Rethinking the generalization of drug target affinity prediction algorithms via Similarity Aware Evaluation

Chenbin Zhang[*1], **Zhiqiang Hu**[*✉2], **Chuchu Jiang**[*3], **Wen Chen**[4], **Jie Xu**[3], **Shaoting Zhang**[3]

[1]MoleculeMind, [2]Peking University, [3]Shanghai AI Laboratory, [4]SenseTime Research

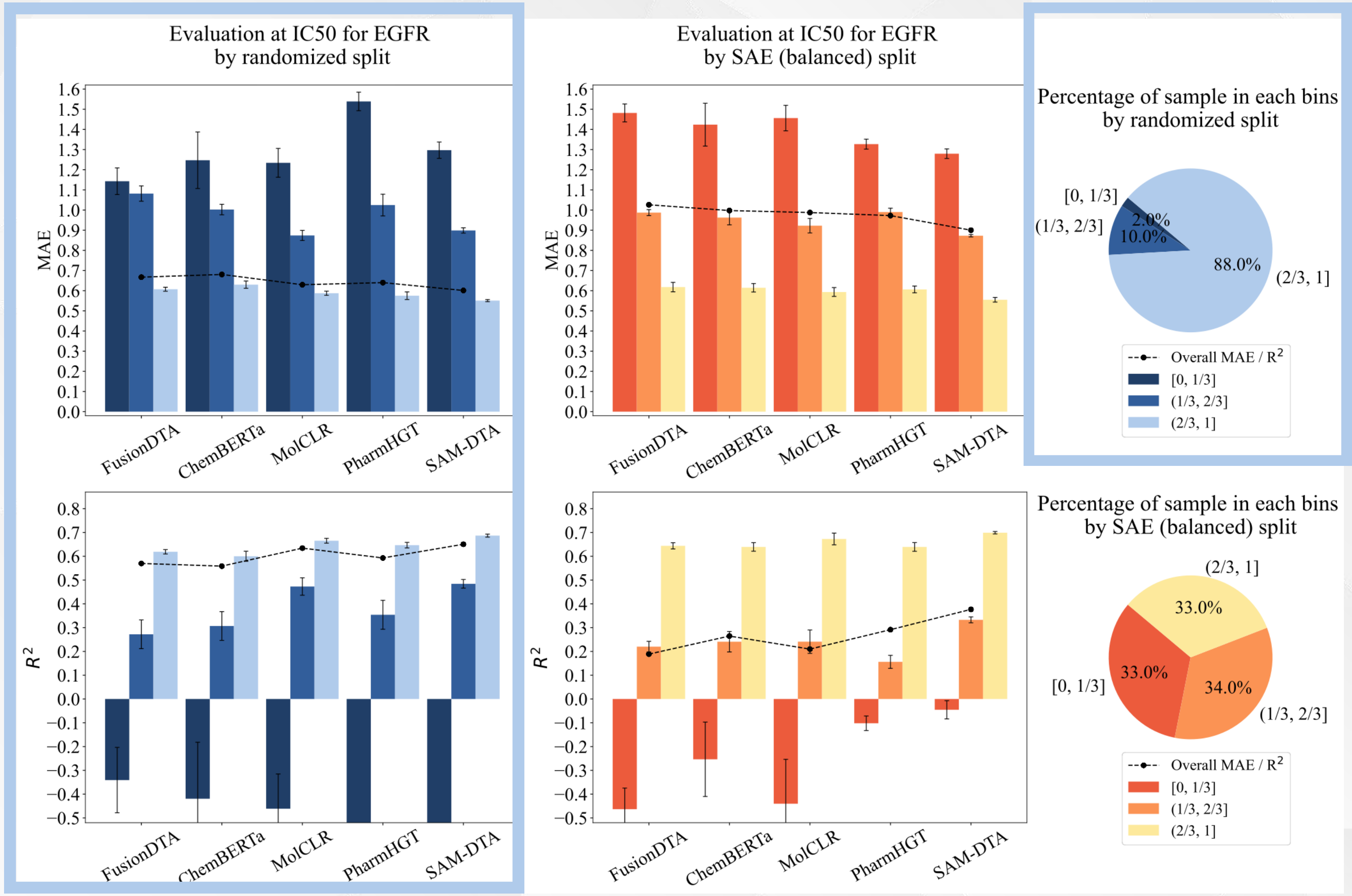
chenbinzhang@moleculemind.com, huzq@pku.edu.cn,
{jiangchuchu,xujie,zhangshaoting}@pjlab.org.cn,
chenwen@sensetime.com

https://github.com/Amshoreline/SAE

1 / Introduction

Insight

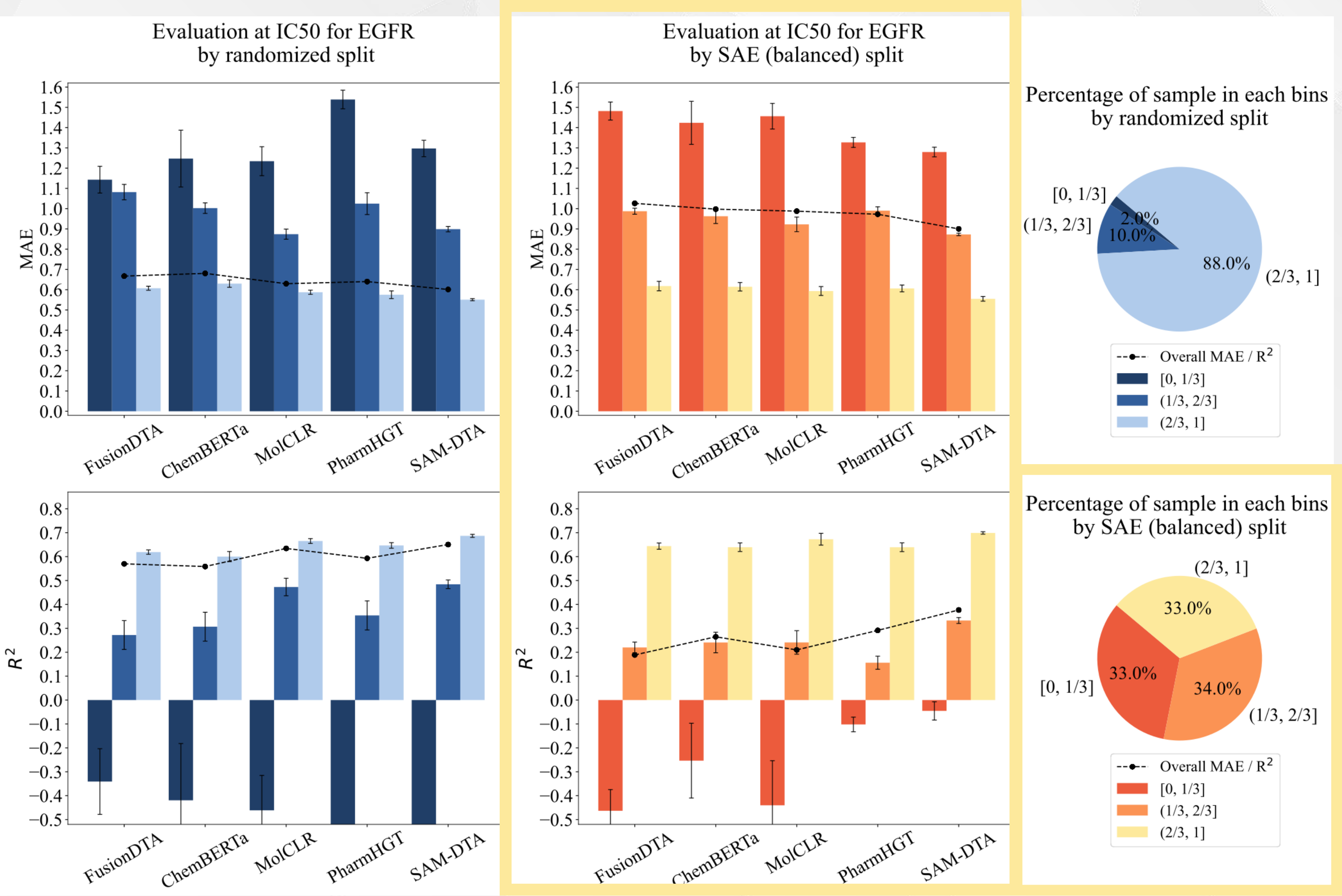
1. The performance of models is severely degraded on samples with lower similarity to the training set.



1 / Introduction

Insight

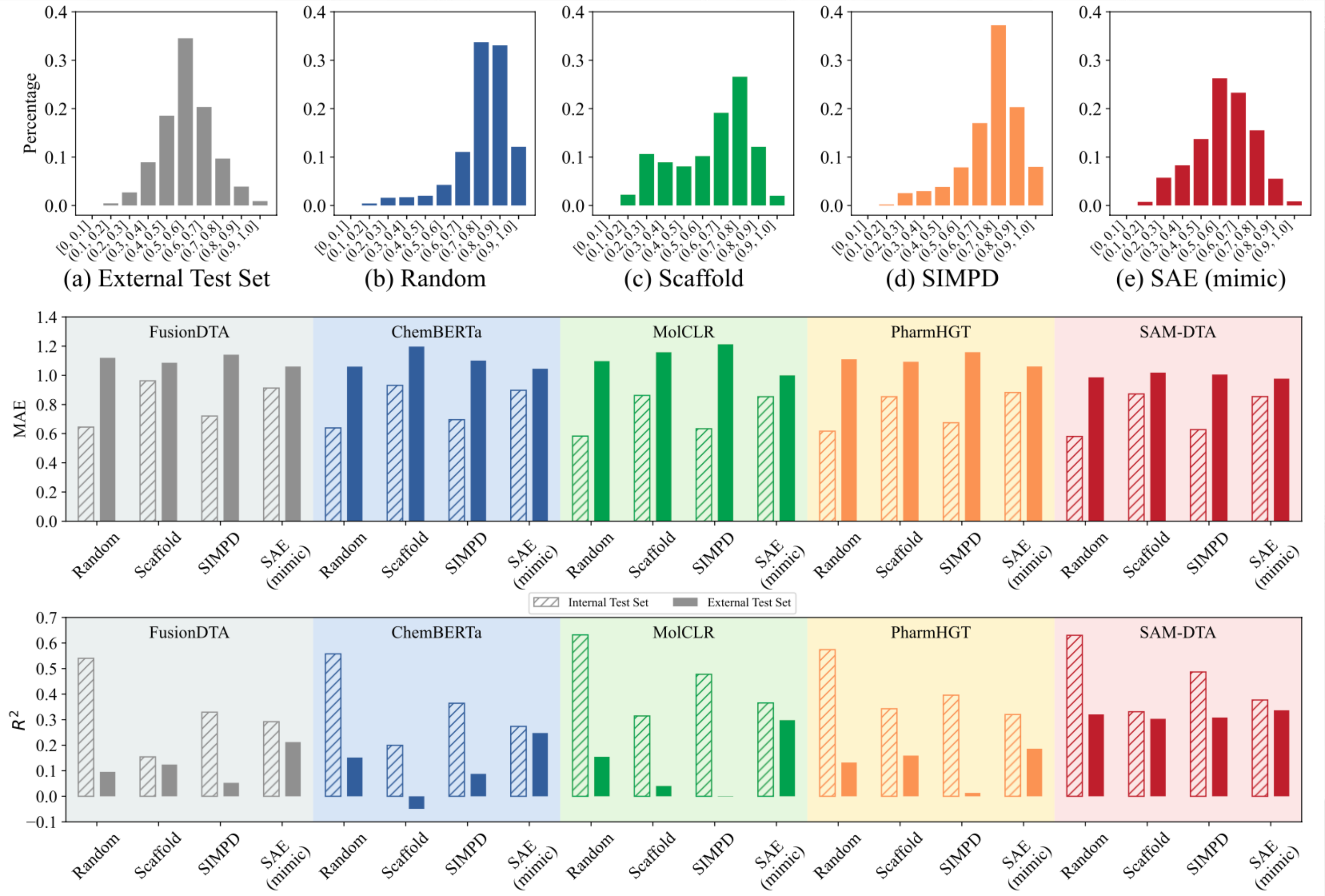
1. The performance of models is severely degraded on samples with lower similarity to the training set.
2. We propose a novel split methodology to adapt to any desired distribution.



2 / Results

Mimic Split

1. Split the training/internal test sets based on the distribution of the external test set.
2. The internal test set is used for hyperparameter search.
3. The external test performance obtained by SAE (mimic) is the best, and its performance is the closest to that of the internal test.



2 / Results

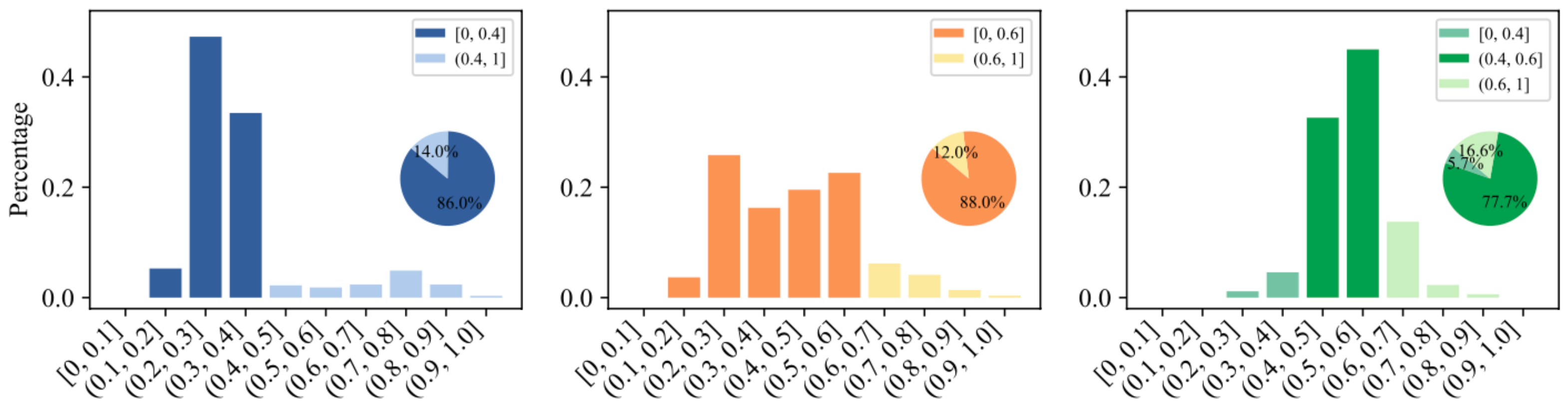
Balanced split

1. Similarity Measure
   - Cosine
   - Sokal
   - Dice
   - Tanimoto
2. Fingerprint
   - Morgan
   - RDKFP
   - Avalon

Similarity Measure	Fingerprint	SAE (balanced)	Random	Scaffold	SIMPD	Stratified (max)	Stratified (avg)
Cosine	Morgan	145, 436, 292	0, 33, 840	6, 159, 708	16, 657, 200	0, 32, 841	0, 27, 846
	RDKFP	18, 426, 429	0, 17, 856	1, 78, 794	1, 124, 748	0, 13, 860	1, 14, 858
	Avalon	9, 429, 435	0, 7, 866	0, 21, 852	0, 16, 857	0, 6, 867	0, 6, 867
Sokal	Morgan	292, 289, 292	33, 398, 442	172, 510, 191	689, 126, 58	34, 423, 416	29, 416, 428
	RDKFP	291, 291, 291	19, 80, 774	85, 236, 552	135, 624, 114	14, 82, 777	15, 74, 784
	Avalon	291, 291, 291	7, 63, 803	26, 275, 572	23, 629, 221	8, 76, 789	6, 73, 794
Dice	Morgan	182, 378, 313	0, 33, 840	9, 163, 701	17, 672, 184	0, 34, 839	2, 27, 844
	RDKFP	60, 463, 350	0, 19, 854	2, 83, 788	1, 134, 738	0, 14, 859	1, 14, 858
	Avalon	32, 416, 425	0, 7, 866	0, 26, 847	0, 23, 850	0, 8, 865	0, 6, 867
Tanimoto	Morgan	290, 299, 284	16, 98, 759	80, 273, 520	228, 547, 98	12, 99, 762	13, 99, 761
	RDKFP	289, 292, 292	8, 34, 831	15, 184, 674	2, 635, 236	1, 39, 833	4, 33, 836
	Avalon	220, 325, 328	0, 28, 845	2, 154, 717	1, 324, 548	0, 30, 843	1, 28, 844

Other Applications

1. 0~0.4 split
2. 0~0.6 split
3. 0.4~0.6 split



(a) Test data distribution with 0~0.4 split

(b) Test data distribution with 0~0.6 split

(c) Test data distribution with 0.4~0.6 split

Formulation of SAE (balanced split)

- Given number of samples N, and K bins with boundaries $\{b_0, \dots, b_K\}$, we define the combinatorial optimization problem as:

$$f(X_{ts}) = \sum_{k=1}^{K} \frac{(o_k - \alpha N/K)^2}{\alpha N/K}$$

$$o_k = |\{x_i \in X_{ts} : b_{k-1} < r_i \le b_k\}|$$

$$r_i = \max_{x_j \in X_{tr}} s_{ij}, \qquad X_{tr} = X - X_{ts}$$

  - s_{ij}: pair-wise similarity matrix
  - α : ratio of the test set

3 / Method

Formulation of SAE (balanced split)

- Given number of samples N, and K bins with boundaries $\{b_0, \ldots, b_K\}$, we define the combinatorial optimization problem as:

$$f(X_{ts}) = \sum_{k=1}^{K} \frac{(o_k - \alpha N/K)^2}{\alpha N/K}$$

$$o_k = |\{x_i \in X_{ts}: b_{k-1} < r_i \leq b_k\}|$$

$$r_i = \max_{x_j \in X_{tr}} s_{ij}, \qquad X_{tr} = X - X_{ts}$$

  - s_{ij}: pair-wise similarity matrix
  - α : ratio of the test set
- Then, we relax it to a continuous optimization problem by introducing a “test” weight ω_i for each sample, which adheres to the constraints:

$$\sum_{i=1}^{N} \omega_i = \alpha N, 0 \leq \omega_i \leq 1$$

Formulation of SAE (balanced split)

- Denote $c_k = \frac{(b_{k-1}+b_k)}{2}$ as the center of each bin, the optimization problem can be approximated as:

$$f(X_{ts}) = \sum_{k=1}^{K} \frac{(o_k - \alpha N/K)^2}{\alpha N/K}$$

$$o_k = |\{x_i \in X_{ts}: b_{k-1} < r_i \leq b_k\}| = \sum_i \omega_i \mathbb{I}\,(b_{k-1} < r_i \leq b_k)$$

$$\approx \sum_i \omega_i \frac{\exp(-(r_i - c_k)^2/(2\sigma^2))}{\sum_{k'} \exp(-(r_i - c_{k'})^2/(2\sigma^2))} = \sum_i \omega_i\, softmax_k\left(-\frac{(r_i - c_k)^2}{2\sigma^2}\right)$$

$$r_i = \max_{x_j \in X_{tr}} s_{ij} = \max_j (1 - \omega_j) s_{ij} \approx \frac{1}{\beta} \log \sum_j \exp(\beta(1 - \omega_j) s_{ij})$$

3 / Method

Formulation of SAE (balanced split)

- Denote $c_k = \frac{(b_{k-1}+b_k)}{2}$ as the center of each bin, the optimization problem can be approximated as:

$$f(X_{ts}) = \sum_{k=1}^{K} \frac{(o_k - \alpha N/K)^2}{\alpha N/K}$$

$$o_k = |\{x_i \in X_{ts}: b_{k-1} < r_i \le b_k\}| = \sum_i \omega_i \mathbb{I}\,(b_{k-1} < r_i \le b_k)$$

$$\approx \sum_i \omega_i \frac{\exp(-(r_i - c_k)^2/(2\sigma^2))}{\sum_{k'} \exp(-(r_i - c_{k'})^2/(2\sigma^2))} = \sum_i \omega_i\, softmax_k\left(-\frac{(r_i - c_k)^2}{2\sigma^2}\right)$$

$$r_i = \max_{x_j \in X_{tr}} s_{ij} = \max_j (1-\omega_j) s_{ij} \approx \frac{1}{\beta} \log \sum_j \exp(\beta(1-\omega_j) s_{ij})$$

- Considering the ideal value of ω_i is neither near 0 nor 1, we propose to add a regularization term:

$$l_{reg} = -\lambda \sum_i (\omega_i \log(\omega_i) + (1-\omega_i)\log(1-\omega_i))$$

Formulation of SAE (balanced split)

- Finally, we have the optimization problem:

$$minimize_{\omega_i} \sum_{k=1}^{K} \frac{(o_k - \alpha N/K)^2}{\alpha N/K} + l_{reg}$$

$$subject\ to\ \sum_{i=1}^{N} \omega_i = \alpha N, 0 \le \omega_i \le 1$$

where

$$o_k = \sum_i \omega_i\, softmax_k\left(-\frac{(r_i - c_k)^2}{2\sigma^2}\right)$$

$$r_i = \frac{1}{\beta} \log \sum_j \exp(\beta(1 - \omega_j) s_{ij})$$

$$l_{reg} = -\lambda \sum_i (\omega_i \log(\omega_i) + (1 - \omega_i) \log(1 - \omega_i))$$

3 / Method

Formulation of SAE (balanced split)

- Finally, we have the optimization problem:

$$minimize_{\omega_i} \sum_{k=1}^{K} \frac{(o_k - \alpha N/K)^2}{\alpha N/K} + l_{reg}$$

$$subject\ to\ \sum_{i=1}^{N} \omega_i = \alpha N, 0 \leq \omega_i \leq 1$$

where

$$o_k = \sum_i \omega_i\, softmax_k \left(-\frac{(r_i - c_k)^2}{2\sigma^2}\right)$$

$$r_i = \frac{1}{\beta} \log \sum_j \exp(\beta(1 - \omega_j) s_{ij})$$

$$l_{reg} = -\lambda \sum_i (\omega_i \log(\omega_i) + (1 - \omega_i) \log(1 - \omega_i))$$

- If the expected count in each bin is e_k, the objective function can be readily modified as:

$$\sum_{k=1}^{K} \frac{(o_k - e_k)^2}{e_k} + l_{reg}$$

THANKS