Transfer Learning via Learning to Transfer: Supplementary Material

Ying Wei [1 2] **Yu Zhang** [1] **Junzhou Huang** [2] **Qiang Yang** [1]

1. Empirical Estimation of $\mathbf{Q}_e$

$$\begin{aligned}\hat{\mathbf{Q}}_e =& \frac{1}{n^2-1}\sum_{i,i'=1}^{n}\sum_{k=1}^{N_k}\Big[\mathcal{K}_k(\mathbf{x}_i^s\mathbf{W},\mathbf{x}_{i'}^s\mathbf{W})+\mathcal{K}_k(\mathbf{x}_i^t\mathbf{W},\mathbf{x}_{i'}^t\mathbf{W})-2\mathcal{K}_k(\mathbf{x}_i^s\mathbf{W},\mathbf{x}_{i'}^t\mathbf{W})\\ &-\frac{1}{n^2}\sum_{i,i'=1}^{n}\Big(\mathcal{K}_k(\mathbf{x}_i^s\mathbf{W},\mathbf{x}_{i'}^s\mathbf{W})+\mathcal{K}_k(\mathbf{x}_i^t\mathbf{W},\mathbf{x}_{i'}^t\mathbf{W})-2\mathcal{K}_k(\mathbf{x}_i^s\mathbf{W},\mathbf{x}_{i'}^t\mathbf{W})\Big)\Big]^2,\end{aligned} \tag{1}$$

where $n=\min(n_e^s,n_e^t)$.

2. The Gradient towards Optimizing W

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}}=\frac{\partial(\boldsymbol{\beta}^*)^T\hat{\mathbf{d}}_{\mathbf{W}}}{\partial\mathbf{W}}+\lambda^*\frac{\partial(\boldsymbol{\beta}^*)^T\hat{\mathbf{Q}}_{\mathbf{W}}\boldsymbol{\beta}^*}{\partial\mathbf{W}}-\mu^*\frac{1}{[(\boldsymbol{\beta}^*)^T\boldsymbol{\tau}_{\mathbf{W}}]^2}\frac{\partial(\boldsymbol{\beta}^*)^T\boldsymbol{\tau}_{\mathbf{W}}}{\partial\mathbf{W}}+2\gamma_2\mathbf{W}, \tag{2}$$

among which

$$\frac{\partial(\boldsymbol{\beta}^*)^T\hat{\mathbf{d}}_{\mathbf{W}}}{\partial\mathbf{W}}=\sum_{k=1}^{N_k}\beta_k^*\Big[\frac{1}{(n^s)^2}\sum_{i,i'=1}^{n_s}\hat{\mathbf{K}}_{k(i,i')}^{ss}+\frac{1}{(n^t)^2}\sum_{j,j'=1}^{n_t}\hat{\mathbf{K}}_{k(j,j')}^{tt}-\frac{2}{n^sn^t}\sum_{i,j=1}^{n_s,n_t}\hat{\mathbf{K}}_{k(i,j)}^{st})\Big], \tag{3}$$

$$\begin{aligned}\frac{\partial(\boldsymbol{\beta}^*)^T\hat{\mathbf{Q}}_{\mathbf{W}}\boldsymbol{\beta}^*}{\partial\mathbf{W}}=&\frac{1}{n^2-1}\sum_{k=1}^{N_k}\sum_{i,i'=1}^{n}\hat{\mathbf{B}}_{k(i,i')}\Big[\hat{\mathbf{K}}_{k(i,i')}^{ss}+\hat{\mathbf{K}}_{k(i,i')}^{tt}-2\hat{\mathbf{K}}_{k(i,i')}^{st})\\ &-\sum_{i,i'=1}^{n}\Big(\hat{\mathbf{K}}_{k(i,i')}^{ss}+\hat{\mathbf{K}}_{k(i,i')}^{tt}-2\hat{\mathbf{K}}_{k(i,i')}^{st})\Big)\Big]\end{aligned} \tag{4}$$

$$\frac{\partial(\boldsymbol{\beta}^*)^T\boldsymbol{\tau}_{\mathbf{W}}}{\partial\mathbf{W}}=\sum_{k=1}^{N_k}\beta_k\frac{2[\mathrm{tr}(\mathbf{W}^T\mathbf{S}_k^L\mathbf{W})]\mathbf{S}_k^N-2[\mathrm{tr}(\mathbf{W}^T\mathbf{S}_k^N\mathbf{W})]\mathbf{S}_k^L}{[\mathrm{tr}(\mathbf{W}^T\mathbf{S}_k^L\mathbf{W})]^2}, \tag{5}$$

where $\hat{\mathbf{K}}_{k(i,i')}^{ss}$, $\hat{\mathbf{K}}_{k(j,j')}^{tt}$, and $\hat{\mathbf{K}}_{k(i,j)}^{st}$ depending on the kernel function are calculated as follows,

$$\hat{\mathbf{K}}_{k(i,i')}^{ss}=-\frac{2}{\delta_k}\mathcal{K}_k(\mathbf{x}_i^s\mathbf{W},\mathbf{x}_{i'}^s\mathbf{W})(\mathbf{x}_i^s-\mathbf{x}_{i'}^s)(\mathbf{x}_i^s-\mathbf{x}_{i'}^s)^T\mathbf{W}, \tag{6}$$

[1]Hong Kong University of Science and Technology, Hong Kong [2]Tencent AI Lab, Shenzhen, China. Correspondence to: Ying Wei <judyweiying@gmail.com>, Qiang Yang <qyang@cse.ust.hk>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

$$\hat{\mathbf{K}}^{tt}_{k(i,i')} = -\frac{2}{\delta_k}\mathcal{K}_k(\mathbf{x}^t_i\mathbf{W}, \mathbf{x}^t_{i'}\mathbf{W})(\mathbf{x}^t_i - \mathbf{x}^t_{i'})(\mathbf{x}^s_i - \mathbf{x}^s_{i'})^T\mathbf{W}, \tag{7}$$

$$\hat{\mathbf{K}}^{st}_{k(i,i')} = -\frac{2}{\delta_k}\mathcal{K}_k(\mathbf{x}^s_i\mathbf{W}, \mathbf{x}^s_{i'}\mathbf{W})(\mathbf{x}^s_i - \mathbf{x}^t_{i'})(\mathbf{x}^s_i - \mathbf{x}^t_{i'})^T\mathbf{W}. \tag{8}$$

In Eqaution 4,

$$\hat{\mathbf{B}}_{k(i,i')} = \sum_{k=1}^{N_k}\sum_{i,i'=1}^{n} 2\beta^*_k \Big[\mathcal{K}_k(\mathbf{x}^s_i\mathbf{W}, \mathbf{x}^s_{i'}\mathbf{W}) + \mathcal{K}_k(\mathbf{x}^t_i\mathbf{W}, \mathbf{x}^t_{i'}\mathbf{W}) - 2\mathcal{K}_k(\mathbf{x}^s_i\mathbf{W}, \mathbf{x}^t_{i'}\mathbf{W}) - \frac{1}{n^2}\sum_{i,i'=1}^{n}(\mathcal{K}_k(\mathbf{x}^s_i\mathbf{W}, \mathbf{x}^s_{i'}\mathbf{W}) + \mathcal{K}_k(\mathbf{x}^t_i\mathbf{W}, \mathbf{x}^t_{i'}\mathbf{W}) - 2\mathcal{K}_k(\mathbf{x}^s_i\mathbf{W}, \mathbf{x}^t_{i'}\mathbf{W}))\Big]. \tag{9}$$

3. Exemplar of A Pair of Source and Target Domains

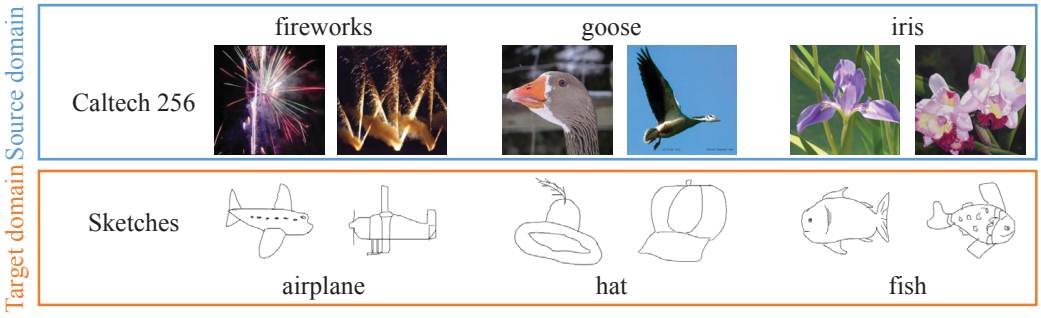


Figure 1. One example pair of source and target domains.

4. Coefficients of RBF Kernels

We plot the values of the coefficients for N_k RBF kernels, i.e., β_k for $k = \{1, \cdots, N_k\}$ in Figure 2. Note that we use 33 RBF kernels as stated in the paper.

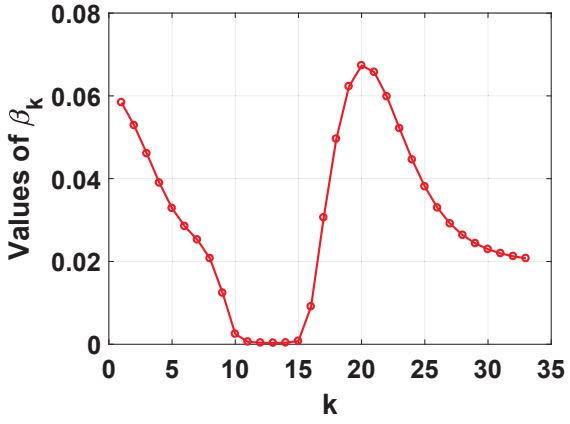
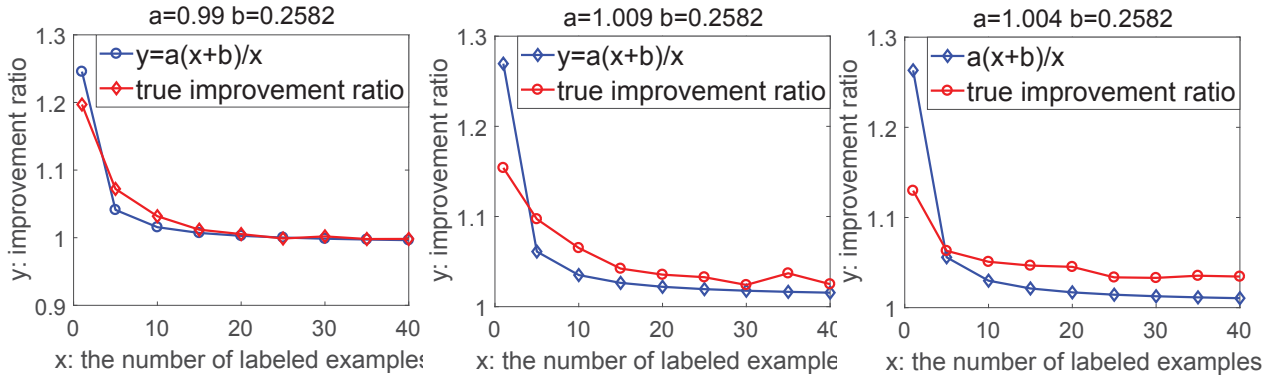


Figure 2. Values of the coefficients for all N_k RBF kernels.

5. Discussion on l_e in Equation (2)

l_e, the performance improvement ratio, heavily depends on the number of labeled examples in the target domain $\mathcal{T}_e$, i.e., n^t_{le}. A smaller number of target labeled examples tends to produce a larger performance improvement ratio, and vice versa.



(a) 1^{st} pair (b) 2^{nd} pair (c) 3^{rd} pair

Figure 3. Empirical validation of the assumed function that bridges the number of labeled examples in a target domain and the performance improvement ratio.

Since the n_{le}^t varies from experience to experience, we adopt a transformed $\hat{l}_e$ instead of l_e to train the reflection function in Equation (2). The $\hat{l}_e$ is expected to be the expectation of the performance improvement ratio in the range of $[p, q]$ for the e-th experience, where p and q are the minimum and maximum number of target labeled examples. To compute the expectation, we first assume that the performance improvement ratio with regard to the number of target labeled examples follows the following monotonically decreasing function,

$$f(x) = \frac{a_e(x+b)}{x}, \tag{10}$$

where a_e and b are two parameters deciding the function. a_e is conditioned on a specific experience but b is shared across all experiences. With the assumed function, we can obtain the expected performance improvement ratio as:

$$\hat{l}_e = \frac{1}{q-p}\int_p^q \frac{a_e(x+b)}{x}dx = a_e(1 + \frac{b}{q-p}\log\frac{q}{p}) \tag{11}$$

Combining with the fact that $f(n_{le}^t) = \frac{a_e(b+n_{le}^t}{n_{le}^t} = l_e$, we can finally obtain the corrected $\hat{l}_e$ as,

$$\hat{l}_e = l_e\frac{n_{le}^t}{n_{le}^t + b}(1 + \frac{b}{q-p}\log\frac{q}{p}),$$

where the parameter b can be learned simultaneously during optimizing the objective (2).

We have empirically validated the soundness of the assumed function in Figure 3. We first randomly selected three transfer learning experiences. For each, we vary the number of labeled examples in the target domain and obtain the curve of the performance improvement ratio w.r.t. the number of labeled examples (see the red lines in Figure 3). Meanwhile, we fix $b = 0.2582$ which is learned from all transfer learning experiences, and fit the function $y = \frac{a_e(x+b)}{x}$ to the true trend by solving a_e. Obviously, the function is qualified to model the trend of the performance improvement ratio w.r.t. the number of labeled examples.

6. Algorithmic Stability and Generalization Bounds

6.1. Latent Feature Factor Based Algorithms

In this subsection, we would discuss the algorithmic stability and generalization bounds for latent feature factor based transfer learning algorithms without considering past transfer leaning experiences. We denote the transferred knowledge from the e-th source domain to the e-th target domain as the latent feature factor matrix $\mathbf{W}_e$. As a result of the knowledge transfer, the target domain is represented on the latent feature factors, i.e., $\mathbf{X}_e^t\mathbf{W}_e$. Without loss of generality, we assume

that the e-th target domain learns a predictor h'_e by minimizing the following regularized least square objective function:

$$\min_{h'_e} \frac{1}{n^t_{le}} \sum_{j=1}^{n^t_{le}} \|y^t_{ej} - \langle h'_e, \mathbf{W}_e^T (\mathbf{x}^t_{ej})^T \rangle\|^2 + \lambda \|h'_e\|^2, \tag{12}$$

where λ is a regularization parameter. Setting $h_e = \mathbf{W}_e h'_e$ and meanwhile constraining $\mathbf{W}_e$ to be orthonormal (i.e., $\mathbf{W}_e^T \mathbf{W}_e = \mathbf{I}$), we obtain an equivalent optimization problem as below:

$$\min_{h_e} \frac{1}{n^t_{le}} \sum_{j=1}^{n^t_{le}} \|y^t_{ej} - \langle h_e, (\mathbf{x}^t_{ej})^T \rangle\|^2 + \lambda \|\mathbf{W}_e^T h_e\|^2. \tag{13}$$

Note that such orthonormal constraint on the latent feature factor matrix $\mathbf{W}_e$ has been widely adopted and proven effective in dimension reduction methods such as PCA as well as in transfer learning algorithms (Baktashmotlagh et al., 2013; 2014). The optimization problem (13), also known as Tikhonov regularization (Golub et al., 1999), is uniformly stable.

Theorem 1. *Suppose that for any $\mathbf{x}^t_e$ we have $\|\boldsymbol{\alpha}_e\| = \|\mathbf{x}^t_e \mathbf{W}_e\| \le r$, and for any y^t_e we have $|y^t_e| \le B$. The algorithm that minimizes problem (13) is uniformly stable. For any $(\mathbf{x}^t_e, y^t_e)$ in the target domain, the following inequality holds:*

$$\left| \|y^t_e - \langle h_e(\mathcal{T}_e), (\mathbf{x}^t_e)^T \rangle\|^2 - \|y^t_e - \langle h_e(\mathcal{T}_{ek}), (\mathbf{x}^t_e)^T \rangle\|^2 \right| \le \frac{4B^2 r^2}{\lambda n^t_{le}}, \tag{14}$$

where $\mathcal{T}_e = \{(\mathbf{x}^t_{e1}, y^t_{e1}), \cdots, (\mathbf{x}^t_{ek-1}, y^t_{ek-1}), (\mathbf{x}^t_{ek}, y^t_{ek}), (\mathbf{x}^t_{ek+1}, y^t_{ek+1}), \cdots, (\mathbf{x}^t_{en^t_{le}}, y^t_{en^t_{le}})\}$ denotes the full set of training samples, and $\mathcal{T}_{ek} = \{(\mathbf{x}^t_{e1}, y^t_{e1}), \cdots, (\mathbf{x}^t_{ek-1}, y^t_{ek-1}), (\tilde{\mathbf{x}}^t_{ek}, \tilde{y}^t_{ek}), (\mathbf{x}^t_{ek+1}, y^t_{ek+1}), \cdots, (\mathbf{x}^t_{en^t_{le}}, y^t_{en^t_{le}})\}$ represents the training samples with the k-th example replaced to $(\tilde{\mathbf{x}}^t_{ek}, \tilde{y}^t_{ek})$. $h_e(\mathcal{T}_e)$ is the hypothesis learned using training samples from $\mathcal{T}_e$.

Proof. Our proof generalizes the proof in (Liu et al., 2017) in which the authors proved the uniform stability of self-taught learning algorithms. Let $f_1(h_e) = \lambda \|\mathbf{W}_e^T h_e\|^2$, $f_{\mathcal{T}_e}(h_e) = \frac{1}{n^t_{le}} \sum_{j=1}^{n^t_{le}} \|y^t_{ej} - \langle h_e, (\mathbf{x}^t_{ej})^T \rangle\|^2 + f_1(h_e)$, and $f_{\mathcal{T}_{ek}}(h_e) = \frac{1}{n^t_{le}} \big(\sum_{j \neq k}^{n^t_{le}} \|y^t_{ej} - \langle h_e, (\mathbf{x}^t_{ej})^T \rangle\|^2 + \|\tilde{y}^t_{ek} - \langle h_e, (\tilde{\mathbf{x}}^t_{ek})^T \rangle\|^2 \big) + f_1(h_e)$. Both f_1 and f_2 are convex functions. Following (Liu et al., 2017), we take advantage of the additive and non-negative properties of Bregman divergence, i.e.,

$$B_{f_{\mathcal{T}_e}}(\tilde{h}_e \parallel h_e) + B_{f_{\mathcal{T}_{ek}}}(h_e \parallel \tilde{h}_e) \ge B_{f_1}(\tilde{h}_e \parallel h_e) + B_{f_1}(h_e \parallel \tilde{h}_e), \tag{15}$$

where h_e is the solution to the optimization problem (13) when training samples are from $\mathcal{T}_e$, and instead $\tilde{h}_e$ is the solution with training samples in the set of $\mathcal{T}_{ek}$. According to the definition of Bregman divergence, the left-hand side of the inequality

$$\begin{aligned}
&B_{f_{\mathcal{T}_e}}(\tilde{h}_e \parallel h_e) + B_{f_{\mathcal{T}_{ek}}}(h_e \parallel \tilde{h}_e) \\
=&f_{\mathcal{T}_e}(\tilde{h}_e) - f_{\mathcal{T}_e}(h_e) - \langle \tilde{h}_e - h_e, \nabla f_{\mathcal{T}_e}(h_e) \rangle + f_{\mathcal{T}_{ek}}(h_e) - f_{\mathcal{T}_{ek}}(\tilde{h}_e) - \langle h_e - \tilde{h}_e, \nabla f_{\mathcal{T}_{ek}}(\tilde{h}_e) \rangle \\
=&f_{\mathcal{T}_e}(\tilde{h}_e) - f_{\mathcal{T}_e}(h_e) + f_{\mathcal{T}_{ek}}(h_e) - f_{\mathcal{T}_{ek}}(\tilde{h}_e) \\
=&\frac{1}{n^t_{le}} \big(\|y^t_{ek} - \langle \tilde{h}_e, (\mathbf{x}^t_{ek})^T \rangle\|^2 - \|y^t_{ek} - \langle h_e, (\mathbf{x}^t_{ek})^T \rangle\|^2 + \|\tilde{y}^t_{ek} - \langle h_e, (\tilde{\mathbf{x}}^t_{ek})^T \rangle\|^2 - \|\tilde{y}^t_{ek} - \langle \tilde{h}_e, (\tilde{\mathbf{x}}^t_{ek})^T \rangle\|^2 \big) \\
\le&\frac{2B}{n^t_{le}} \big(|\langle h_e - \tilde{h}_e, (\mathbf{x}^t_{ek})^T \rangle| + |\langle h_e - \tilde{h}_e, (\tilde{\mathbf{x}}^t_{ek})^T \rangle| \big) \\
=&\frac{2B}{n^t_{le}} \big(|\langle h_e - \tilde{h}_e, \mathbf{W}_e (\boldsymbol{\alpha}^t_{ek})^T \rangle| + |\langle h_e - \tilde{h}_e, \mathbf{W}_e (\tilde{\boldsymbol{\alpha}}^t_{ek})^T \rangle| \big) \\
=&\frac{2B}{n^t_{le}} \big(|\langle \mathbf{W}_e^T (h_e - \tilde{h}_e), (\boldsymbol{\alpha}^t_{ek})^T \rangle| + |\langle \mathbf{W}_e^T (h_e - \tilde{h}_e), (\tilde{\boldsymbol{\alpha}}^t_{ek})^T \rangle| \big) \\
\le&\frac{4Br}{n^t_{le}} \sqrt{\sum_u \langle h_e - \tilde{h}_e, \mathbf{W}_{eu} \rangle^2},
\end{aligned} \tag{16}$$

where the first inequality comes from the fact that the squared loss function is $2B$-admissible given $|y_e^t| \leq B$, and the second follows the famous Cauchy-Schwarz inequality. Meanwhile, the right-hand side of the inequality (15) $B_{f_1}(\tilde{h}_e \parallel h_e) + B_{f_1}(h_e \parallel \tilde{h}_e)$ equals,

$$B_{f_1}(\tilde{h}_e \parallel h_e) + B_{f_1}(h_e \parallel \tilde{h}_e) = 2\lambda \sum_u \langle h_e - \tilde{h}_e, \mathbf{W}_{eu} \rangle^2. \tag{17}$$

Provided in (15) that the right-hand side is upper-bounded by the left-hand side, we combine (16) and (17) and give,

$$\sqrt{\sum_u \langle h_e - \tilde{h}_e, \mathbf{W}_{eu} \rangle^2} \leq \frac{2Br}{\lambda n_{le}^t}. \tag{18}$$

Finally, we reach the conclusion that,

$$\begin{aligned} &\left| \|y_e^t - \langle h_e, (\mathbf{x}_e^t)^T \rangle\|^2 - \|y_e^t - \langle \tilde{h}_e, (\mathbf{x}_e^t)^T \rangle\|^2 \right| \\ \leq & 2B |\langle h_e - \tilde{h}_e, (\mathbf{x}_e^t)^T \rangle| \\ \leq & 2B \sqrt{\sum_u \langle h_e - \tilde{h}_e, \mathbf{W}_{eu} \rangle^2} \|\boldsymbol{\alpha}_e^t\| \\ \leq & \frac{4B^2 r^2}{\lambda n_{le}^t} \end{aligned} \tag{19}$$

□

Upon Theorem 1, the generalization bound for the e-th target domain with knowledge transferred from the e-th source domain can be derived in the following theorem according to (Bousquet & Elisseeff, 2002).

Theorem 2. *Suppose that the squared loss function is upper-bounded by M. For any h_e that is the solution of the optimization problem 13, and any $\delta > 0$, with probability at least $1 - \delta$,*

$$R^t(h_e) \leq R_{n_{le}^t}^t(h_e) + \frac{4B^2 r^2}{\lambda n_{le}^t} + \left(\frac{16B^2 r^2}{\lambda} + M \right) \sqrt{\frac{\log 1/\delta}{2n_{le}^t}}, \tag{20}$$

where $R^t(h_e) = E_{(\mathbf{x},y)} \|y - \langle h_e, \mathbf{x}^T \rangle\|^2$ is the expected risk and $R_{n_{le}^t}^t(h_e) = \frac{1}{n_{le}^t} \sum_{j=1}^{n_{le}^t} \|y_{ej}^t - \langle h_e, (\mathbf{x}_{ej}^t)^T \rangle\|^2$ denotes the empirical risk we have mentioned above.

6.2. Learning to Transfer

In the next, we conduct a theoretic investigation into how previous transfer learning experiences influence a transfer learning task of interest. Consider $\mathbf{S} = \{\langle \mathcal{S}_1, \mathcal{T}_1 \rangle, \cdots, \langle \mathcal{S}_{N_e}, \mathcal{T}_{N_e} \rangle\}$ to be N_e transfer learning experiences or the so-called meta-samples (Maurer, 2005). Meanwhile, we focus on the $(N_e + 1)$-th transfer learning task $\langle \mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1} \rangle$ in which the $(N_e + 1)$-th source domain is expected to improve learning performance of the $(N_e + 1)$-th target domain via knowledge transfer. Let $\mathbf{L}(\mathbf{S})$ be our algorithm that learns meta-cognitive knowledge from N_e transfer learning experiences in $\mathbf{S}$ and applies the knowledge to the $(N_e + 1)$-th transfer learning task. Before proceeding to give the generalization bound, we first prove that $\mathbf{L}(\mathbf{S})$ is uniformly stable.

To make the proof self-contained, we first present Lemma 11 in (Maurer, 2005) as following.

Lemma 1. *(Maurer, 2005) Let $\mathbf{G}_1$ and $\mathbf{G}_2$ be positive operators and $\lambda > 0$. Then*

1. *$\mathbf{G}_i + \lambda \mathbf{I}$ is invertible;*
2. *$\|(\mathbf{G}_i + \lambda \mathbf{I})^{-1}\|_\infty \leq \frac{1}{\lambda}$;*
3. *we have $\|(\mathbf{G}_1 + \lambda \mathbf{I})^{-1} - (\mathbf{G}_2 + \lambda \mathbf{I})^{-1}\|_\infty \leq \frac{1}{\lambda^2} \|\mathbf{G}_1 - \mathbf{G}_2\|_\infty$;*

4. let $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *satisfy* $(\mathbf{G}_i + m\lambda\mathbf{I})\mathbf{x}_i = y$*, then* $\big|\|\mathbf{x}_1\|^2 - \|\mathbf{x}_2\|^2\big| \leq 2(m\lambda)^{-3}\|\mathbf{G}_1 - \mathbf{G}_2\|_\infty\|y\|^2$.

Theorem 3. *Suppose that for any* $\mathbf{x}_e^t$ *we have* $\|\mathbf{x}_e^t\|^2 \leq r_x$*, and for any* y_e^t *we have* $|y_e^t| \leq B$*. Meanwhile, for any* e*-th transfer learning experience, we assume that the latent feature factor matrix* $\|\mathbf{W}_e\| \leq r_W$*. We also reasonably assume that the latent feature factor matrix for the* (N_e+1)*-th transfer learning task is a linear combination of all* N_e *historical latent factor feature matrices plus a noisy latent feature matrix* $\mathbf{W}_\epsilon$ *satisfying* $\|\mathbf{W}_\epsilon\| \leq r_\epsilon$*, i.e.,* $\mathbf{W}_{N_e+1} = \sum_{e=1}^{N_e} c_e\mathbf{W}_e + \mathbf{W}_\epsilon$ *with each coefficient* $0 \leq c_e \leq 1$*. Our algorithm* $\mathbf{L}(\mathbf{S})$ *is uniformly stable. For any* $\langle\mathcal{S},\mathcal{T}\rangle$ *as the coming transfer learning task, the following inequality holds:*

$$\Big|l_{emp}(\mathbf{L}(\mathbf{S}),(\mathcal{S},\mathcal{T})) - l_{emp}(\mathbf{L}(\mathbf{S}^{e_0}),(\mathcal{S},\mathcal{T}))\Big| \leq \frac{4(4N_e - 3 + r_\epsilon/r_W)B^2r_x}{\lambda N_e^2} \sim \mathcal{O}\bigg(\frac{B^2r_x}{\lambda N_e}\bigg), \tag{21}$$

where $\mathbf{S} = \{\langle\mathcal{S}_1,\mathcal{T}_1\rangle,\cdots,\langle\mathcal{S}_{e_0-1},\mathcal{T}_{e_0-1}\rangle,\langle\mathcal{S}_{e_0},\mathcal{T}_{e_0}\rangle,\langle\mathcal{S}_{e_0+1},\mathcal{T}_{e_0+1}\rangle,\cdots,\langle\mathcal{S}_{N_e},\mathcal{T}_{N_e}\rangle\}$ *denotes the full set of meta-samples, and* $\mathbf{S}^{e_0} = \{\langle\mathcal{S}_1,\mathcal{T}_1\rangle,\cdots,\langle\mathcal{S}_{e_0-1},\mathcal{T}_{e_0-1}\rangle,\langle\mathcal{S}_{e_0'},\mathcal{T}_{e_0'}\rangle,\langle\mathcal{S}_{e_0+1},\mathcal{T}_{e_0+1}\rangle,\cdots,\langle\mathcal{S}_{N_e},\mathcal{T}_{N_e}\rangle\}$ *represents the meta-samples with the* e_0*-th meta-example replaced as* $\langle\mathcal{S}_{e_0'},\mathcal{T}_{e_0'}\rangle$*.*

Proof. Out proof generalizes the proof in (Maurer, 2005) where the authors proved the uniform stability of a meta-algorithm for regularized least squares regression. Before proceeding to prove the stability of our algorithm $\mathbf{L}(\mathbf{S})$, we first present the solution to the optimization problem (12) as well as the empirical loss in a different perspective. Following (Maurer, 2005), we make a reasonable assumption that the solution to the optimization problem (12) is generated by all training samples $\{\mathbf{x}_1,\cdots,\mathbf{x}_{n_{le}^t}\}$, i.e., $h_e' = \sum_{j=1}^{n_{le}^t}\gamma_j\mathbf{x}_{ej}^t\mathbf{W}_e$. As a result, the problem (12) in a matrix form is

$$\min_{\boldsymbol{\gamma}} \frac{1}{n_{le}^t}\|\mathbf{y}_e^t - \mathbf{X}_e^t\mathbf{W}_e\mathbf{W}_e^T(\mathbf{X}_e^t)^T\boldsymbol{\gamma}\|^2 + \lambda\|\mathbf{W}_e^T(\mathbf{X}_e^t)^T\boldsymbol{\gamma}\|^2, \tag{22}$$

where $\mathbf{y}_e^t \in \mathbb{R}^{n_{le}^t\times 1}$, $\boldsymbol{\gamma} = \{\gamma_1,\cdots,\gamma_{n_{le}^t}\} \in \mathbb{R}^{n_{le}^t\times 1}$, and $\mathbf{X}_e^t \in \mathbb{R}^{n_{le}^t\times m}$ as we stated in the paper. To solve $\boldsymbol{\gamma}$, we set the gradient of (22) w.r.t. $\boldsymbol{\gamma}$ to zero, which gives birth to the following equation:

$$(\mathbf{G} + n_{le}^t\lambda\mathbf{I}_{(n_{le}^t\times n_{le}^t)})\boldsymbol{\gamma} = \mathbf{y}_e^t, \tag{23}$$

where $\mathbf{G} \in \mathbb{R}^{n_{le}^t\times n_{le}^t}$ is the well-known Gram matrix with $G_{jj'} = \langle\mathbf{W}_e^T(\mathbf{x}_{ej}^t)^T, \mathbf{W}_e^T(\mathbf{x}_{ej'}^t)^T\rangle$. It follows from (Maurer, 2005) that the empirical loss

$$\frac{1}{n_{le}^t}\|\mathbf{y}_e^t - \mathbf{X}_e^t\mathbf{W}_e\mathbf{W}_e^T(\mathbf{X}_e^t)^T\boldsymbol{\gamma}\|^2 = n_{le}^t\lambda^2\sum_{j=1}^{n_{le}^t}\gamma_j^2. \tag{24}$$

Now let the meta-sample $\mathbf{S} = \{\langle\mathcal{S}_1,\mathcal{T}_1\rangle,\cdots,\langle\mathcal{S}_{N_e},\mathcal{T}_{N_e}\rangle\}$ be given. The sequence of latent factor feature matrices $(\mathbf{W}_1,\cdots,\mathbf{W}_{N_e})$ for the N_e transfer learning experiences capture different knowledge transferred between different pairs of domains. Here we simplify our algorithm $\mathbf{L}(\mathbf{S})$ to keep its key idea - harnessing the collective power of all past transfer learning experiences and determining the knowledge to be transferred via maximizing the improvement ratio. We do this by assuming that the latent feature factor matrix for the (N_e+1)-th transfer learning task of interest is the linear combination of all N_e historical latent factor feature matrices plus a noisy latent feature matrix $\mathbf{W}_\epsilon$ satisfying $\|\mathbf{W}_\epsilon\| \leq r_\epsilon$, i.e., $\mathbf{W}_{N_e+1} = \sum_{e=1}^{N_e} c_e\mathbf{W}_e + \mathbf{W}_\epsilon$. The coefficients $\{c_1,\cdots,c_{N_e}\}$, satisfying $0 \leq c_e \leq 1$, can be learned to maximize the improvement ratio. In this case, the N_e-th latent feature factor matrix can be either dependent or independent ($c_1 = \cdots = c_{N_e} = 0$) on previous transfer learning experiences, which sticks to the L2T. Following the similar idea in (Maurer, 2005), here we define a new inner product $\langle\cdot,\cdot\rangle_{\mathbf{S}}$ by

$$\langle(\mathbf{x}_j^t)^T,(\mathbf{x}_{j'}^t)^T\rangle_{\mathbf{S}} = \frac{1}{r_W^2N_e^2}\langle(\mathbf{x}_j^t)^T,\sum_{e=1}^{N_e}c_e\mathbf{W}_e + \mathbf{W}_\epsilon\rangle\langle\sum_{e'=1}^{N_e}c_{e'}\mathbf{W}_{e'} + \mathbf{W}_\epsilon,(\mathbf{x}_{j'}^t)^T\rangle, \tag{25}$$

so that we could formulate our algorithm $\mathbf{L}(\mathbf{S})$ as

$$\min_{h_{N_e+1}} \frac{1}{n_{N_e+1}^t}\sum_{j=1}^{n_{N_e+1}^t}\|y_{(N_e+1)j}^t - \mathbf{G}_{\mathbf{S}}\boldsymbol{\gamma}\|^2 + \|\sum_{e=1}^{N_e}c_e\mathbf{W}_e^T(\mathbf{X}_{N_e+1}^t)^T\boldsymbol{\gamma}\|^2, \tag{26}$$

where $\mathbf{G}_{\mathbf{S}} \in \mathbb{R}^{n^t_{N_e+1} \times n^t_{N_e+1}}$ is the new Gram matrix with $G_{\mathbf{S}jj'} = \langle (\mathbf{x}^t_{(N_e+1)j})^T, (\mathbf{x}^t_{(N_e+1)j'})^T \rangle_{\mathbf{S}}$.

By replacing the e_0-th transfer learning task in $\mathbf{S}$, we obtain another meta-sample $\mathbf{S}^{e_0}$. To infer the stability bound, we would upper bound the difference between the empirical loss using $\mathbf{S}$ and that using $\mathbf{S}^{e_0}$, i.e., $\big|l_{emp}(\mathbf{L}(\mathbf{S}), (\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1})) - l_{emp}(\mathbf{L}(\mathbf{S}^{e_0}), (\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1}))\big|$. Using the equation (24) gives

$$\big|l_{emp}(\mathbf{L}(\mathbf{S}), (\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1})) - l_{emp}(\mathbf{L}(\mathbf{S}^{e_0}), (\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1}))\big| = n^t_{N_e+1}\lambda^2 \big|\|\boldsymbol{\gamma}\|^2 - \|\boldsymbol{\gamma}^{e_0}\|^2\big|, \tag{27}$$

where $\boldsymbol{\gamma}$ is the solution to $(\mathbf{G}_{\mathbf{S}} + n^t_{N_e+1}\lambda \mathbf{I}_{(n^t_{N_e+1} \times n^t_{N_e+1})})\boldsymbol{\gamma} = \mathbf{y}^t_{N_e+1}$, while $\boldsymbol{\gamma}^{e_0}$ is the solution to $(\mathbf{G}_{\mathbf{S}^{e_0}} + n^t_{N_e+1}\lambda \mathbf{I}_{(n^t_{N_e+1} \times n^t_{N_e+1})})\boldsymbol{\gamma} = \mathbf{y}^t_{N_e+1}$. Meanwhile, we obtain

$$\begin{aligned}
&G_{\mathbf{S}jj'} - G_{\mathbf{S}^{e_0}jj'} \\
=&\frac{1}{r_W^2 N_e^2}\Bigg[\sum_{e\neq e_0}^{N_e}\Big(\langle (\mathbf{x}^t_{(N_e+1)j})^T, c_e\mathbf{W}_e\rangle\langle c_{e_0}\mathbf{W}_{e_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle - \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_e\mathbf{W}_e\rangle\langle c_{e'_0}\mathbf{W}_{e'_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle\Big) \\
&\quad + \sum_{e'\neq e_0}^{N_e}\Big(\langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e_0}\mathbf{W}_{e_0}\rangle\langle c_{e'}\mathbf{W}_{e'}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle - \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle c_{e'}\mathbf{W}_{e'}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle\Big) \\
&\quad + \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e_0}\mathbf{W}_{e_0}\rangle\langle c_{e_0}\mathbf{W}_{e_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle - \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle c_{e'_0}\mathbf{W}_{e'_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle \\
&\quad + \langle (\mathbf{x}^t_{(N_e+1)j})^T, \mathbf{W}_\epsilon\rangle\langle c_{e_0}\mathbf{W}_{e_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle - \langle (\mathbf{x}^t_{(N_e+1)j})^T, \mathbf{W}_\epsilon\rangle\langle c_{e'_0}\mathbf{W}_{e'_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle \\
&\quad + \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e_0}\mathbf{W}_{e_0}\rangle\langle \mathbf{W}_\epsilon, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle - \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle \mathbf{W}_\epsilon, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle\Bigg] \\
=&\frac{1}{r_W^2 N_e^2}\Bigg[\sum_{e=1}^{N_e}\langle (\mathbf{x}^t_{(N_e+1)j})^T, c_e\mathbf{W}_e\rangle\langle c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle \\
&\quad + \sum_{e'\neq e_0}^{N_e}\langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle c_{e'}\mathbf{W}_{e'}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle \\
&\quad + \langle (\mathbf{x}^t_{(N_e+1)j})^T, \mathbf{W}_\epsilon\rangle\langle c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle \\
&\quad + \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle \mathbf{W}_\epsilon, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle \\
&\quad + \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e_0}\mathbf{W}_{e_0}\rangle\langle c_{e_0}\mathbf{W}_{e_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle - \langle (\mathbf{x}^t_{(N_e+1)j})^T, c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle c_{e'_0}\mathbf{W}_{e'_0}, (\mathbf{x}^t_{(N_e+1)j'})^T\rangle\Bigg].
\end{aligned} \tag{28}$$

Take two arbitrary unit vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, and if we let $\mathbf{v}_1 = \sum_{j=1}^{n^t_{N_e+1}} \theta_{1j}\mathbf{x}^t_{(N_e+1)j}$ and $\mathbf{v}_2 = \sum_{j'=1}^{n^t_{N_e+1}} \theta_{2j'}\mathbf{x}^t_{(N_e+1)j'}$, we have

$$\begin{aligned}
&\big|\langle (\mathbf{G}_{\mathbf{S}} - \mathbf{G}_{\mathbf{S}^{e_0}})\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\rangle\big| \\
=&\frac{1}{r_W^2 N_e^2}\Bigg[\sum_{e\neq e_0}^{N_e}\langle \mathbf{v}_1^T, c_e\mathbf{W}_e\rangle\langle c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}, \mathbf{v}_2^T\rangle + \sum_{e'\neq e_0}^{N_e}\langle \mathbf{v}_1^T, c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle c_{e'}\mathbf{W}_{e'}, \mathbf{v}_2^T\rangle \\
&\quad + \langle \mathbf{v}_1^T, \mathbf{W}_\epsilon\rangle\langle c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}, \mathbf{v}_2^T\rangle + \langle \mathbf{v}_1^T, c_{e_0}\mathbf{W}_{e_0} - c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle \mathbf{W}_\epsilon, \mathbf{v}_2^T\rangle \\
&\quad + \langle \mathbf{v}_1^T, c_{e_0}\mathbf{W}_{e_0}\rangle\langle c_{e_0}\mathbf{W}_{e_0}, \mathbf{v}_2^T\rangle - \langle \mathbf{v}_1^T, c_{e'_0}\mathbf{W}_{e'_0}\rangle\langle c_{e'_0}\mathbf{W}_{e'_0}, \mathbf{v}_2^T\rangle\Bigg] \\
\leq&\frac{1}{r_W^2 N_e^2}\Bigg[\sum_{e\neq e_0}^{N_e}\|\mathbf{v}_1^T\|\|c_e\mathbf{W}_e\|(\|c_{e0}\mathbf{W}_{e0}\| + \|c_{e'0}\mathbf{W}_{e'0}\|)\|\mathbf{v}_2^T\| + \sum_{e'\neq e_0}^{N_e}\|\mathbf{v}_1^T\|(\|c_{e0}\mathbf{W}_{e0}\| + \|c_{e'0}\mathbf{W}_{e'0}\|)\|c_{e'}\mathbf{W}_{e'}\|\|\mathbf{v}_2^T\| \\
&\quad + \|\mathbf{v}_1^T\|\|\mathbf{W}_\epsilon\|(\|c_{e0}\mathbf{W}_{e0}\| + \|c_{e'0}\mathbf{W}_{e'0}\|)\|\mathbf{v}_2^T\| + \|\mathbf{v}_1^T\|(\|c_{e0}\mathbf{W}_{e0}\| + \|c_{e'0}\mathbf{W}_{e'0}\|)\|\mathbf{W}_\epsilon\|\|\mathbf{v}_2^T\| \\
&\quad + \|\mathbf{v}_1^T\|\|c_{e0}\mathbf{W}_{e0}\|^2\|\mathbf{v}_2^T\|\Bigg] \\
\leq&\frac{4(4N_e - 3 + r_\epsilon/r_W)}{N_e^2}\|\mathbf{v}_1^T\|\|\mathbf{v}_2^T\|,
\end{aligned} \tag{29}$$

where the last inequality comes from the assumptions: $\|\mathbf{W}_\epsilon\| \leq r_\epsilon$, and for any e, $c_e \leq 1$ and $\|\mathbf{W}_e\| \leq r_W$. $\|v_1\|$ and $\|v_2\|$ can also be bounded as following,

$$\|\mathbf{v}_1\| = \left\| \sum_{j=1}^{n^t_{N_e+1}} \theta_{1j} \mathbf{x}^t_{(N_e+1)j} \right\| \leq \sum_{j=1}^{n^t_{N_e+1}} |\theta_{1j}| \|\mathbf{x}^t_{(N_e+1)j}\| \leq \sum_{j=1}^{n^t_{N_e+1}} \|\mathbf{x}^t_{(N_e+1)j}\| = (n^t_{N_e+1} r_x)^{1/2}, \tag{30}$$

where the second inequality holds because $\boldsymbol{\theta}_1$ is a unit vector, and the last equality follows the assumption $\|\mathbf{x}^t_{(N_e+1)j}\|^2 \leq r_x$. Similarly, we infer $\|\mathbf{v}_2\| \leq (n^t_{N_e+1} r_x)^{1/2}$. Consequently, we have

$$\|\mathbf{G}_\mathbf{S} - \mathbf{G}_{\mathbf{S}^{e_0}}\|_\infty \leq \frac{4(4N_e - 3 + r_\epsilon / r_W) n^t_{N_e+1} r_x}{N_e^2}. \tag{31}$$

Combining equation (27), the 4-th conclusion of Lemma 1, and inequality (31), we reach the result,

$$\begin{aligned} &\left| l_{emp}(\mathbf{L}(\mathbf{S}), (\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1})) - l_{emp}(\mathbf{L}(\mathbf{S}^{e_0}), (\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1})) \right| \\ \leq & 2(n^t_{N_e+1})^{-2} \lambda^{-1} \|\mathbf{G}_\mathbf{S} - \mathbf{G}_{\mathbf{S}^{e_0}}\|_\infty \|y\|^2_{n^t_{N_e}} \\ \leq & \frac{4(4N_e - 3 + r_\epsilon / r_W) B^2 r_x}{\lambda N_e^2} \sim \mathcal{O}\left(\frac{B^2 r_x}{\lambda N_e} \right) \end{aligned} \tag{32}$$

□

To give the generalization bound upon Theorem 3, we make an assumption on the distribution from which all N_e transfer learning experiences as meta-samples are sampled. For every environment $\mathcal{E}$ we have, all N_e pairs of source and target domains $\mathbf{S} = \{\langle \mathcal{S}_e, \mathcal{T}_e \rangle, \cdots, \langle \mathcal{S}_{N_e}, \mathcal{T}_{N_e} \rangle\}$ are drawn according to an algebraic β-mixing stationary distribution $(\mathbf{D}_\mathcal{E})^{N_e}$, which is not i.i.d.. Intuitively, the algebraical β-mixing stationary distribution (see Definition 2 in (Mohri & Rostamizadeh, 2010)) with the β-mixing coefficient $\beta(m) \leq \beta_0 / m^r$ models the dependence between future samples and past samples by a distance of at least m. The independent block technique initiated by (Bernstein, 1927) has been widely adopted to deal with non-i.i.d. learning problems. By directly applying Corollary 21 in (Mohri & Rostamizadeh, 2010), we give the generalization bound of our algorithm $\mathbf{L}(\mathbf{S})$ in Theorem 4.

Theorem 4. *Let $\delta' = \delta - (N_e)^{\frac{1}{2(r+1)} - \frac{1}{4}}$. Then for any sample $\mathbf{S}$ of size N_e drawn according to an algebraic β-mixing stationary distribution, and $\delta \geq 0$ such that $\delta' \geq 0$, the following generalization bound holds with probability at least $1 - \delta$:*

$$\left| R(\mathbf{L}(\mathbf{S})) - R_{N_e}(\mathbf{L}(\mathbf{S})) \right| < \mathcal{O}\left((N_e)^{\frac{1}{2(r+1)} - \frac{1}{4}} \sqrt{\log(\frac{1}{\delta'})} \right) \tag{33}$$

where $R(\mathbf{L}(\mathbf{S}))$ is the expected risk and $R_{N_e}(\mathbf{L}(\mathbf{S}))$ denotes the empirical risk. A larger mixing parameter r, indicating more independence, would lead to a tighter bound.

References

Baktashmotlagh, Mahsa, Harandi, Mehrtash T, Lovell, Brian C, and Salzmann, Mathieu. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, pp. 769–776, 2013.

Baktashmotlagh, Mahsa, Harandi, Mehrtash T, Lovell, Brian C, and Salzmann, Mathieu. Domain adaptation on the statistical manifold. In *CVPR*, pp. 2481–2488, 2014.

Bernstein, Serge. Sur l'extension du théorème limite du calcul des probabilités aux sommes de quantités dépendantes. *Mathematische Annalen*, 97(1):1–59, 1927.

Bousquet, Olivier and Elisseeff, André. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Golub, Gene H, Hansen, Per Christian, and O'Leary, Dianne P. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

Liu, Tongliang, Yang, Qiang, and Tao, Dacheng. Understanding how feature structure transfers in transfer learning. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence*, pp. 2365–2371, 2017.

Maurer, Andreas. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6(Jun):967–994, 2005.

Mohri, Mehryar and Rostamizadeh, Afshin. Stability bounds for stationary φ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 11(Feb):789–814, 2010.