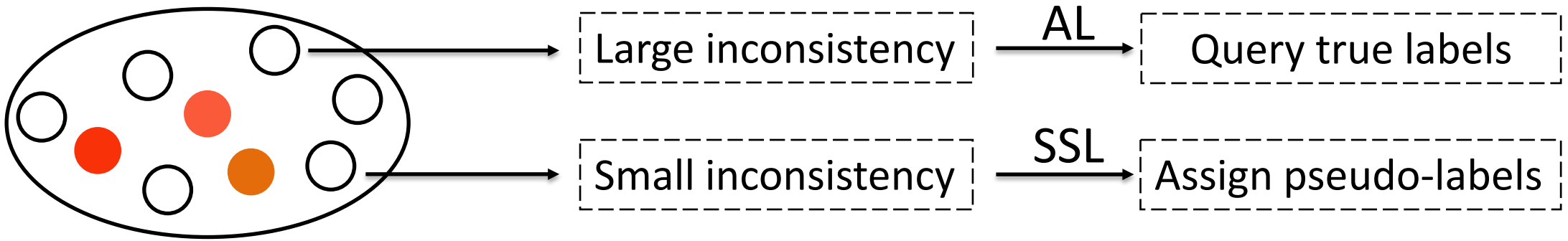
Results: Percentages of Saved Queries

Number of samples and percentages of saved queries (in the parentheses) when different ALR approaches are compared with BL1. The best two in each row are marked in bold.

Emotion Primitive	Performance Measure	α%	No. BL1 Samples	Number of samples and percentage saving over BL1						
				EMCM	QBC	GSx	GSy	iGS	MT-GSy	MT-iGS
Valence	RMSE	1%	261	**242 (8%)**	248 (5%)	247 (6%)	246 (6%)	242 (8%)	243 (7%)	**233 (12%)**
		2%	241	218 (11%)	217 (11%)	221 (9%)	218 (11%)	216 (12%)	**207 (16%)**	**202 (19%)**
		3%	222	197 (13%)	197 (13%)	201 (10%)	194 (14%)	197 (13%)	**183 (21%)**	**179 (24%)**
		5%	197	168 (17%)	168 (17%)	175 (13%)	164 (20%)	162 (22%)	**148 (33%)**	**144 (37%)**
		10%	154	123 (25%)	126 (22%)	129 (19%)	118 (31%)	116 (33%)	**106 (45%)**	**101 (52%)**
	CC	1%	258	**236 (9%)**	242 (7%)	242 (7%)	244 (6%)	238 (8%)	242 (7%)	**230 (12%)**
		2%	235	202 (16%)	211 (11%)	211 (11%)	215 (9%)	208 (13%)	**201 (17%)**	**194 (21%)**
		3%	215	181 (19%)	187 (15%)	190 (13%)	189 (14%)	185 (16%)	**175 (23%)**	**172 (25%)**
		5%	184	149 (23%)	154 (19%)	162 (14%)	157 (17%)	150 (23%)	**144 (28%)**	**135 (36%)**
		10%	138	109 (27%)	115 (20%)	112 (23%)	110 (25%)	104 (33%)	**98 (41%)**	**93 (48%)**
Arousal	RMSE	1%	261	239 (9%)	**237 (10%)**	245 (7%)	252 (4%)	247 (6%)	**231 (13%)**	238 (10%)
		2%	242	213 (14%)	210 (15%)	216 (12%)	227 (7%)	220 (10%)	**199 (22%)**	**196 (23%)**
		3%	225	193 (17%)	192 (17%)	196 (15%)	208 (8%)	197 (14%)	**176 (28%)**	**174 (29%)**
		5%	196	165 (19%)	165 (19%)	167 (17%)	175 (12%)	161 (22%)	**143 (37%)**	**139 (41%)**
		10%	152	125 (22%)	125 (22%)	126 (21%)	126 (21%)	116 (31%)	**98 (55%)**	**97 (57%)**
	CC	1%	261	235 (11%)	235 (11%)	242 (8%)	247 (6%)	241 (8%)	**231 (13%)**	**228 (14%)**
		2%	241	202 (19%)	203 (19%)	208 (16%)	219 (10%)	210 (15%)	**198 (22%)**	**188 (28%)**
		3%	223	181 (23%)	184 (21%)	186 (20%)	200 (12%)	188 (19%)	**174 (28%)**	**165 (35%)**
		5%	185	147 (26%)	155 (19%)	155 (19%)	168 (10%)	152 (22%)	**135 (37%)**	**132 (40%)**
		10%	136	110 (24%)	113 (20%)	105 (30%)	115 (18%)	105 (30%)	**92 (48%)**	**91 (49%)**
Dominance	RMSE	1%	261	237 (10%)	238 (10%)	237 (10%)	249 (5%)	242 (8%)	**232 (13%)**	**235 (11%)**
		2%	242	210 (15%)	212 (14%)	213 (14%)	221 (10%)	216 (12%)	**204 (19%)**	**201 (20%)**
		3%	225	192 (17%)	194 (16%)	191 (18%)	204 (10%)	193 (17%)	**182 (24%)**	**180 (25%)**
		5%	197	165 (19%)	166 (19%)	162 (22%)	174 (13%)	164 (20%)	**148 (33%)**	**146 (35%)**
		10%	151	123 (23%)	127 (19%)	124 (22%)	126 (20%)	121 (25%)	**106 (42%)**	**103 (47%)**
	CC	1%	258	236 (9%)	239 (8%)	231 (12%)	248 (4%)	238 (8%)	**226 (14%)**	**229 (13%)**
		2%	234	200 (17%)	210 (11%)	203 (15%)	219 (7%)	211 (11%)	**196 (19%)**	**196 (19%)**
		3%	217	181 (20%)	190 (14%)	181 (20%)	197 (10%)	186 (17%)	**175 (24%)**	**173 (25%)**
		5%	184	148 (24%)	159 (16%)	147 (25%)	167 (10%)	155 (19%)	**143 (29%)**	**139 (32%)**
		10%	133	110 (21%)	116 (15%)	104 (28%)	113 (18%)	107 (24%)	**96 (39%)**	**95 (40%)**

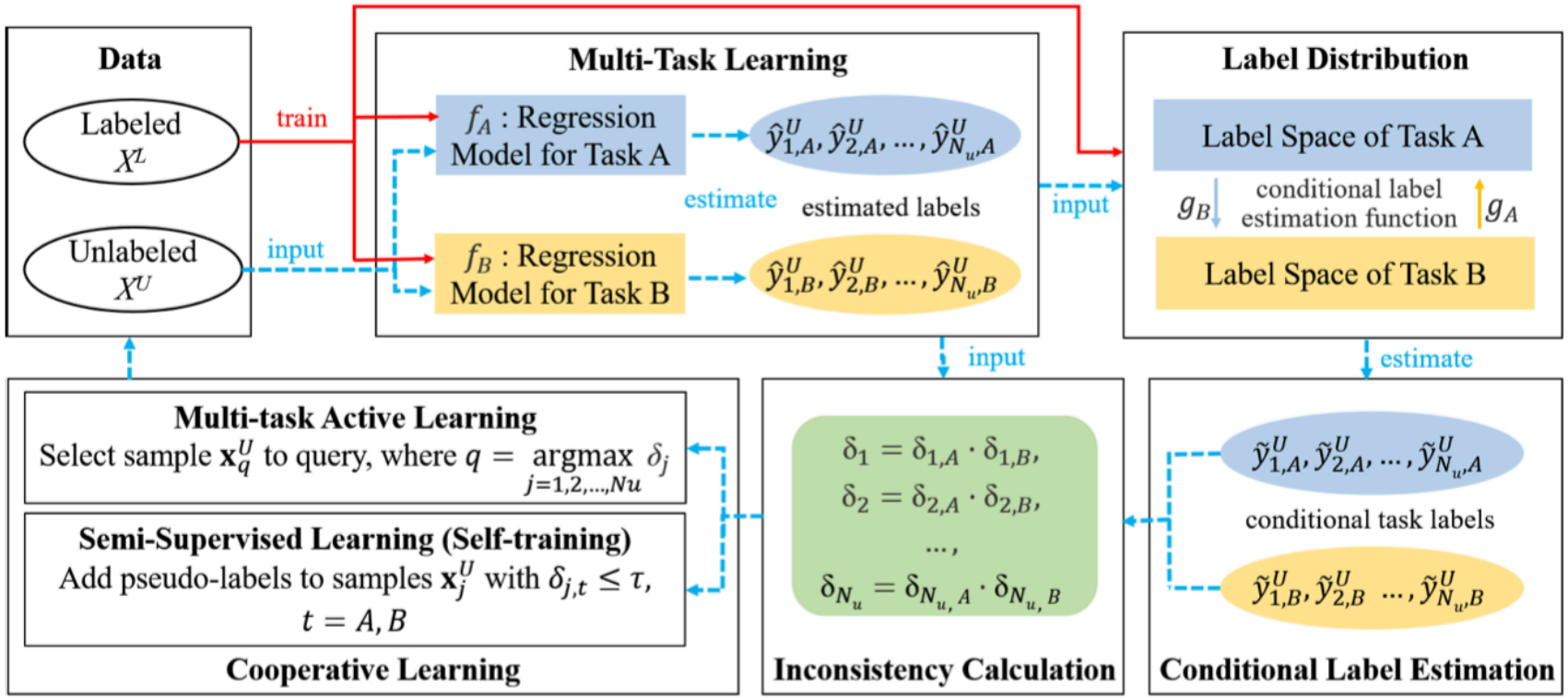
Multi-Task AL and SSL

- **Multi-task AL**: Query the unlabeled samples with maximal inconsistency of the labels estimated from features and labels of the other tasks.
- **Multi-task SSL**: Assign pseudo-labels for part of the unlabeled data with maximal consistency.



Y. Xu, Y. Cui, X. Jiang, Y. Yin, J. Ding, L. Li, and **D. Wu**, "Inconsistency-based multi-task cooperative learning for emotion recognition," *IEEE Trans. on Affective Computing*, vol. 13, no. 4, pp. 20172027, 2022.

Inconsistency-based Multi-task Cooperative Learning



Input: Labeled training data $X^L = \{\boldsymbol{x}^L_i, \boldsymbol{y}^L_{i,\mathcal{T}}\}_{i=1}^{N_L}$;
Unlabeled training data $X^U = \{\boldsymbol{x}^U_j\}_{j=1}^{N_U}$;
α, weight of the estimated label in (8);
K, number of samples to be queried;
Sample selection rules in SSL.
Output: $|\mathcal{T}|$ emotion recognition models $\{f_t\}_{t\in\mathcal{T}}$.
for $t \in \mathcal{T}$ **do**
  Use $\{\boldsymbol{x}^L_i, y^L_{i,t}\}_{i=1}^{N_L}$ to train f_t;
end
for $k = 1 : K$ **do**
  Estimate $\{\hat{\boldsymbol{y}}^U_{j,\mathcal{T}}\}_{j=1}^{N_U}$ of X^U using (1);
  Initialize X^P_t to $\emptyset$, $t \in \mathcal{T}^{dis}$;
  for $t \in \mathcal{T}^{dis}$ **do**
    Construct the conditional label estimation function g_t using $\{\boldsymbol{y}^L_{i,\mathcal{T}^{rel}_t}, y^L_{i,t}\}_{i=1}^{N_L}$;
    Obtain the conditional task labels $\{\tilde{y}^U_{j,t}\}_{j=1}^{N_U}$ of X^U using (2) for MDEE or (4) for SECE;
  end
  for $j = 1 : N_U$ **do**
    for $t \in \mathcal{T}^{dis}$ **do**
      Add sample $\boldsymbol{x}^U_j$ and its corresponding pseudo-label $\bar{y}_{j,t}$ computed by (8) to X^P_t, if it satisfies sample selection rules in SSL;
    end
  end
  Compute the inconsistency $\{\delta_j\}_{j=1}^{N_U}$ using (5) for MDEE or (6) for SECE;
  Select the most inconsistent sample $\boldsymbol{x}^U_q$ using (7);
  Query for $\boldsymbol{y}^U_{q,\mathcal{T}}$, groundtruth labels of $\boldsymbol{x}^U_q$ in all the tasks;
  $X^U \leftarrow X^U \backslash \boldsymbol{x}^U_q$, $N_U \leftarrow N_U - 1$;
  $X^L \leftarrow X^L \cup (\boldsymbol{x}^U_q, \boldsymbol{y}^U_{q,\mathcal{T}})$, $N_L \leftarrow N_L + 1$;
  for $t \in \mathcal{T}$ **do**
    if $t \in \mathcal{T}^{dis}$ **then**
      Use $\{\boldsymbol{x}^L_i, y^L_{i,t}\}_{i=1}^{N_L} \cup X^P_t$ to update f_t;
    else
      Use $\{\boldsymbol{x}^L_i, y^L_{i,t}\}_{i=1}^{N_L}$ to update f_t;
    end
  end
end

Inconsistency-based Multi-task Cooperative Learning

1. Estimate the labels from features $\hat{\boldsymbol{y}}_{j,\mathcal{T}}^{U}=\left[\hat{y}_{j,1}^{U};\ldots;\hat{y}_{j,|\mathcal{T}|}^{U}\right]=\left[f_1(\boldsymbol{x}_j^{U});\ldots;f_{|\mathcal{T}|}(\boldsymbol{x}_j^{U})\right]$
2. Estimated the labels from labels in the other tasks

$$\tilde{y}_{j,t}^{U}=g_t\left(\hat{\boldsymbol{y}}_{j,\mathcal{T}_t^{rel}}^{U}\right)=kNN\left(\hat{\boldsymbol{y}}_{j,\mathcal{T}_t^{rel}}^{U}\right),\quad t\in\mathcal{T}^{dis}$$

Multi-dimensional emotion estimation

$$\tilde{\boldsymbol{y}}_{j,\mathcal{T}^{dis}}^{U}=g(\hat{\boldsymbol{y}}_{j,C}^{U})=\sum_{e\in E}\hat{y}_{j,e}^{U}\cdot\boldsymbol{h}_e.$$

$$\boldsymbol{h}_e=\frac{1}{\sum_{i=1}^{N^L}y_{i,e}}\sum_{i=1}^{N^L}y_{i,e}^{L}\cdot\left[y_{i,v}^{L};y_{i,a}^{L};y_{i,d}^{L}\right]$$

Simultaneous emotion classification and estimation

3. Compute the inconsistency of the above two estimated labels

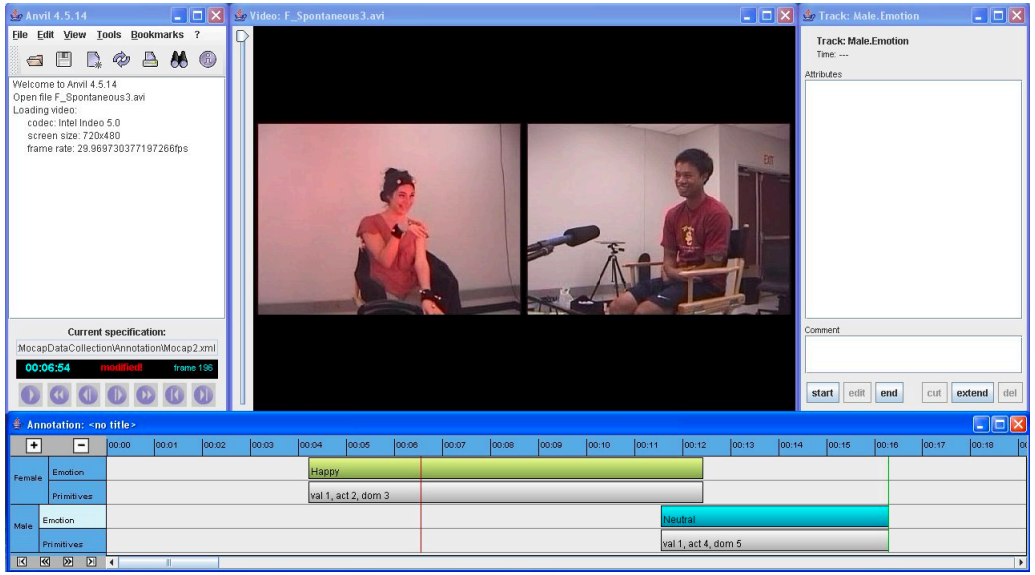
$$\delta_j=\sum_{t\in\mathcal{T}^{dis}}\delta_{j,t}=\sum_{t\in\mathcal{T}^{dis}}\left|\hat{y}_{j,t}^{U}-\tilde{y}_{j,t}^{U}\right|\quad\text{or}\quad\delta_j=\left\|\hat{\boldsymbol{y}}_{j,\mathcal{T}^{dis}}^{U}-\tilde{\boldsymbol{y}}_{j,\mathcal{T}^{dis}}^{U}\right\|_2$$

4. Query true labels of the sample $\boldsymbol{x}_q^U$ with maximal inconsistency
5. Assign pseudo-labels of dimensional emotions for samples having low inconsistency by $\bar{y}_{j,t}=\alpha\times\hat{y}_{j,t}^{U}+(1-\alpha)\times\tilde{y}_{j,t}^{U}$
6. Use the samples with manual labels and pseudo-labels to update the models

Data Description

Dataset	Size	d	Valence (mean±std)	Arousal (mean±std)	Dominance (mean±std)
VAM	947	46	-0.2282 ±0.1991	0.0280 ±0.3425	0.0924 ±0.3025
IAPS	1,178	30	5.0314 ±1.7708	4.8159 ±1.1509	5.1580 ±1.0811
IEMOCAP	2,815	35	2.8272 ±1.0576	3.1829 ±0.7606	3.2481 ±0.8077

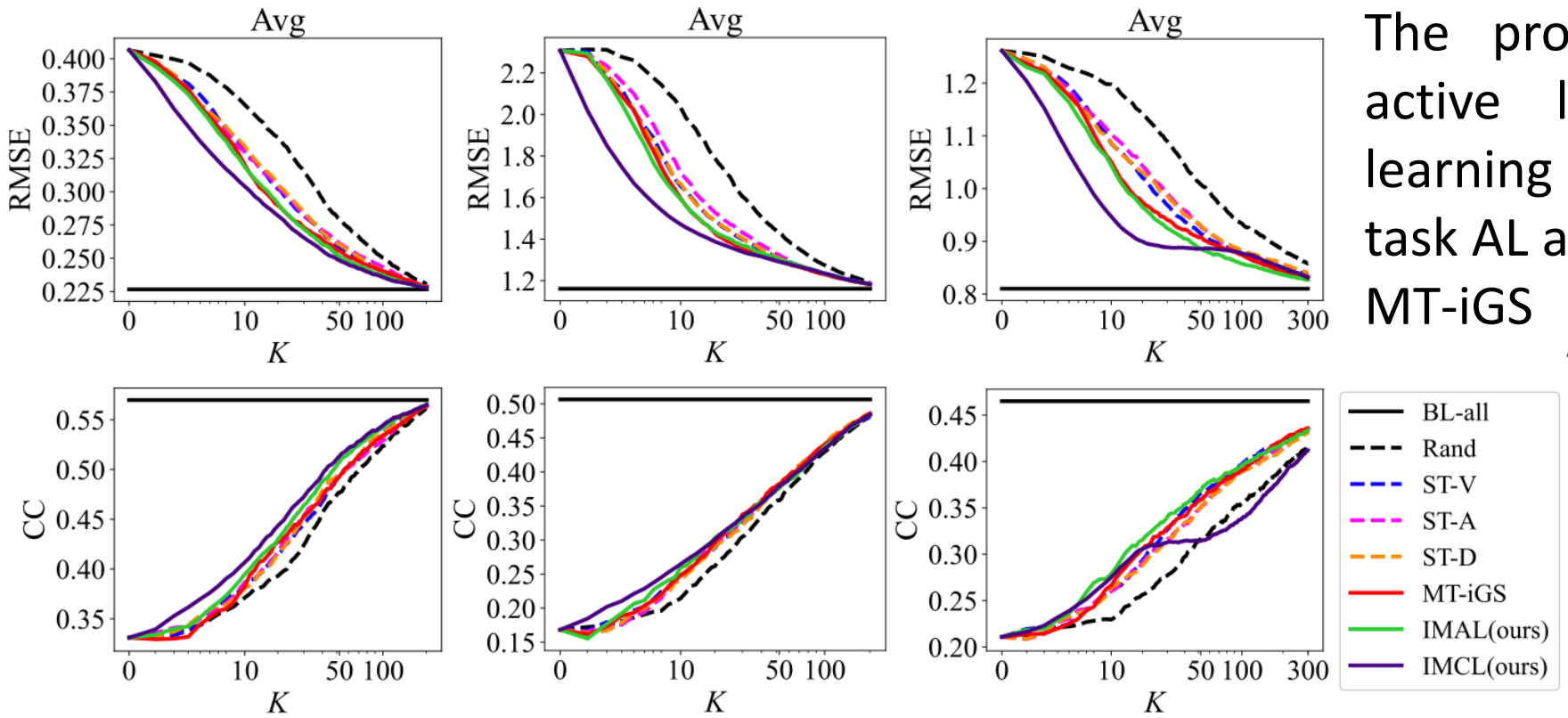
Dataset	Feature Extraction
VAM	Nine pitch features, five duration features, six energy features, and 26 MFCC features
IAPS	Principal component analysis was applied to the features extracted by the ResNet-50 pretrained on ImageNet
IEMOCAP	Two amplitude features, two energy features, one pause feature, one harmonics feature, two pitch features, one zero-crossing rate feature, and 26 MFCC features



interface of annotation

pictures in IAPS that elicit different emotions

Results: RMSE & CC in Multi-Dimensional Emotion Estimation



VAM IAPS IEMOCAP

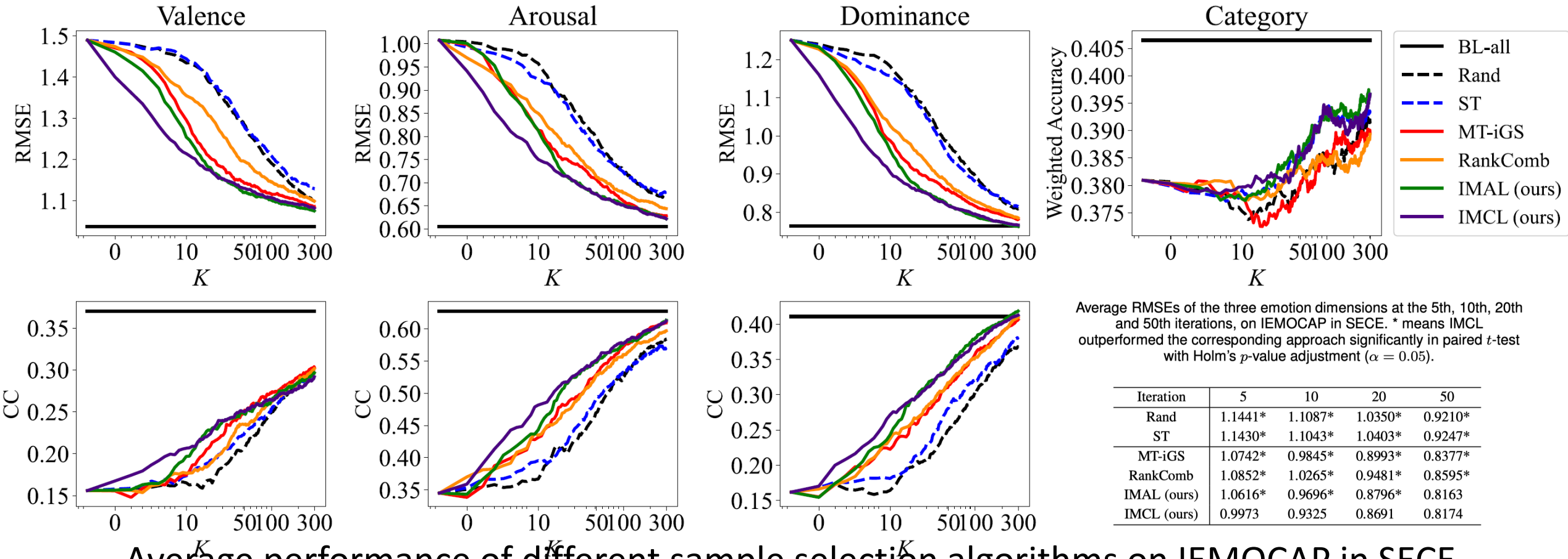
Average performance of different sample selection algorithms in MDEE on three datasets

The proposed inconsistency base active learning and cooperative learning outperformed the single task AL approaches and the previous MT-iGS

Average RMSEs of the three emotion dimensions at the 5th, 10th, 20th and 50th iterations, on IEMOCAP in MDEE. * means IMCL outperformed the corresponding approach significantly in paired t-test with Holm's p-value adjustment ($\alpha = 0.05$).

Iteration	5	10	20	50
Rand	1.2232*	1.1975*	1.1302*	1.0083*
ST-V	1.1731*	1.0854*	1.0087*	0.9140*
ST-A	1.1610*	1.1031*	1.0307*	0.9287*
ST-D	1.1536*	1.0865*	1.0210*	0.9303*
MT-iGS	1.1579*	1.0512*	0.9664*	0.9072*
IMAL (ours)	1.1351*	1.0441*	0.9550*	0.8869
IMCL (ours)	1.0362	0.9470	0.8951	0.8860

Results: RMSE & CC in Simultaneous Emotion Classification and Estimation



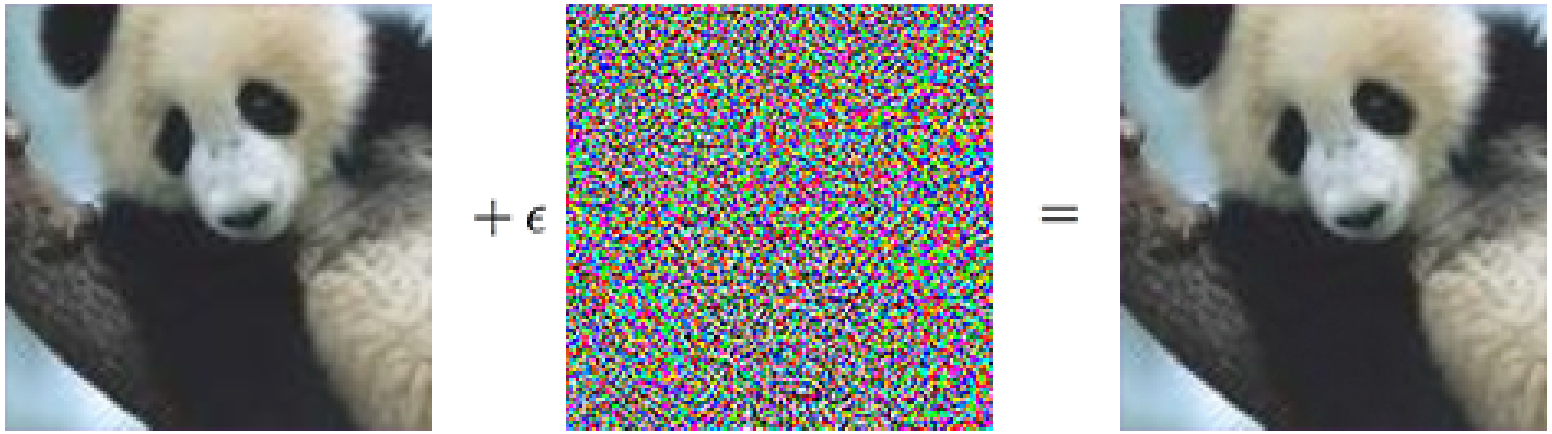
Average RMSEs of the three emotion dimensions at the 5th, 10th, 20th and 50th iterations, on IEMOCAP in SECE. * means IMCL outperformed the corresponding approach significantly in paired t-test with Holm's p-value adjustment ($\alpha = 0.05$).

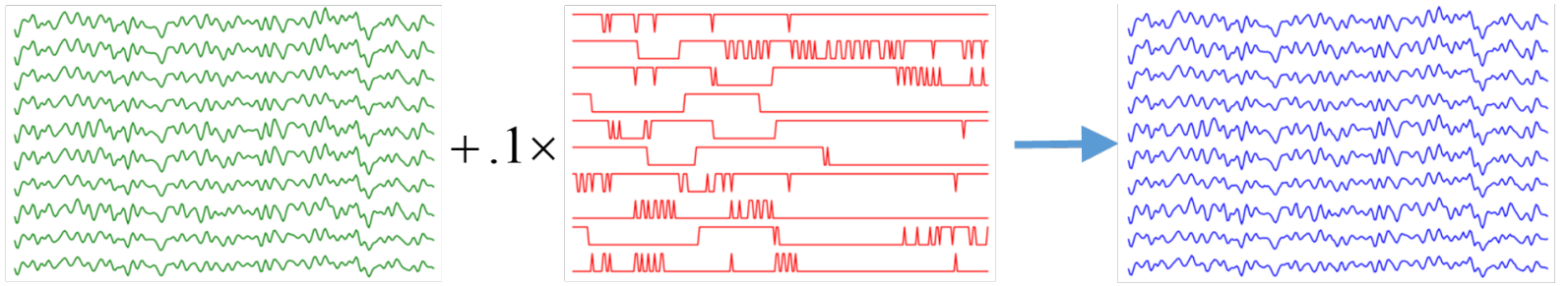
Iteration	5	10	20	50
Rand	1.1441*	1.1087*	1.0350*	0.9210*
ST	1.1430*	1.1043*	1.0403*	0.9247*
MT-iGS	1.0742*	0.9845*	0.8993*	0.8377*
RankComb	1.0852*	1.0265*	0.9481*	0.8595*
IMAL (ours)	1.0616*	0.9696*	0.8796*	0.8163
IMCL (ours)	0.9973	0.9325	0.8691	0.8174

Average performance of different sample selection algorithms on IEMOCAP in SECE (valence-arousal-dominance estimation and also emotion classification). Generally our proposed IMCL achieved the best performance, and IMAL the second best

Application 3: Adversarial Attacks

Deliberately designed small perturbations, which may be hard to notice even by human, are added to normal examples to fool a machine learning model and cause dramatic performance degradation.





Adversarial Attack Types

According to how much the attacker knows the target model:

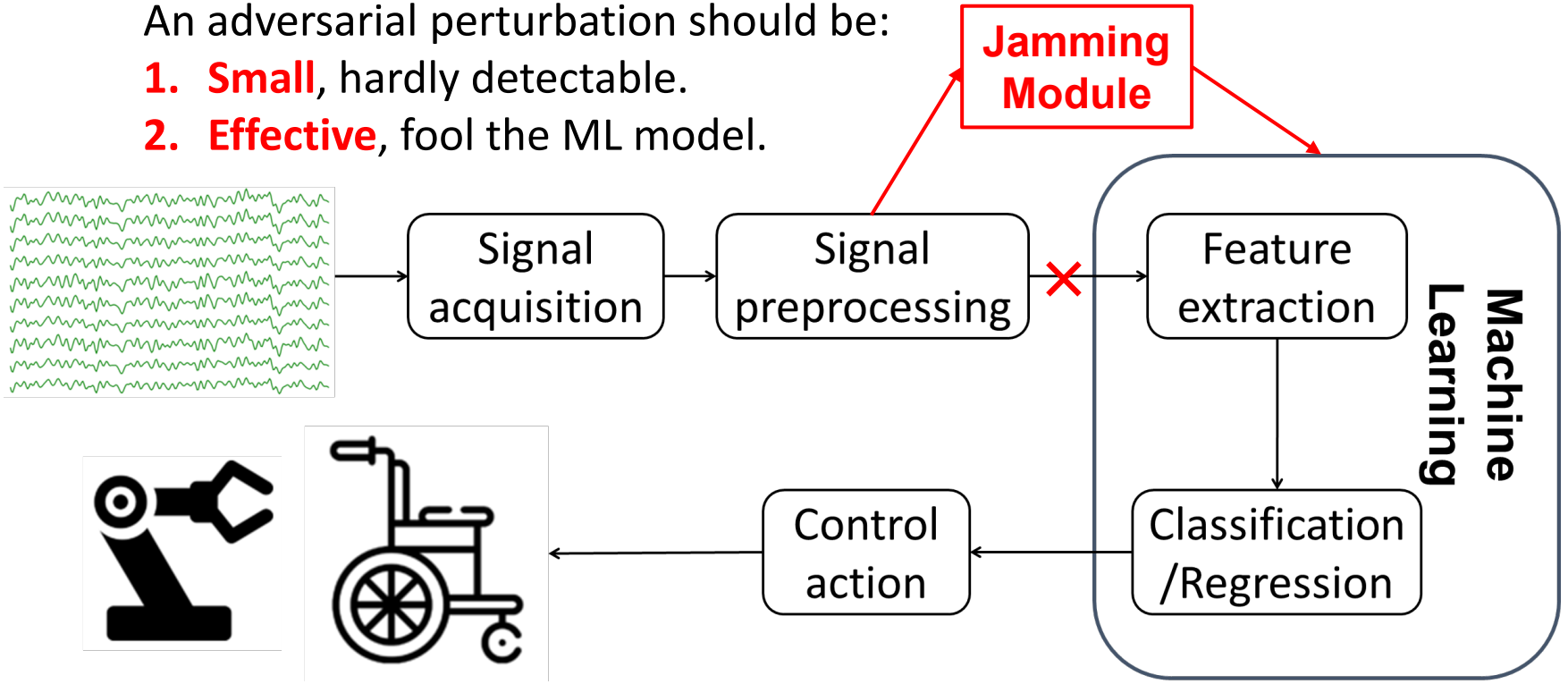
- **White-Box Attacks**: The attacker has access to all information of the target model, including its architecture and parameters.
- **Black-Box Attacks**: The attacker knows neither the architecture nor the parameters of the target model, but can observe its responses to inputs.

Target model information	White-Box	Black-Box
Know its architecture	✓	×
Know its parameters $\boldsymbol{\theta}$	✓	×
Can observe its response	–	✓

Adversarial Attacks in BCIs

An adversarial perturbation should be:

1. **Small**, hardly detectable.
2. **Effective**, fool the ML model.



Applications:

- **EEG-based BCIs controlled wheelchairs or exoskeleton**: User confusion and frustration, significantly reduce the user's quality of life, and even hurt the user by driving him/her into danger on purpose.
- **Clinical applications of BCIs in awareness evaluation/detection for disorder of consciousness patients**: Misdiagnosis.

3 Strategies in Black-Box Attacks

- **Black-Box Attack:** The attacker ~~knows the architecture and parameters of the target model.~~ can only sends input to the target model and observe its output.
- 3 Strategies:

  1) **Decision-based**: Gradually reduce the magnitude of the adversarial perturbation while ensuring its effectiveness

  2) **Score-based**: Use the model's output scores, e.g., class probabilities or logits, to estimate the gradients and then generate adversarial examples

  3) **Transferability-based**: Train a substitute model, which solves the same classification problem as the target model, to generate adversarial examples for the target model

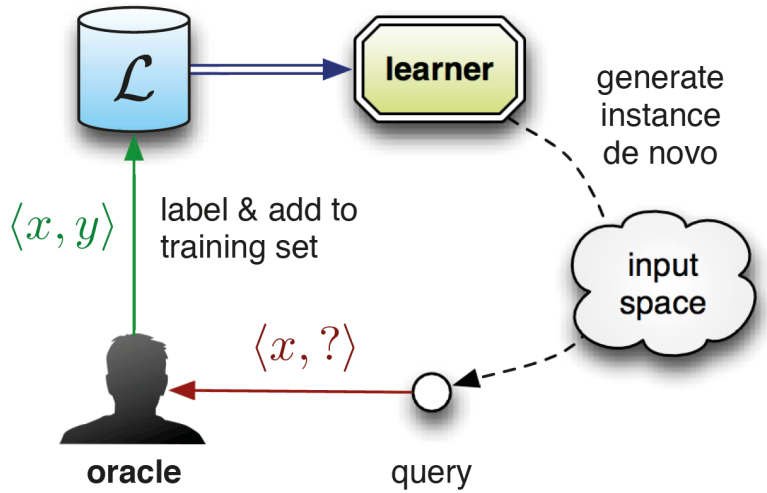
Transferability-Based Black-Box Attacks

- Our previous approach has 3 steps:
  1) Query the target model to obtain some input-output pairs
  2) Train a substitute model
  3) Use UFGSM to generate adversarial examples.

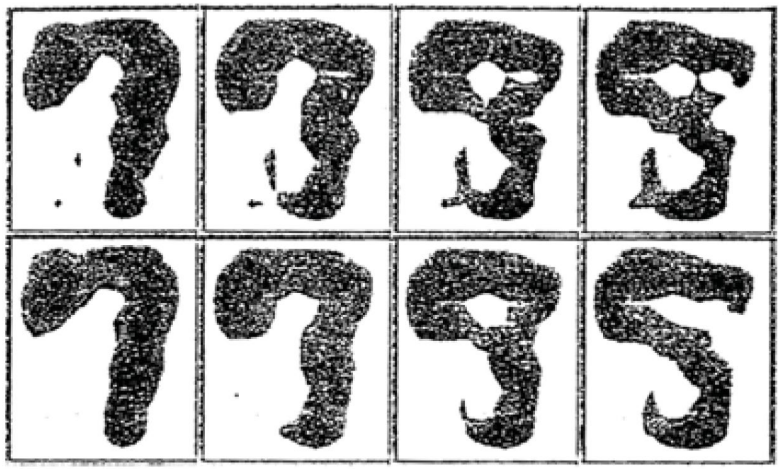
- How to reduce the number of queries?

Query Synthesis Based AL

- Can query any sample in the input space, including synthesized ones



(a) query synthesis



(b) an example from handwriting recognition

- Not popular in traditional classification tasks, because the synthesized samples may be hardly recognizable by a human
- Very suitable for query generation in constructing the substitute model in black-box attacks: The target model can label any inputs

Our Query Synthesis Based AL

1. Binary Search Synthesis

Assume a small labeled training set S_0 has been obtained from querying the target model. An initial substitute model f_0' can be trained on this set. Suppose $\{\mathbf{x}_0^+, \mathbf{x}_0^-\}$ is an opposite-pair in S_0. Then, we query their middle point on the substitute model to find another opposite-pair closer to the decision boundary.

2. Mid-Perpendicular Synthesis

- If we always use binary search to generate training epochs, they may concentrate in one area and lack diversity.
- We synthesize the next query along the mid-perpendicular direction after we find an opposite-pair close enough to the decision boundary.

X. Jiang, X. Zhang and D. Wu, Active Learning for Black-Box Adversarial Attacks in EEG-Based Brain-Computer Interfaces, *IEEE Symposium Series on Computational Intelligence*, Xiamen, China, Dec. 2019.

Mid-Perpendicular Synthesis

Algorithm 2: Mid-perpendicular synthesis. $\mathbf{x}_s = MidPerp(\{\mathbf{x}_b^+, \mathbf{x}_b^-\}, f', k, q)$

Input: $\{\mathbf{x}_b^+, \mathbf{x}_b^-\}$, an opposite-pair of EEG epochs; f', current substitute model; m, maximum number of binary search iterations; q, magnitude of the orthogonal vector.

Output: $\mathbf{x}_s$, an synthesized EEG epoch.

$\mathbf{x}_1 = \mathbf{x}_b^+ - \mathbf{x}_b^-$;
Generate an EEG epoch $\mathbf{x}_2$ randomly;
Find the orthogonal direction by Gram-Schmidt process:
$\mathbf{x}_2 = q \cdot (\mathbf{x}_2 - \langle \mathbf{x}_1, \mathbf{x}_2 \rangle / \langle \mathbf{x}_1, \mathbf{x}_2 \rangle \times \mathbf{x}_1)$;
$\{\mathbf{x}^+, \mathbf{x}^-\} = BinarySearch(\{\mathbf{x}_b^+, \mathbf{x}_b^-\}, f', m)$;
$\mathbf{x}_s = \mathbf{x}_2 + (\mathbf{x}^+ + \mathbf{x}^-)/2$.

return $\mathbf{x}_s$

Attack Performance

Dataset	Target Model f	Baselines		Method	Substitute Model f'		
		Original	Noisy		EEGNet	DeepCNN	ShallowCNN
P300	EEGNet	73.59/71.95	73.52/71.83	Ours	**41.40/42.44**	**32.66/39.75**	**64.95/64.91**
				Jacobian-based	47.02/48.49	39.79/41.99	65.16/64.16
	DeepCNN	75.99/74.10	76.20/73.94	Ours	**53.29/55.10**	**36.57/44.55**	**68.78/66.88**
				Jacobian-based	59.18/60.01	44.54/49.66	70.05/66.98
	ShallowCNN	72.23/71.90	72.24/71.85	Ours	**59.66/60.17**	51.76/**55.15**	53.80/**49.60**
				Jacobian-based	59.75/61.73	**51.40**/57.91	**52.38**/52.54
ERN	EEGNet	73.89/72.94	73.23/72.72	Ours	**45.71/47.29**	**46.78/48.94**	71.27/71.14
				Jacobian-based	54.42/54.47	52.54/54.53	**71.22/70.97**
	DeepCNN	74.24/72.69	73.86/72.38	Ours	**56.78/55.07**	**53.85/53.33**	**72.44/70.89**
				Jacobian-based	59.64/58.32	57.77/57.54	72.73/71.43
	ShallowCNN	71.86/71.45	71.77/71.21	Ours	**70.15**/70.46	**68.49/69.28**	**58.28/59.42**
				Jacobian-based	70.44/**70.30**	70.31/70.18	63.51/63.80
MI	EEGNet	60.85/60.71	59.48/59.39	Ours	**35.10/35.27**	**46.47/46.45**	**43.60/43.67**
				Jacobian-based	39.13/39.13	46.58/46.61	53.19/53.17
	DeepCNN	55.83/55.57	55.65/55.39	Ours	**47.06/46.90**	**45.32/45.31**	**42.54/42.46**
				Jacobian-based	48.68/48.59	48.45/48.38	48.72/48.57
	ShallowCNN	64.71/64.62	64.18/64.09	Ours	**57.26/57.32**	**57.94/57.98**	**46.74/46.84**
				Jacobian-based	59.44/59.49	59.04/59.07	54.56/54.56

Attack Performance vs # of Queries

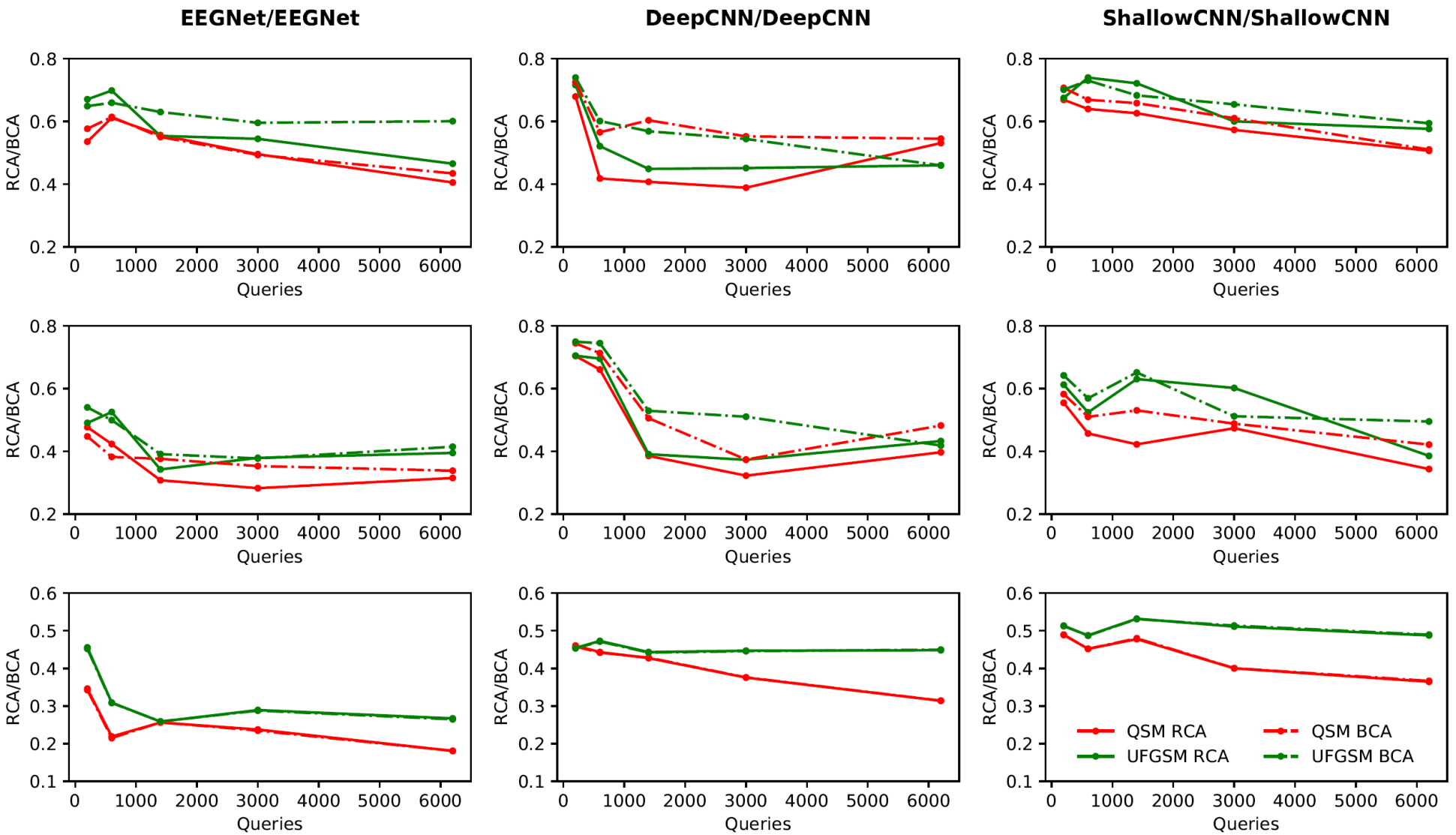
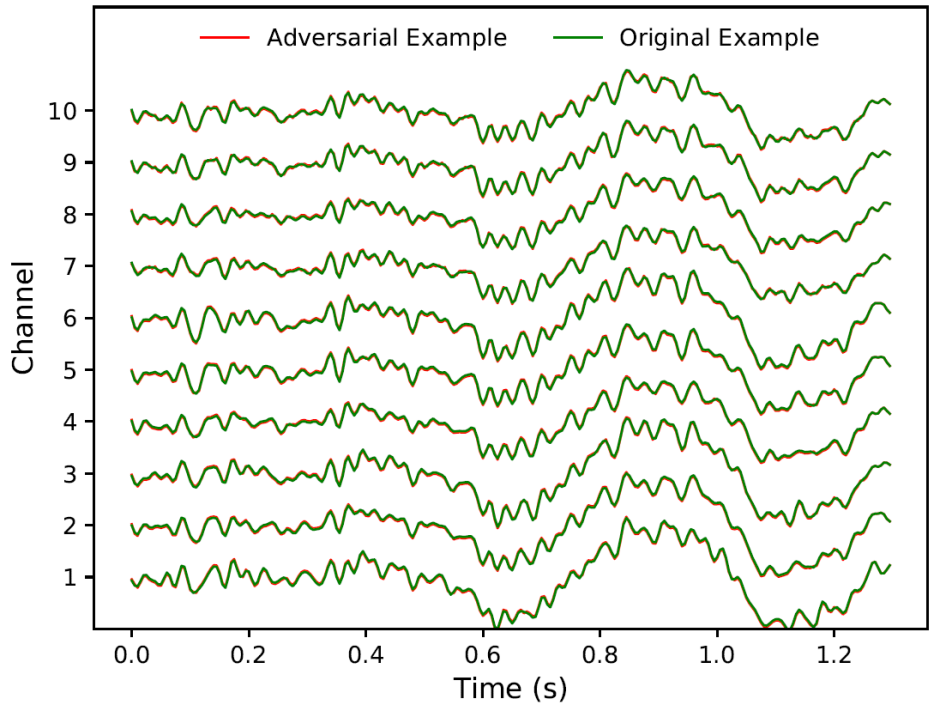
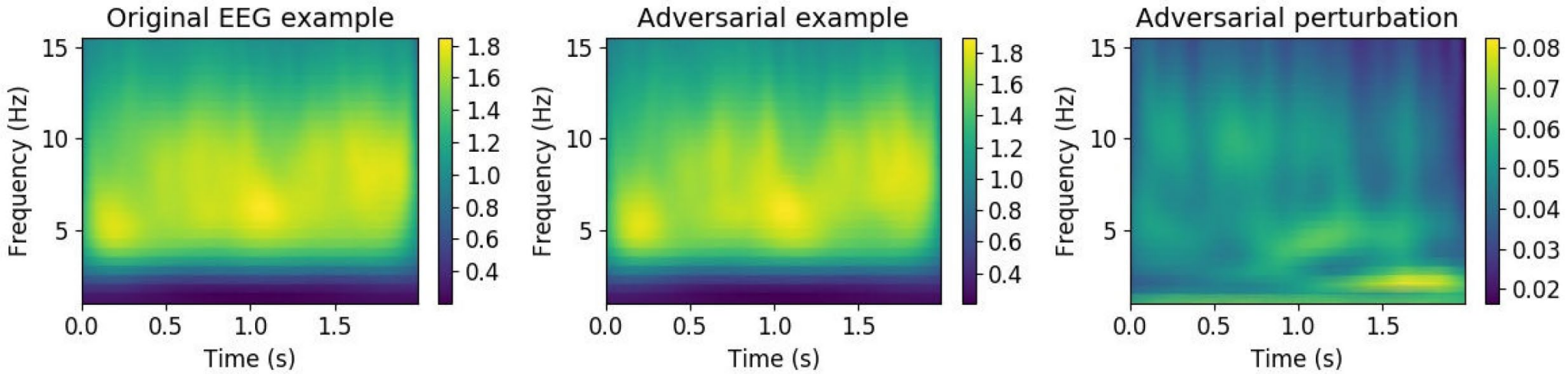


Fig. 1. RCAs and BCAs of the two black-box attack methods with different number of queries on P300 (top row), ERN (middle row) and MI (bottom row).

Visualization of the Perturbations





Outline

- Weakly Supervised Learning
- Active Learning
- Active Learning for Classification
- Active Learning for Regression
- Deep Active Learning
- Applications
- **Conclusions**

Conclusions

- **Active Learning Goal:** Minimize the number of queries such that the labeling cost for training a good model can be minimized.
- **Typical considerations:**
  - ✓ **Informativeness**: Measured by uncertainty (entropy, distance to the decision boundary, confidence of the prediction, etc.), expected model change, expected error reduction, etc.
  - ✓ **Representativeness**: Evaluated by the number of samples that are similar or close to a target sample (or its density)
  - ✓ **Diversity**: The selected sample should scatter across the full feature space, instead of concentrating in a small local region

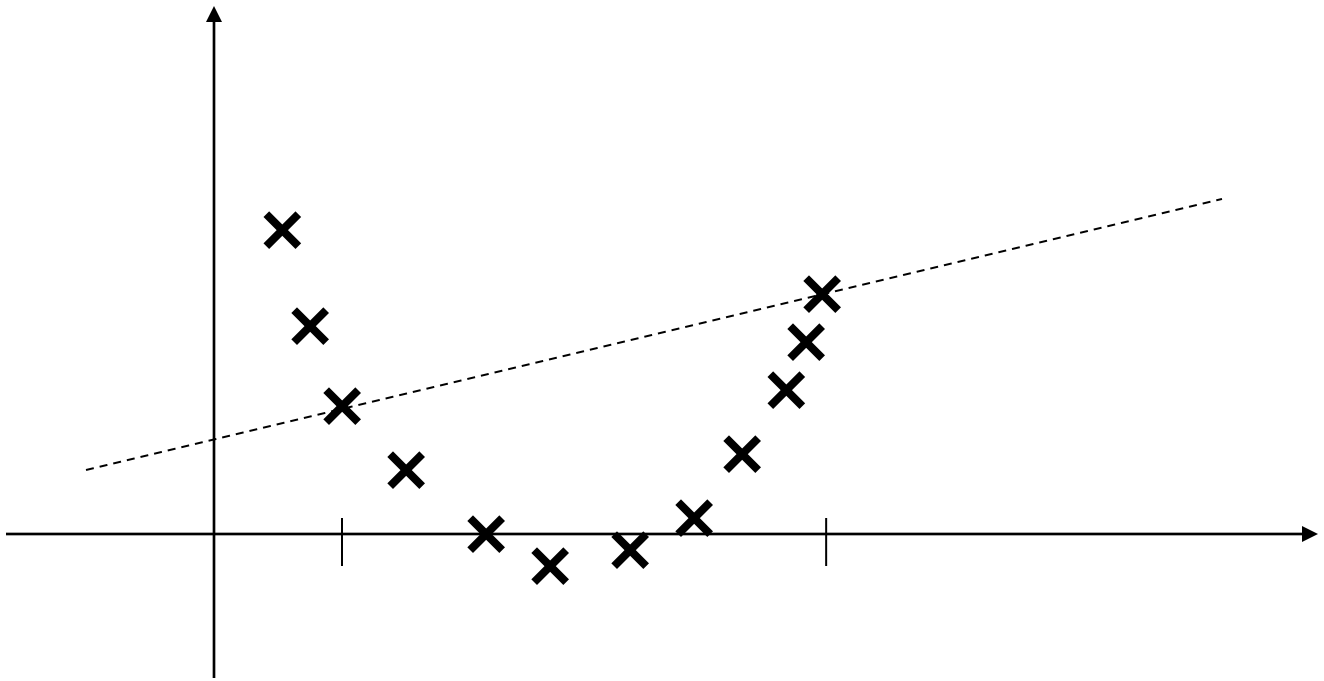
References

1. D. Wu and J. Huang*, "Affect Estimation in 3D Space Using Multi-Task Active Learning for Regression," *IEEE Trans. on Affective Computing*, 13(1):16-27, 2022.
2. D. Wu*, C-T Lin and J. Huang*, "Active learning for regression using greedy sampling," *Information Sciences*, 474:90-105, 2019.
3. D. Wu, "Pool-based sequential active learning for regression," *IEEE Trans. on Neural Networks and Learning Systems*, 30(5):1348-1359, 2019.
4. D. Wu, V.J. Lawhern, W.D. Hairston and B.J. Brent, "Switching EEG headsets made easy: Reducing offline calibration effort using active weighted adaptation regularization," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 24(11):1125-1137, 2016.
5. X. Jiang, L. Meng, S. Li and D. Wu*, "Active Poisoning: Efficient Backdoor Attacks to Transfer Learning Based BCIs," *Science China Information Sciences*, 2023, in press.
6. Z. Liu, X. Jiang, H. Luo, W. Fang, J. Liu and D. Wu*, "Pool-Based Unsupervised Active Learning for Regression Using Iterative Representativeness-Diversity Maximization (iRDM)," *Pattern Recognition Letters*, 142:11-19, 2021.
7. 刘子昂, 蒋雪, 伍冬睿*, "基于池的无监督线性回归主动学习," 自动化学报, 47(21):2771-2783, 2021.
8. Y. Xu, L. Meng, R. Peng, Y. Yin, J. Ding, L. Li and D. Wu, "Cross-Modal Diversity-Based Active Learning for Multi-Modal Emotion Estimation," *IEEE Int'l Joint Conf. on Neural Networks (IJCNN),* Golden Coast, Australia, June 2023.
9. X. Jiang, L. Meng, J. Huang and D. Wu, "Multi-Task Active Learning for Simultaneous Emotion Classification and Regression," *IEEE Int'l Conf. on Systems, Man and Cybernetics*, virtual, October 2021.
10. D. Wu, C. Guo, F. Liu and C. Liu, "Active Stacking for Heart Rate Estimation," *Int'l Joint Conf. on Neural Networks*, Glasgow, UK, July 2020.
11. Z. Liu and D. Wu, "Integrating Informativeness, Representativeness and Diversity in Pool-Based Sequential Active Learning for Regression," *Int'l Joint Conf. on Neural Networks*, Glasgow, UK, July 2020.
12. X. Jiang, X. Zhang and D. Wu*, "Active Learning for Black-Box Adversarial Attacks in EEG-Based Brain-Computer Interfaces," *IEEE Symposium Series on Computational Intelligence*, Xiamen, China, 2019.
13. D. Wu, "Active semi-supervised transfer learning (ASTL) for offline BCI calibration," *IEEE Int'l. Conf. on Systems, Man and Cybernetics*, Banff, Canada, 2017.
14. D. Wu, V. Lawhern, S. Gordon, B. Lance and C-T Lin, "Offline EEG-based driver drowsiness estimation using enhanced batch-mode active learning (EBMAL) for regression," *IEEE Int'l Conf. on Systems, Man and Cybernetics*, Budapest, Hungary, 2016.
15. D. Wu, B. Lance, and V. Lawhern, "Transfer learning and active transfer learning for reducing calibration data in single-trial classification of visually-evoked potentials," *IEEE Int'l Conf. on Systems, Man, and Cybernetics*, San Diego, CA, October 2014.
16. D. Wu and T.D. Parsons, "Active class selection for arousal classification," *Affective Computing and Intelligent Interaction Conf.,* Memphis, TN, October 2011.

Thank you!

Active Learning Warning

- Choice of data is only as good as the model itself
- Assume a linear model, then two samples are sufficient
- What happens when data are not linear?

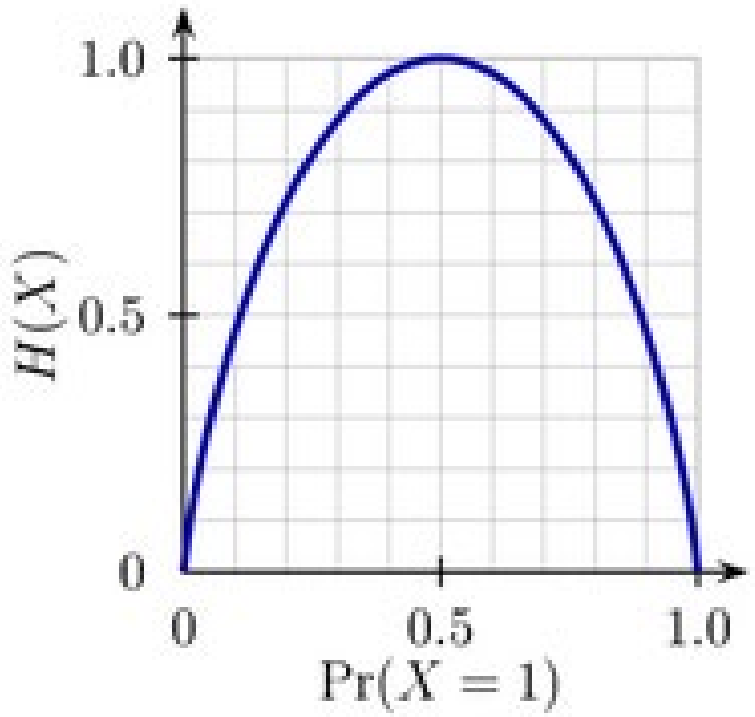


Entropy Function

- A measure of information in random event X with possible outcomes $\{x_1, \ldots, x_N$

$$H(x) = -\sum_{i=1}^{N} p(x_i) \log_2 p(x_i)$$

- Comments on entropy function:
  - Entropy of an event is zero when the outcome is known
  - Entropy is maximal when all outcomes are equally likely
- The minimum number of yes/no questions to answer some question
  - Related to binary search



[Shannon, 1948]

Kullback Leibler (KL) Divergence

- P = True distribution;
- Q = Alternative distribution that is used to encode data
- KL divergence is the expected extra message length per datum that must be transmitted using Q

$$D_{KL}(P||Q) = \sum_{i=1}^{N} P(x_i) \log(\frac{P(x_i)}{Q(x_i)}$$

$$= \sum_{i=1}^{N} P(x_i) \log P(x_i) - \sum_{i=1}^{N} P(x_i) \log Q(x_i)$$

$$= H(P,Q) - H(P)$$

Cross-entropy - entropy

- **Measures how different the two distributions are**

KL Divergence Properties

- Non-negative:

$$D(P||Q) \geq 0$$

- Divergence *0* if and only if P and Q are equal*:*

$$D(P||Q) = 0 \; iff \; P = Q$$

$$D(P||Q) \neq D(Q||P)$$

- Non-symmetric*:*

$$D(P||Q) \nleq D(P||R) + D(R||Q)$$

Not a distance metric!

- Does **not** satisfy triangle inequality

KL Divergence as Gain

- KL divergence of the posteriors measures the amount of information gain expected from query (x' is the queried data):

$$D(p(\theta|x,x')||p(\theta|x))$$

- **Goal**: Choose a query that *maximizes* the KL divergence between the posteriors after and before the query
- **Basic idea**: Largest KL divergence between updated posterior probability and the current posterior probability represents largest gain

Reminder: Risk Function

- Given an estimation procedure/decision function d
- Frequentist risk given the true parameter θ is the expected loss after seeing new data.

$$R(\theta, d) = \sum_{x_{new}} L(\theta, d(x_{new}))p(x_{new}|\theta)$$

- Bayesian integrated risk given a prior π is defined as the posterior expected loss:

$$R(\pi, d|x) = \sum_{\theta} L(\theta, d(x))p(\theta|x, \pi)$$

- Loss includes cost of query, prediction error, etc.