Appendix for "Can Infinitely Wide Deep Nets Help Small-data Multi-label Learning?"

A Proof of Proposition 1

Proposition 1. *Consider minimizing $C(\boldsymbol{\theta})$ in Eq.*(3) by gradient descent with infinitesimally small learning rate (i.e., gradient flow): $\frac{\mathrm{d}\boldsymbol{\theta}(t)}{\mathrm{d}t} = -\nabla C(\boldsymbol{\theta}(t))$. Let $\mathbf{F}(t) = [f(\mathbf{x}_1, \boldsymbol{\theta}(t)); ...; f(\mathbf{x}_n, \boldsymbol{\theta}(t))] \in \mathbb{R}^{n \times c}$ be network outputs on all $\mathbf{x}_i$'s at time t. When the hidden widths $d_1, ..., d_L \longrightarrow \infty$, then $\mathbf{F}(t)$ follows the following evolution:

$$\frac{\mathrm{d}\mathbf{F}(t)}{\mathrm{d}t} = -\frac{1}{n}\mathbf{K} \cdot \frac{\partial \ell(t)}{\partial \mathbf{F}(t)}, \tag{1}$$

where $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the NTK kernel matrix and $\frac{\partial \ell(t)}{\partial \mathbf{F}(t)} \triangleq \left[\frac{\partial \ell(f(\mathbf{x}_1, \boldsymbol{\theta}(t)), \mathbf{y}_1)}{\partial f(\mathbf{x}_1, \boldsymbol{\theta}(t))}; ...; \frac{\partial \ell(f(\mathbf{x}_n, \boldsymbol{\theta}(t)), \mathbf{y}_n)}{\partial f(\mathbf{x}_n, \boldsymbol{\theta}(t))}\right] \in \mathbb{R}^{n \times c}$.

Proof. First, when discarding the infinity width assumption, we can get the dynamics of $\mathbf{F}(t)$ w.r.t. a matrix-based kernel matrix $\Theta^{(L)}(\boldsymbol{\theta}) \in \mathbb{R}^{nc \times nc}$, where its corresponding matrix-based kernel function is $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{c \times c}$. Besides, the kernel matrix $\Theta^{(L)}(\boldsymbol{\theta})$ depends on the parameters $\boldsymbol{\theta}$ and varies during training.

Second, when adding the infinity width assumption, based on the Theorem 1 and 2 in [4], we can get the deterministic kernel matrix $\Theta^{(L)} = \mathbf{K} \otimes \mathbf{I}_c$, where $\mathbf{I}_c \in \mathbb{R}^{c \times c}$ is the identity matrix. The kernel matrix $\Theta^{(L)}$ doesn't vary during training and always equals the state in the random initialization. Besides, the matrix-based kernel is separable and can be equivalently transformed into the corresponding scalar-based kernel [1], which is used here. □

B Derivation of the gradient $\nabla g_i(\mathbf{W})$

Here, we take the logistic base loss function for example, which is used in our experiments.

Define

$$g_i(\mathbf{W}) = \frac{1}{c}\sum_{j=1}^{c} \log(1 + \exp(-y_{ij}\langle \mathbf{w}^j, z(\mathbf{x}_i)\rangle)) + \frac{\tau}{2}\|\mathbf{W}\|_F^2 + \bar{g}_i(\mathbf{W}), \tag{2}$$

where

$$\bar{g}_i(\mathbf{W}) = \frac{\lambda}{|Y_i^+||Y_i^-|}\sum_{p \in Y_i^+}\sum_{q \in Y_i^-} \log(1 + \exp(\langle \mathbf{w}^q - \mathbf{w}^p, z(\mathbf{x}_i)\rangle)). \tag{3}$$

Then, the gradient of $g_i(\mathbf{W})$ w.r.t. $\mathbf{W}$ is as follows:

$$\nabla g_i(\mathbf{W}) = \frac{1}{c}\mathbf{x}_i^\top\Big(\frac{1}{\mathbf{1} + \exp(-\mathbf{y}_i \circ (z(\mathbf{x}_i)\mathbf{W}))} \circ \exp(-\mathbf{y}_i \circ (z(\mathbf{x}_i)\mathbf{W}))\Big) + \tau\mathbf{W} + \nabla \bar{g}_i(\mathbf{W}), \tag{4}$$

where $\nabla \bar{g}_i(\mathbf{W}) = \left[\frac{\partial \bar{g}_i}{\partial \mathbf{w}^1}, ..., \frac{\partial \bar{g}_i}{\partial \mathbf{w}^c}\right]$ and for $j \in [c]$,

$$\frac{\partial \bar{g}_i}{\partial \mathbf{w}^j} = \frac{\lambda}{|Y_i^+||Y_i^-|}\Big\{[\![j \in Y_i^+]\!]\sum_{q \in Y_i^-}\frac{-z(\mathbf{x}_i)}{1 + \exp(\langle \mathbf{w}^j - \mathbf{w}^q, z(\mathbf{x}_i)\rangle)} + [\![j \in Y_i^-]\!]\sum_{p \in Y_i^+}\frac{z(\mathbf{x}_i)}{1 + \exp(\langle \mathbf{w}^p - \mathbf{w}^j, z(\mathbf{x}_i)\rangle)}\Big\}. \tag{5}$$

C Proofs for the generalization analyses

Technically, we provide the generalization analyses based on Rademacher complexity [2] and the recent vector-contraction inequality [5], following recent work [6].

First, we define the surrogate expected loss w.r.t. Hamming Loss (HL) and Ranking Loss (RL) as follows.

$$R^h(f) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim P}\Big[L_h(f(\mathbf{x}),\mathbf{y}_i)\Big], \qquad R^r(f) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim P}\Big[L_r(f(\mathbf{x}),\mathbf{y}_i)\Big]. \tag{6}$$

Then, we give the following for the subsequent analyses.

Theorem C.1 (The base theorem [6])**.** *Assume the loss function $L : \mathbb{R}^c \times \{-1,+1\}^c \to \mathbb{R}_+$ is μ-Lipschitz continuous w.r.t. the first argument and bounded by M. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of an i.i.d. sample S of size n, the following generalization bound holds for all $f \in \mathcal{F}$:*

$$R(f) \le \hat{R}_S(f) + 2\sqrt{2}\mu\hat{\mathfrak{R}}_S(\mathcal{F}) + 3M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}. \tag{7}$$

Lemma C.1 (The Rademacher complexity of the kernel-based hypothesis set)**.** *Consider the hypothesis set $\mathcal{F} = \{\mathbf{x} \mapsto [\langle f_1^\kappa, \Phi(\mathbf{x})\rangle, ..., \langle f_c^\kappa, \Phi(\mathbf{x})\rangle] : \sum_{j=1}^c \|f_j^\kappa\|_{\mathbb{H}}^2 \le \Lambda\}$, which can be equivalently transformed into the following form $\mathcal{F} = \{\mathbf{x} \mapsto \mathbf{W}^\top\phi(\mathbf{x}) : \mathbf{W} = (\mathbf{w}_1,\ldots,\mathbf{w}_c)^\top, \|\mathbf{W}\|_{\mathbb{H},2} \le \Lambda\}$, $\hat{\mathfrak{R}}_S(\mathcal{F})$, where $\|\mathbf{W}\|$ denotes $\|\mathbf{W}\|_{\mathbb{H},2}$ for convenience. Then, $\hat{\mathfrak{R}}_S(\mathcal{F})$ can be bounded bellow:*

$$\hat{\mathfrak{R}}_S(\mathcal{F}) \le \sqrt{\frac{c\max\{\mathbf{K}_{ii}\}\Lambda^2}{n}}. \tag{8}$$

Proof. For the kernel-based hypothesis set $\mathcal{F} = \{\mathbf{x} \mapsto \mathbf{W}^\top\phi(\mathbf{x}) : \mathbf{W} = (\mathbf{w}_1,\ldots,\mathbf{w}_c)^\top, \|\mathbf{W}\| \le \Lambda\}$, the following inequalities about $\hat{\mathfrak{R}}_S(\mathcal{F})$ hold:

$$\begin{aligned}
\hat{\mathfrak{R}}_S(\mathcal{F}) &= \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\sup_{\|\mathbf{W}\|\le\Lambda}\sum_{i=1}^n\sum_{j=1}^c \epsilon_{ij}\langle\mathbf{w}_j,\phi(\mathbf{x}_i)\rangle\Big] \\
&= \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\sup_{\|\mathbf{W}\|\le\Lambda}\sum_{j=1}^c\langle\mathbf{w}_j,\sum_{i=1}^n \epsilon_{ij}\phi(\mathbf{x}_i)\rangle\Big] \\
&= \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\sup_{\|\mathbf{W}\|\le\Lambda}\langle\mathbf{W},\mathbf{X}_\epsilon\rangle\Big] \qquad (\mathbf{X}_\epsilon = [\sum_{i=1}^n \epsilon_{i1}\phi(\mathbf{x}_i),\ldots,\sum_{i=1}^n \epsilon_{ic}\phi(\mathbf{x}_i)]) \\
&\le \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\sup_{\|\mathbf{W}\|\le\Lambda}\|\mathbf{W}\|\,\|\mathbf{X}_\epsilon\|\Big] \qquad (\textit{Cauchy-Schwarz Inequality}) \\
&= \frac{\Lambda}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\sum_{j=1}^c\|\sum_{i=1}^n \epsilon_{ij}\phi(\mathbf{x}_i)\|^2\Big]^{1/2} \\
&= \frac{\Lambda}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\sum_{j=1}^c\sum_{p=1}^n\sum_{q=1}^n \epsilon_{pj}\epsilon_{qj}\langle\phi(\mathbf{x}_p),\phi(\mathbf{x}_q)\rangle\Big]^{1/2} \\
&= \frac{\Lambda}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\sum_{j=1}^c\sum_{i=1}^n\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_i)\rangle\Big]^{1/2} \qquad (\forall p \ne q, \mathbb{E}[\epsilon_{pj}\epsilon_{qj}] = \mathbb{E}[\epsilon_{pj}]\mathbb{E}[\epsilon_{qj}] = 0 \textit{ and } \mathbb{E}[\epsilon_{ij}\epsilon_{ij}] = 1) \\
&= \frac{\Lambda\sqrt{c\,\mathrm{Tr}(\mathbf{K})}}{n} \qquad (\kappa(\mathbf{x}_i,\mathbf{x}_i) = \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_i)\rangle, \mathbf{K} = [\kappa(\mathbf{x}_i,\mathbf{x}_j)] \textit{ is the kernel matrix}) \\
&\le \sqrt{\frac{c\max\{\mathbf{K}_{ii}\}\Lambda^2}{n}}.
\end{aligned} \tag{9}$$

□

Lemma C.2 (The property of the surrogate loss function [6])**.** *Assume that the base loss function $\ell(u)$ is ρ-Lipschitz continuous and bounded by B. Then, the surrogate Hamming Loss L_h is $\frac{\rho}{\sqrt{c}}$-Lipschitz, and the surrogate Ranking Loss L_r is ρ-Lipschitz w.r.t. the first argument. Besides, they are all bounded by B.*

C.1 Proof of Theorem 5

Theorem 5 (Learning guarantee w.r.t. the Hamming Loss)**.** *Consider the hypothesis space $\mathcal{F}_h = \{f \in \mathcal{F} : \hat{R}_S^r(f) \le \Lambda_1\}$ and the loss function L_h. Besides,* Assumption 1 *holds. Then, for any $\delta > 0$, the following holds for any $f \in \mathcal{F}_h$ with* *probability at least* $1-\delta$

$$R^h_{0/1}(f) \leq \hat{R}^h_S(f) + 2\rho\sqrt{\frac{2}{c}}\hat{\mathfrak{R}}_S(\mathcal{F}_h) + 3B\sqrt{\frac{\log\frac{2}{\delta}}{2n}}, \tag{10}$$

where the empirical Rademacher complexity satisfies

$$\hat{\mathfrak{R}}_S(\mathcal{F}_h) \leq \hat{\mathfrak{R}}_S(\mathcal{F}) \leq \sqrt{\frac{c\max\{\mathbf{K}_{ii}\}\Lambda^2}{n}}. \tag{11}$$

Proof. Since $L = L_h$, we can get its Lipschitz constant (i.e. $\frac{\rho}{\sqrt{c}}$) and bounded value (i.e. B) from Lemma C.2. Then, applying Theorem C.1 and the inequality $R^h_{0/1}(sgn \circ f) \leq R^h(f)$, we can Eq.(10). Besides, it can be easily verified that $\hat{\mathfrak{R}}_S(\mathcal{F}_h) \leq \hat{\mathfrak{R}}_S(\mathcal{F})$. Then, based on Lemma C.1, we can get Eq.(11). □

C.2 Proof of Theorem 6

Theorem 6 (Learning guarantee w.r.t. the Ranking Loss)**.** *Consider the hypothesis space* $\mathcal{F}_r = \{f \in \mathcal{F} : \hat{R}^h_S(f) \leq \Lambda_2\}$ *and the loss function* L_r*. Besides,* Assumption 1 *holds. Then, for any* $\delta > 0$*, the following holds for any* $f \in \mathcal{F}_r$ *with probability at least* $1-\delta$

$$R^r_{0/1}(f) \leq \hat{R}^r_S(f) + 2\sqrt{2}\rho\hat{\mathfrak{R}}_S(\mathcal{F}_r) + 3B\sqrt{\frac{\log\frac{2}{\delta}}{2n}}, \tag{12}$$

where the empirical Rademacher complexity satisfies

$$\hat{\mathfrak{R}}_S(\mathcal{F}_r) \leq \hat{\mathfrak{R}}_S(\mathcal{F}) \leq \sqrt{\frac{c\max\{\mathbf{K}_{ii}\}\Lambda^2}{n}}. \tag{13}$$

Proof. Since $L = L_r$, we can get its Lipschitz constant (i.e. ρ) and bounded value (i.e. B) from Lemma C.2. Then, applying Theorem C.1 and the inequality $R^h_{0/1}(sgn \circ f) \leq R^h(f)$, we can Eq.(12). Besides, it can be easily verifed that $\hat{\mathfrak{R}}_S(\mathcal{F}_h) \leq \hat{\mathfrak{R}}_S(\mathcal{F})$. Then, based on Lemma C.1, we can get Eq.(13). □

D Additional experimental results

D.1 Results w.r.t. other measures

The experimental results w.r.t. the measures of Subset Accuracy, instance-F1, Coverage, and Average Precision are summarized in Table 1, 2, 3 and 4, respectively.

Table 1: Experimental results of benchmark approaches ($\mathrm{mean}_{\mathrm{std}}$) w.r.t. Subset Accuracy (↑) on all datasets. Best results are highlighted in bold.

Dataset	Rank-SVM	BP-MLL	BR-SVM	CPNL	MLFE	MLC-RBF	MLC-NTK
flags	$0.073_{0.025}$	$0.107_{0.015}$	$0.132_{0.041}$	$\mathbf{0.156_{0.053}}$	$0.151_{0.041}$	$0.144_{0.035}$	$0.153_{0.031}$
birds	$0.148_{0.018}$	$0.121_{0.019}$	$0.214_{0.028}$	$0.214_{0.026}$	$0.170_{0.018}$	$0.206_{0.024}$	$\mathbf{0.234_{0.025}}$
emotions	$0.291_{0.025}$	$0.268_{0.039}$	$0.313_{0.015}$	$\mathbf{0.324_{0.035}}$	$0.291_{0.035}$	$\mathbf{0.324_{0.025}}$	$0.310_{0.027}$
image	$0.451_{0.018}$	$0.217_{0.024}$	$0.482_{0.018}$	$0.533_{0.017}$	$0.463_{0.015}$	$0.539_{0.008}$	$\mathbf{0.563_{0.021}}$
scene	$0.563_{0.018}$	$0.587_{0.019}$	$0.655_{0.009}$	$0.699_{0.014}$	$0.617_{0.009}$	$0.713_{0.010}$	$\mathbf{0.735_{0.009}}$
yeast	$0.156_{0.012}$	$0.111_{0.007}$	$0.190_{0.009}$	$0.179_{0.006}$	$0.172_{0.014}$	$0.176_{0.007}$	$\mathbf{0.202_{0.009}}$
enron	$0.119_{0.033}$	$0.099_{0.008}$	$0.128_{0.013}$	$0.128_{0.006}$	$0.124_{0.014}$	$0.100_{0.009}$	$\mathbf{0.142_{0.010}}$
business	$0.454_{0.085}$	$0.359_{0.003}$	$0.565_{0.010}$	$0.557_{0.008}$	$0.538_{0.006}$	$0.542_{0.010}$	$\mathbf{0.569_{0.008}}$

D.2 Computational costs

The computational costs of the six algorithms on benchmark datasets are illustrated in Fig. 1. Note that the CPU time is plotted in the log scale in Figure 1. Besides, BR-SVM is implemented that based on the efficient LibSVM library [3] and thus is more efficient than the others implemented by the original Matlab code. For the last datasets, we use the linear kernel for CPNL, where it has more efficient implementation that is tailored for the linear model. In comparison, for the first six datasets, Rank-SVM and CPNL involves the kernel matrix in training. In this case, we can observe that when the number of the dataset becomes relatively larger (i.e., Image, scene, and yeast), our method MLC-NTK (with Nyström method) is more efficient than these two. Note that in experiments we set $m = r = n$ in the Nyström method for high performance although it can reduce r (and m) to accelerate the efficiency.

Table 2: Experimental results of benchmark approaches ($\mathrm{mean}_{\mathrm{std}}$) w.r.t. instance-F1 (↑) on all datasets. Best results are highlighted in bold.

Dataset	Rank-SVM	BP-MLL	BR-SVM	CPNL	MLFE	MLC-RBF	MLC-NTK
flags	$0.703_{0.017}$	$\mathbf{0.717_{0.004}}$	$0.700_{0.017}$	$0.697_{0.031}$	$0.688_{0.011}$	$0.703_{0.017}$	$0.711_{0.018}$
birds	$0.338_{0.023}$	$0.425_{0.014}$	$0.469_{0.017}$	$\mathbf{0.476_{0.031}}$	$0.400_{0.029}$	$0.417_{0.030}$	$0.468_{0.024}$
emotions	$0.645_{0.020}$	$0.655_{0.034}$	$0.620_{0.020}$	$\mathbf{0.684_{0.019}}$	$0.621_{0.021}$	$0.656_{0.022}$	$0.656_{0.026}$
image	$0.631_{0.016}$	$0.523_{0.026}$	$0.623_{0.014}$	$0.698_{0.012}$	$0.593_{0.014}$	$0.682_{0.010}$	$\mathbf{0.700_{0.013}}$
scene	$0.664_{0.015}$	$0.730_{0.011}$	$0.717_{0.010}$	$\mathbf{0.802_{0.009}}$	$0.685_{0.010}$	$0.783_{0.006}$	$0.795_{0.009}$
yeast	$\mathbf{0.632_{0.007}}$	$0.614_{0.015}$	$0.623_{0.006}$	$0.630_{0.007}$	$0.607_{0.011}$	$0.614_{0.007}$	$0.629_{0.005}$
enron	$0.563_{0.026}$	$0.566_{0.006}$	$0.529_{0.010}$	$\mathbf{0.585_{0.007}}$	$0.538_{0.012}$	$0.491_{0.007}$	$0.566_{0.008}$
business	$0.734_{0.036}$	$0.722_{0.001}$	$0.763_{0.005}$	$\mathbf{0.770_{0.006}}$	$0.765_{0.005}$	$0.739_{0.006}$	$\mathbf{0.770_{0.005}}$

Table 3: Experimental results of benchmark approaches ($\mathrm{mean}_{\mathrm{std}}$) w.r.t. Coverage (↓) on all datasets. Best results are highlighted in bold.

Dataset	Rank-SVM	BP-MLL	BR-SVM	CPNL	MLFE	MLC-RBF	MLC-NTK
flags	$0.525_{0.035}$	$0.551_{0.010}$	$0.548_{0.019}$	$0.558_{0.021}$	$0.561_{0.019}$	$0.531_{0.021}$	$\mathbf{0.525_{0.022}}$
birds	$0.256_{0.014}$	$0.285_{0.008}$	$0.262_{0.011}$	$0.254_{0.019}$	$0.276_{0.024}$	$0.258_{0.011}$	$\mathbf{0.235_{0.021}}$
emotions	$0.294_{0.011}$	$0.310_{0.014}$	$0.386_{0.017}$	$0.277_{0.010}$	$0.282_{0.013}$	$0.279_{0.008}$	$\mathbf{0.275_{0.010}}$
image	$0.171_{0.008}$	$0.210_{0.005}$	$0.227_{0.008}$	$\mathbf{0.157_{0.006}}$	$0.165_{0.006}$	$0.161_{0.004}$	$\mathbf{0.157_{0.007}}$
scene	$0.068_{0.004}$	$0.085_{0.001}$	$0.119_{0.004}$	$0.064_{0.002}$	$0.067_{0.003}$	$0.065_{0.002}$	$\mathbf{0.061_{0.004}}$
yeast	$0.446_{0.006}$	$0.480_{0.010}$	$0.627_{0.007}$	$0.445_{0.006}$	$0.452_{0.006}$	$0.441_{0.006}$	$\mathbf{0.433_{0.005}}$
enron	$0.235_{0.021}$	$0.242_{0.001}$	$0.580_{0.012}$	$0.232_{0.006}$	$0.228_{0.010}$	$0.245_{0.005}$	$\mathbf{0.207_{0.009}}$
business	$0.068_{0.005}$	$0.092_{0.005}$	$0.338_{0.009}$	$0.065_{0.002}$	$0.082_{0.004}$	$0.069_{0.003}$	$\mathbf{0.063_{0.002}}$

Table 4: Experimental results of benchmark approaches ($\mathrm{mean}_{\mathrm{std}}$) w.r.t. Average Precision (↑) on all datasets. Best results are highlighted in bold.

Dataset	Rank-SVM	BP-MLL	BR-SVM	CPNL	MLFE	MLC-RBF	MLC-NTK
flags	$0.821_{0.014}$	$0.816_{0.002}$	$0.808_{0.016}$	$0.803_{0.022}$	$0.803_{0.012}$	$0.824_{0.019}$	$\mathbf{0.824_{0.015}}$
birds	$0.635_{0.015}$	$0.591_{0.015}$	$0.637_{0.016}$	$0.650_{0.020}$	$0.634_{0.028}$	$0.619_{0.016}$	$\mathbf{0.656_{0.023}}$
emotions	$0.808_{0.010}$	$0.786_{0.018}$	$0.760_{0.015}$	$0.828_{0.010}$	$0.822_{0.012}$	$0.827_{0.010}$	$\mathbf{0.830_{0.018}}$
image	$0.823_{0.010}$	$0.762_{0.013}$	$0.772_{0.011}$	$0.839_{0.007}$	$0.826_{0.008}$	$0.832_{0.007}$	$\mathbf{0.841_{0.007}}$
scene	$0.882_{0.008}$	$0.857_{0.005}$	$0.834_{0.006}$	$\mathbf{0.893_{0.006}}$	$0.885_{0.005}$	$0.885_{0.004}$	$\mathbf{0.893_{0.006}}$
yeast	$0.755_{0.005}$	$0.732_{0.022}$	$0.680_{0.007}$	$0.775_{0.009}$	$0.769_{0.008}$	$0.770_{0.008}$	$\mathbf{0.780_{0.005}}$
enron	$0.672_{0.025}$	$0.674_{0.005}$	$0.482_{0.010}$	$0.702_{0.010}$	$0.705_{0.008}$	$0.663_{0.005}$	$\mathbf{0.714_{0.006}}$
business	$0.860_{0.036}$	$0.865_{0.002}$	$0.742_{0.005}$	$0.891_{0.005}$	$0.885_{0.004}$	$0.878_{0.004}$	$\mathbf{0.893_{0.004}}$

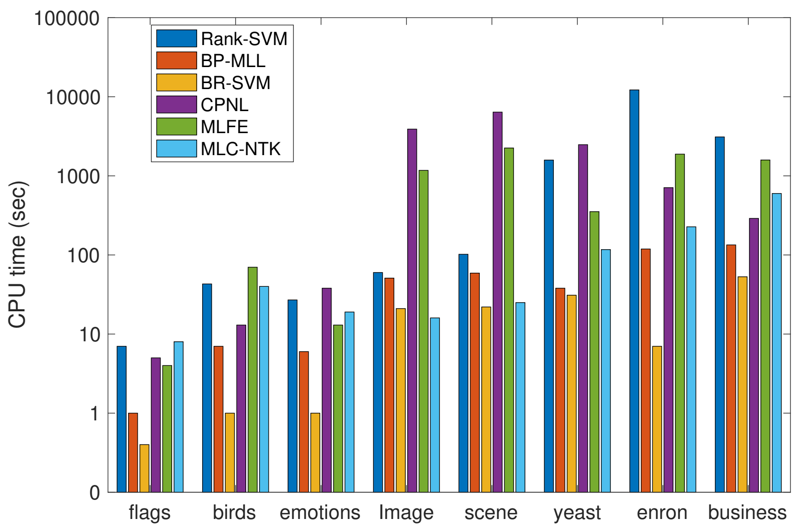

Figure 1: Computational costs of the six algorithms on benchmark datasets.

D.3 Sensitivity analysis w.r.t. the hyper-parameters τ and λ

In this section, we give the sensitivity analysis w.r.t. the hyper-parameters τ and λ, which are illustrated in Fig. 2 and 3 respectively. We can observe that the hyper-parameter τ is more sensitive to λ because we argue that τ has more power to control the model complexity than λ.

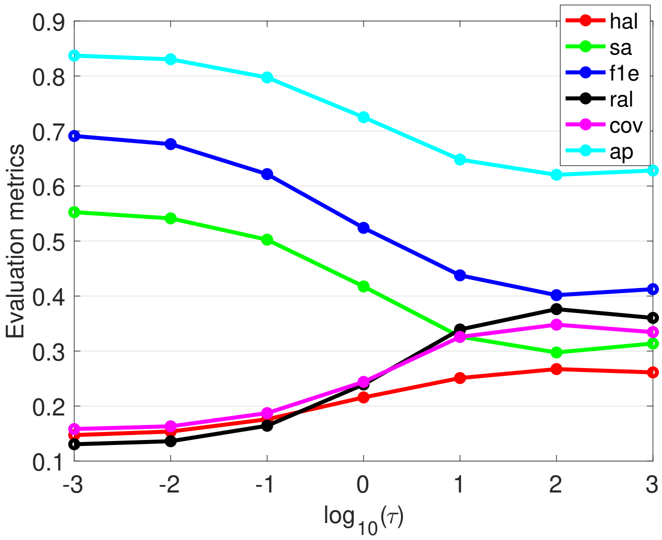

Figure 2: Sensitivity analysis w.r.t. the hyper-parameter τ. Hal is the abbreviation of Hamming Loss, and so on.

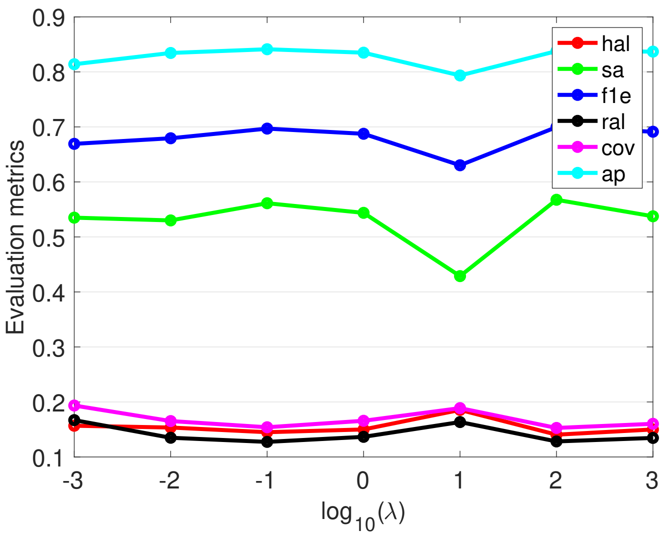

Figure 3: Sensitivity analysis w.r.t. the hyper-parameter λ. Hal is the abbreviation of Hamming Loss, and so on.

References

[1] Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

[2] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[3] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[4] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[5] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016.

[6] Guoqiang Wu and Jun Zhu. Multi-label classification: do hamming loss and subset accuracy really conflict with each other? *Advances in Neural Information Processing Systems*, 33, 2020.