Simple Transferability Estimation for Regression Tasks (Supplementary Material)

Cuong N. Nguyen[1] **Phong Tran**[2,3] **Lam Si Tung Ho**[4] **Vu Dinh**[5]
Anh T. Tran[2] **Tal Hassner**[6] **Cuong V. Nguyen**[1]

[1]Florida International University, USA [2]VinAI Research, Vietnam [3]MBZUAI, UAE
[4]Dalhousie University, Canada [5]University of Delaware, USA [6]Meta AI, USA

The contents of this supplementary include:

1. **Appendix A.1**: Proof of Lemma 5.1 in the main paper.
2. **Appendix A.2**: Proof of Theorem 5.2 in the main paper.
3. **Appendix A.3**: Proof of Lemma 5.3 in the main paper.
4. **Appendix A.4**: Proof of Theorem 5.4 in the main paper.
5. **Appendix B.1**: More details for the experiment settings in Sections 6.1–6.6 of the main paper.
6. **Appendix B.2**: More details for the experiment setting in Section 6.7 of the main paper.
7. **Appendix C.1**: An additional experiment to show the usefulness of our theoretical bounds.
8. **Appendix C.2**: Additional experiment results for Section 6.1 of the main paper.
9. **Appendix C.3**: Additional experiment results for Section 6.2 of the main paper.

A MATHEMATICAL PROOFS

A.1 PROOF OF LEMMA 5.1

Denote $A^*, b^* = \underset{A,b}{\operatorname{argmin}} \left\{ \frac{1}{n_t} \sum_{i=1}^{n_t} \|y_i^t - Az_i - b\|^2 + \lambda \|A\|_F^2 \right\}$.

For all k, we have:

$$\begin{aligned}
\sqrt{\mathcal{L}(w^*, k^*; \mathcal{D}_t)} &\leq \sqrt{\mathcal{L}(w^*, k; \mathcal{D}_t)} && \text{(definition of } k^*\text{)} \\
&= \left[\frac{1}{n_t} \sum_{i=1}^{n_t} \|y_i^t - k(w^*(x_i^t))\|^2 \right]^{1/2} && \text{(definition of } \mathcal{L}\text{)} \\
&\leq \left[\frac{1}{n_t} \sum_{i=1}^{n_t} \|y_i^t - A^* z_i - b^*\|^2 \right]^{1/2} + \left[\frac{1}{n_t} \sum_{i=1}^{n_t} \|A^* z_i + b^* - k(w^*(x_i^t))\|^2 \right]^{1/2} && \text{(triangle inequality)} \\
&\leq \sqrt{-\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)} + \left[\frac{1}{n_t} \sum_{i=1}^{n_t} \|A^* z_i + b^* - k(w^*(x_i^t))\|^2 \right]^{1/2} \\
&= \sqrt{-\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)} + \left[\frac{1}{n_t} \sum_{i=1}^{n_t} \|A^* h^*(w^*(x_i^t)) + b^* - k(w^*(x_i^t))\|^2 \right]^{1/2}. && \text{(definition of } z_i\text{)}
\end{aligned}$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

By choosing $k(\cdot) = A^*h^*(\cdot) + b^*$, the second term in the above inequality becomes 0. This implies $\sqrt{\mathcal{L}(w^*, k^*; \mathcal{D}_t)} \leq \sqrt{-\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)}$ and thus the lemma.

A.2 PROOF OF THEOREM 5.2

First, we need to define the notion of expected (true) risk. Given any model (w, k) for the target task, the expected risk of (w, k) is defined as:

$$\mathcal{R}(w, k) := \mathbb{E}_{(x^t, y^t)\sim\mathbb{P}_t} \left\{ \|y^t - k(w(x^t))\|^2 \right\}. \tag{1}$$

Note that $\text{Tr}(\mathcal{D}_s, \mathbb{P}_t) = -\mathcal{R}(w^*, k^*)$. We prove the uniform bound in Lemma A.1 below that can help us prove Theorem 5.2.

Lemma A.1. *For any $\delta > 0$, with probability at least $1 - \delta$, for all ReLU feed-forward neural network (w, k) of the target task, we have:*

$$|\mathcal{R}(w, k) - \mathcal{L}(w, k; \mathcal{D}_t)| \leq C(d, d_t, M, H, L, \delta)/\sqrt{n_t}.$$

Proof. We recall the definition of Rademacher complexity. Given a real-valued function class $\mathcal{G}$ and a set of data points $\mathcal{D} = \{u_i\}_{i=1}^n$, the (empirical) Rademacher complexity $\widehat{R}_\mathcal{D}(\mathcal{G})$ is defined as:

$$\widehat{R}_\mathcal{D}(\mathcal{G}) = \mathbb{E}_\epsilon \left[\sup_{g\in\mathcal{G}} \frac{1}{n} \sum_{i=1}^n \epsilon_i g(u_i) \right],$$

where $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ is a vector uniformly distributed in $\{-1, +1\}^n$.

In our setting, the hypothesis space Φ is the class of L-layer ReLU feed-forward neural networks whose number of hidden nodes and parameters in each layer are bounded from above by H and $M \geq 1$ respectively. For all $(w, k) \in \Phi$ and x such that $\|x\|_\infty \leq 1$, we have:

$$\|k(w(x))\|_\infty \leq dM^{L+1}H^L.$$

Define $f_{w,k}(x, y) = y - k(w(x))$ and note that $f_{w,k}(x, y) \in \mathbb{R}^{d_t}$. For any $j = 1, 2, \ldots, d_t$, let $[\cdot]_j$ be the projection map to the j-th coordinate. We consider the following real-valued function classes:

$$\begin{aligned}
\mathcal{F} &= \{\|f_{w,k}\|^2 \colon (w, k) \in \Phi\}, \\
\mathcal{F}_j &= \{[f_{w,k}]_j : (w, k) \in \Phi\}, \\
\Phi_j &= \{[k(w(\cdot)]_j : (w, k) \in \Phi\},
\end{aligned}$$

where each element of $\mathcal{F}$ or $\mathcal{F}_j$ is a function with variables (x, y), and each element of Φ_j is a function with variable x. Let $\mathcal{D}_t^x = \{x_i^t\}_{i=1}^{n_t}$ be the set of target inputs. By Theorem 2 of Golowich et al. [2018], for all $j = 1, 2, \ldots, d_t$, we have:

$$\widehat{R}_{\mathcal{D}_t^x}(\Phi_j) \leq 2d_t M^{L+1}H^L \sqrt{\frac{L + 1 + \ln d}{n_t}}.$$

We note that for any $i = 1, 2, \ldots, n_t$, the function $r_i(a) = (a - y_i^t)^2$ mapping from $a \in [-dM^{L+1}H^L, dM^{L+1}H^L]$ to $\mathbb{R}$ is Lipschitz with constant $4dM^{L+1}H^L$. Thus, applying the Contraction Lemma (Lemma 26.9 in Shalev-Shwartz and Ben-David [2014]), we obtain:

$$\widehat{R}_{\mathcal{D}_t}(\mathcal{F}_j) \leq 4dM^{L+1}H^L \widehat{R}_{\mathcal{D}_t^x}(\Phi_j) \leq 8dd_t M^{2L+2}H^{2L} \sqrt{\frac{L + 1 + \ln d}{n_t}}.$$

Therefore,

$$\widehat{R}_{\mathcal{D}_t}(\mathcal{F}) \leq \sum_{j=1}^{d_t} \widehat{R}_{\mathcal{D}_t}(\mathcal{F}_j) \leq 8dd_t^2 M^{2L+2}H^{2L} \sqrt{\frac{L + 1 + \ln d}{n_t}}.$$

Using this inequality, the result of Lemma A.1 follows from Theorem 26.5 in Shalev-Shwartz and Ben-David [2014]. □

To prove Theorem 5.2, we apply Lemma 5.1 in the main paper and Lemma A.1 above for the transferred target model (w^*, k^*). Thus, for any $\lambda \geq 0$ and $\delta > 0$, with probability at least $1 - \delta$, we have:

$$\begin{aligned}\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) &\leq -\mathcal{L}(w^*, k^*; \mathcal{D}_t) \\ &\leq -\mathcal{R}(w^*, k^*) + C(d, d_t, M, H, L, \delta)/\sqrt{n_t} \\ &= \text{Tr}(\mathcal{D}_s, \mathbb{P}_t) + C(d, d_t, M, H, L, \delta)/\sqrt{n_t}.\end{aligned}$$

Therefore, Theorem 5.2 holds.

A.3 PROOF OF LEMMA 5.3

Note that $A_\lambda^*, b_\lambda^* = \underset{A,b}{\text{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \|y_i^t - Ay_i^s - b\|^2 + \lambda \|A\|_F^2 \right\}$.

For all k, we have:

$$\begin{aligned}\sqrt{\mathcal{L}(w^*, k^*; \mathcal{D}_t)} &\leq \sqrt{\mathcal{L}(w^*, k; \mathcal{D}_t)} && \text{(definition of } k^*) \\ &= \left[\frac{1}{n} \sum_{i=1}^{n} \|y_i^t - k(w^*(x_i))\|^2 \right]^{1/2} && \text{(definition of } \mathcal{L}) \\ &\leq \left[\frac{1}{n} \sum_{i=1}^{n} \|y_i^t - A_\lambda^* y_i^s - b_\lambda^*\|^2 \right]^{1/2} + \left[\frac{1}{n} \sum_{i=1}^{n} \|A_\lambda^* y_i^s + b_\lambda^* - k(w^*(x_i))\|^2 \right]^{1/2} && \text{(triangle inequality)} \\ &\leq \sqrt{-\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)} + \left[\frac{1}{n} \sum_{i=1}^{n} \|A_\lambda^* y_i^s + b_\lambda^* - k(w^*(x_i))\|^2 \right]^{1/2}. && \text{(definition of } \widehat{\mathcal{T}}_\lambda^{\text{lab}})\end{aligned}$$

Picking $k(\cdot) = A_\lambda^* h^*(\cdot) + b_\lambda^*$, this inequality becomes:

$$\begin{aligned}\sqrt{\mathcal{L}(w^*, k^*; \mathcal{D}_t)} &\leq \sqrt{-\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)} + \left[\frac{1}{n} \sum_{i=1}^{n} \|A_\lambda^* [y_i^s - h^*(w^*(x_i))]\|^2 \right]^{1/2} \\ &\leq \sqrt{-\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)} + \|A_\lambda^*\|_F \left[\frac{1}{n} \sum_{i=1}^{n} \|y_i^s - h^*(w^*(x_i))\|^2 \right]^{1/2} \\ &= \sqrt{-\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)} + \|A_\lambda^*\|_F \sqrt{\mathcal{L}(w^*, h^*; \mathcal{D}_s)}.\end{aligned}$$

Note that if $a \leq b + c$, then $a^2 \leq 2b^2 + 2c^2$. Applying this fact to the above inequaility, we have:

$$\mathcal{L}(w^*, k^*; \mathcal{D}_t) \leq -2\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) + 2\|A_\lambda^*\|_F^2 \mathcal{L}(w^*, h^*; \mathcal{D}_s).$$

Thus, Lemma 5.3 holds.

A.4 PROOF OF THEOREM 5.4

For any $\lambda \geq 0$ and $\delta > 0$, applying Lemma A.1 for (w^*, k^*) and Lemma 5.3, with probability at least $1 - \delta$:

$$\begin{aligned}\mathcal{R}(w^*, k^*) &\leq \mathcal{L}(w^*, k^*; \mathcal{D}_t) + C(d, d_t, M, H, L, \delta)/\sqrt{n} \\ &\leq -2\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) + 2\|A_\lambda^*\|_F^2 \, \mathcal{L}(w^*, h^*; \mathcal{D}_s) + C(d, d_t, M, H, L, \delta)/\sqrt{n}.\end{aligned}$$

Since $\text{Tr}(\mathcal{D}_s, \mathbb{P}_t) = -\mathcal{R}(w^*, k^*)$, Theorem 5.4 holds.

B MORE DETAILS FOR EXPERIMENT SETTINGS

B.1 MORE DETAILS FOR SECTIONS 6.1–6.6

For these experiments, we train our source models from scratch using the MSE loss with the AdamW optimizer [Loshchilov and Hutter, 2019], which we run for 40 epochs with batch size of 64 and the cosine learning rate scheduler. To obtain good source models, we resize all input images to 256×256 and apply basic image augmentations without horizontal flipping (i.e., affine transformation, Gaussian blur, and color jitter). We also scale all labels into $[0, 1]$ using the width and height of the input images.

For the transfer learning setting with head re-training, we freeze the trained feature extractor and re-train the regression head on the target dataset using the same setting above, except that we run 15 epochs on the CUB-200-2011 dataset and 30 epochs on the OpenMonkey dataset. For half fine-tuning, we unfreeze the last convolution layer and the head classifier since the number of trainable parameters is around half of the total number of parameters. For full fine-tuning, we unfreeze the whole network. In these two fine-tuning settings, we fine-tune for 15 epochs on both datasets. We use PyTorch [Paszke et al., 2019] for implementation.

B.2 MORE DETAILS FOR SECTION 6.7

For this experiment, we use the following 8 ImageNet pre-trained models as the source models: ResNet50, ResNet101, ResNet152 [He et al., 2016], DenseNet121, DenseNet169, DenseNet201 [Huang et al., 2017], GoogleNet [Szegedy et al., 2015], and Inceptionv3 [Szegedy et al., 2016]. These models are taken from the PyTorch Model Zoo.

We use the dSprites dataset [Matthey et al., 2017] for the target task. This dataset contains 737,280 images with 4 outputs for regression: x and y positions, scale, and orientation. The train-test split is similar to the settings in You et al. [2021]: 60% for training, 20% for validation, and 20% for testing. The transferred MSE is computed on the test set. We train our models with 10 epochs using the AdamW optimizer. The initial learning rate is 10^{-3}, which is divided by 10 every 3 epochs.

C ADDITIONAL EXPERIMENT RESULTS

C.1 USEFULNESS OF THEORETICAL BOUNDS

Although the theoretical bounds in Section 5 show the relationships between the transferability of the optimal transferred model and our transferability estimators, these bounds could be loose in practice unless the number of samples is large. This is in fact a limitation of this type of generalization bounds. To show the usefulness of our bounds in practice, we conduct an experiment to investigate the generalization gap using the head re-training setting in Section 6.1.

The generalization gap is defined as the *difference between our transferability score and the negative MSE (the transferability) of the transferred model*. According to our theorems, this generalization gap is bounded above by the complexity term. We will compare the generalization gap with the absolute value of our transferability score and also inspect whether it has any significant correlation with the actual transferred MSE.

From this experiment, the ratios between the absolute value of transferability score and the generalization gap for our transferability estimators are: 1.6 (LinMSE0), 2.0 (LinMSE1), 2.3 (LabMSE0), and 2.3 (LabMSE1). These results show that the transferability scores dominate the generalization gap in practice. More importantly, there is *no significant correlation* between the generalization gap and the actual transferred MSE. These findings indicate that the complexity term in our bounds may have little effects for transferability estimation, as opposed to the transferability score term that has a strong effect (shown by the high correlations in our main experiments).

C.2 ADDITIONAL RESULTS FOR SECTION 6.1

Detailed correlation plots for Table 1. In Figures C.1, C.2, and C.3, we show the detailed correlation plots and p-values for our experiment results reported in Table 1 of the main paper. From these plots, all correlations are statistically significant with $p < 0.001$, except for TransRate and LabTransRate with head re-training.

Additional results with non-linear correlation metrics. In Tables C.1 and C.2, we report the Kendall's-τ and Spearman

Table C.1: **Kendall's-τ correlation coefficients when transferring from OpenMonkey to CUB-200-2011**. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods. Our estimators improve up to 28.4% in comparison with SotA (LogME) while being 13% better on average.

Transfer setting	Label-based method				Feature-based method			
	LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
Head re-training	0.728	0.028	**0.935***	0.924	0.906	0.104	0.896	0.922*
Half fine-tuning	0.525	0.392	0.644	0.646*	0.651	0.291	**0.667***	0.646
Full fine-tuning	0.497	0.289	0.606*	0.594	0.611	0.328	**0.616***	0.594

Table C.2: **Spearman correlation coefficients when transferring from OpenMonkey to CUB-200-2011**. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods. Our estimators improve up to 19.9% in comparison with SotA (LogME) while being 9.7% better on average.

Transfer setting	Label-based method				Feature-based method			
	LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
Head re-training	0.857	0.102	**0.994***	0.991	0.988	0.215	0.984	0.990*
Half fine-tuning	0.726	0.409	0.857	0.858*	0.857	0.437	**0.865***	0.858
Full fine-tuning	0.689	0.433	0.826*	0.823	**0.827***	0.474	**0.827***	0.823

correlation coefficients to complement the results in Table 1 of the main paper. These coefficients, as described in Bolya et al. [2021], are used to assess the ranking associations or the monotonic relationships between the transferability measures and the model performance. Based on the findings presented in these tables, our proposed scores are generally on par with or outperform the current state-of-the-art (SotA) approach, LogME [You et al., 2021], with an average correlation improvement of 9.7% and 13% for Spearman and Kendall's-τ coefficients, respectively. This serves as a strong evidence illustrating the effectiveness of our proposed measures, not only in the linear relationship assessment, but also in the non-linear one.

Additional result with high-dimensional labels. Using the setting in Section 6.1, we also conducted an additional experiment where both source and target tasks have 10-dimensional labels. In particular, we train a source model to predict five OpenMonkey keypoints: *right eye*, *left eye*, *nose*, *head*, and *neck* simultaneously (i.e., this source model returns a 10-dimensional output). The source model is then transferred to a target task that predicts a combination of five CUB-200-2011 keypoints. We consider each combination of 5 keypoints among 10 CUB-200-2011 keypoints as a target task, resulting in 252 target tasks that all have 10-dimensional labels.

We also run 3 transfer learning algorithms: head re-training, half fine-tuning, and full fine-tune, using the same training settings as in Section 6.1. For TransRate and LabTransRate, we use 2 bins per dimension instead of 5 bins to reduce the computational costs. The results for this experiment are reported in Table C.3. From these results, our approaches are better than the baselines for both λ values.

C.3 ADDITIONAL RESULTS FOR SECTION 6.2

Detailed correlation plots for Table 2. In Figures C.4– C.9, we show the detailed correlation plots and p-values for our experiment results reported in Table 2 of the main paper. From these plots, all correlations are statistically significant with $p < 0.001$, except for TransRate and LabTransRate as well as the full fine-tuning setting on the CUB-200-2011 dataset.

Additional result for each individual source task. We report in Tables C.5 and C.6 more comprehensive results for all source tasks on CUB-200-2011 and OpenMonkey respectively. Each row of the tables corresponds to one source task and shows the correlation coefficients when transferring to all other tasks in the respective dataset. From the tables, our transferability estimators are consistently better than LogME, LabLogME, TransRate, and LabTransRate for most source tasks on both datasets. These results confirm the effectiveness of our proposed methods.

Additional result with high-dimensional labels. In this additional experiment, we further show the effectiveness of our proposed methods when the target tasks have higher dimensional labels. In particular, we transfer from 4 source tasks on CUB-200-2011 (*back*, *beak*, *belly*, and *breast*) to all the combinations of 5 attributes among the remaining tasks (except for *right eye*, *right leg*, and *right wing*, which may not always be available in the data). In total, we have 224 source-target pairs,

Table C.3: **Correlation coefficients when transferring between 10d-output tasks from OpenMonkey to CUB-200-2011**. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods. All correlations are statistically significant with $p < 0.001$. Our estimators with both λ values are better than SotA (LogME).

Transfer setting	Label-based method				Feature-based method			
	LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
Head re-training	0.970	0.719	0.991*	0.989	0.968	0.656	0.990	**0.995***
Half fine-tuning	0.944	0.742	0.963*	0.943	0.954	0.684	**0.980***	0.958
Full fine-tuning	0.878	0.736	0.892*	0.863	0.892	0.669	**0.916***	0.881

Table C.4: **Correlation coefficients when transferring from 2d-output tasks to 10d-output tasks on CUB-200-2011**. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods. Except for TransRate with half and full fine-tuning, all correlations are statistically significant with $p < 0.001$. Our estimators are better than SotA (LogME) in most cases.

Transfer setting	Label-based method				Feature-based method			
	LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
Head re-training	0.602	0.632	0.868*	0.816	0.885	0.549	0.901	**0.973***
Half fine-tuning	0.491	0.645	0.771	0.881*	0.804	0.072	**0.913***	0.818
Full fine-tuning	0.397	0.632	0.727	**0.888***	0.756	0.050	0.884*	0.833

where the source tasks have 2-dimensional labels and the target tasks have 10-dimensional labels. We use the same training settings as in Section 6.2 of the main paper, except that we also use 2 bins per dimension when calculating TransRate and LabTransRate to reduce computational costs. Table C.4 reports the results for this experiment. These results clearly show that our methods, LinMSE0 and LinMSE1, are better than the LogME and TransRate baselines in most cases.

References

Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. In *Advances in Neural Information Processing Systems*, 2021.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Annual Conference on Learning Theory*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing Sprites dataset, 2017. `https://github.com/deepmind/dsprites-dataset/`.

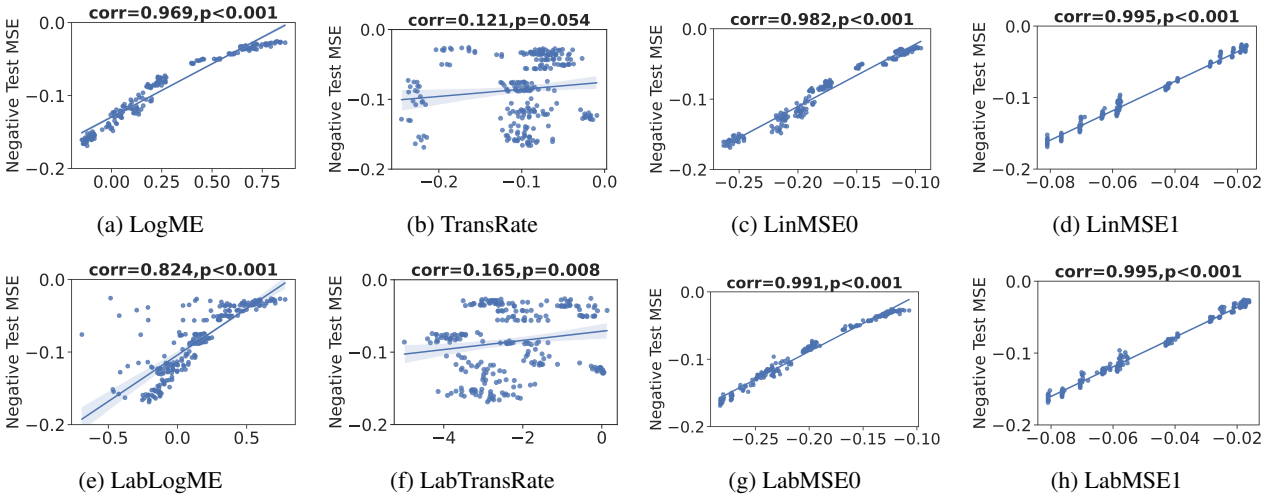
Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

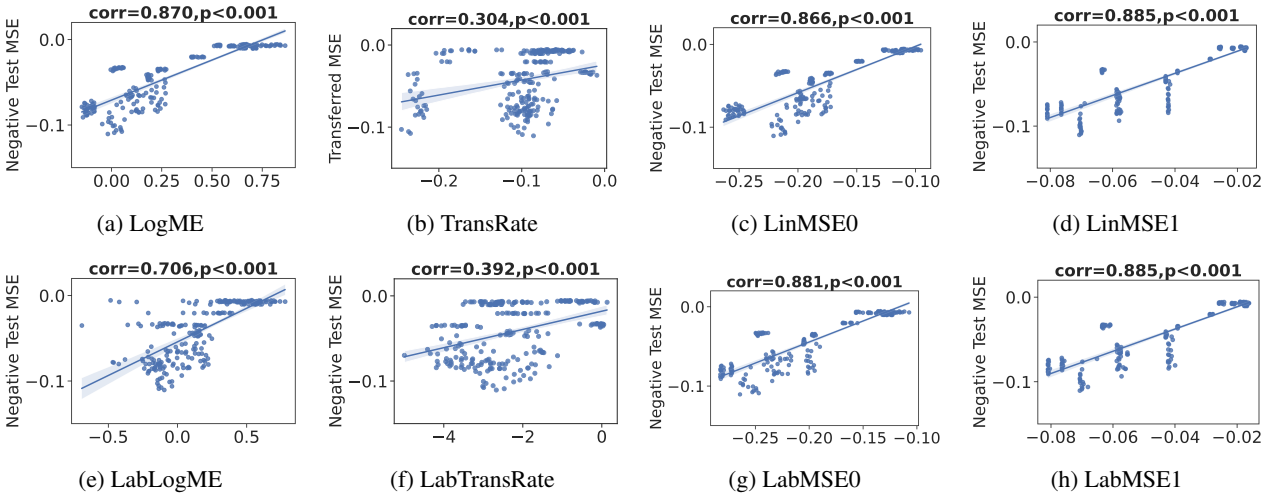
Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, 2021.



(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

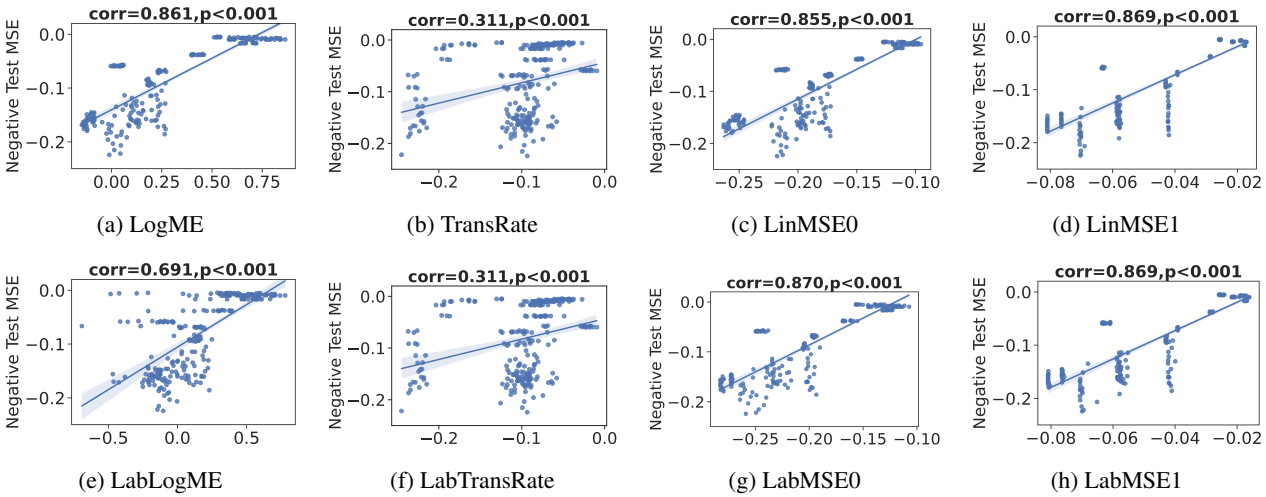
Figure C.1: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with head re-training from OpenMonkey to CUB-200-2011.



(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

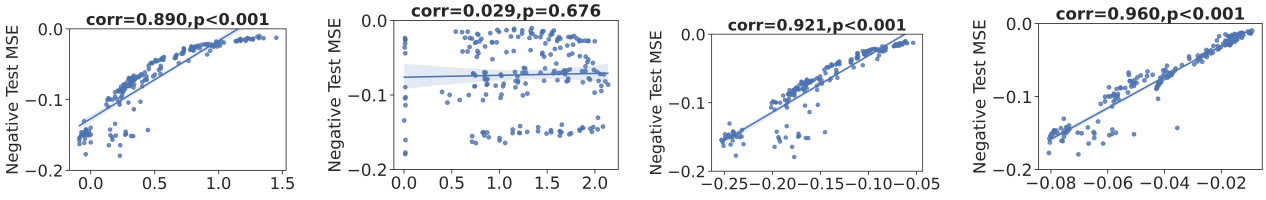
Figure C.2: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with half fine-tuning from OpenMonkey to CUB-200-2011.



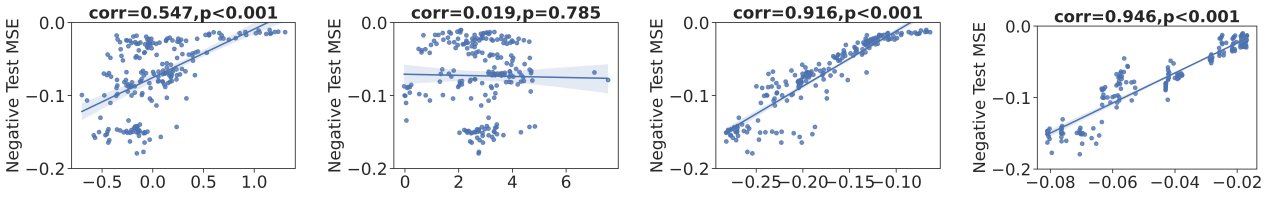
(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

Figure C.3: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with full fine-tuning from OpenMonkey to CUB-200-2011.

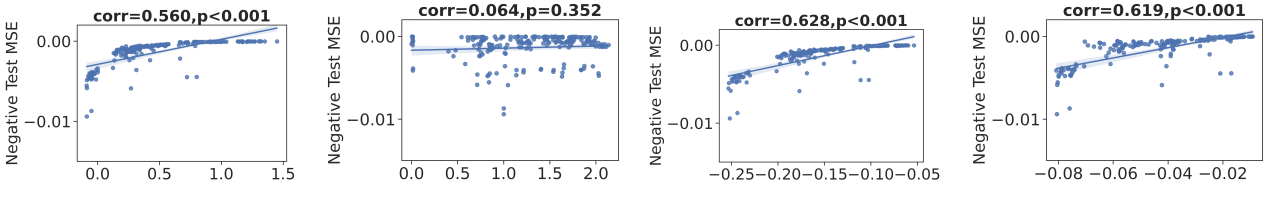


(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

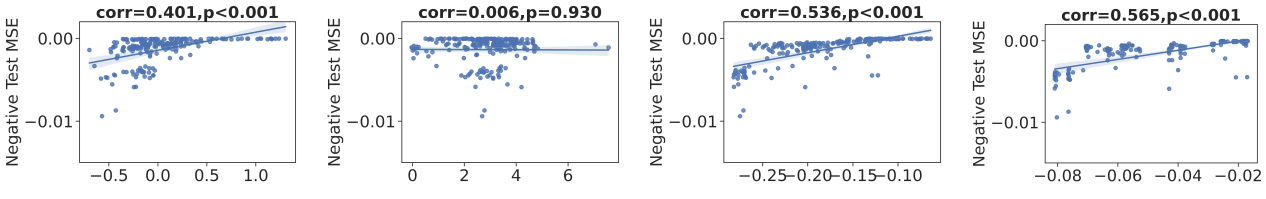


(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

Figure C.4: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with head re-training between any two different keypoints (with shared inputs) on CUB-200-2011.

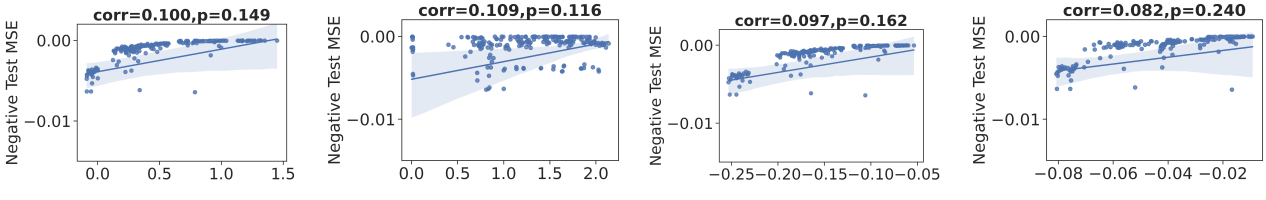


(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

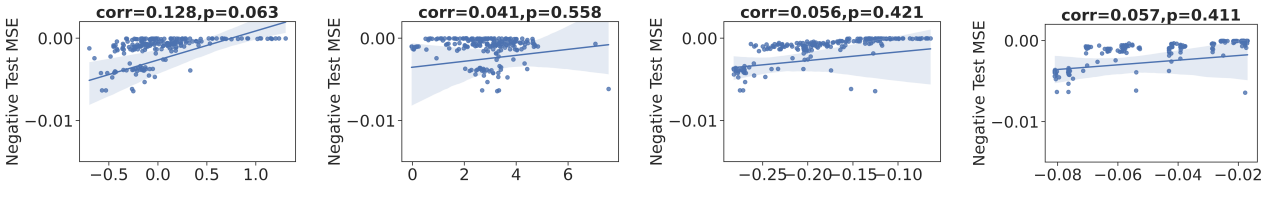


(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

Figure C.5: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with half fine-tuning between any two different keypoints (with shared inputs) on CUB-200-2011.

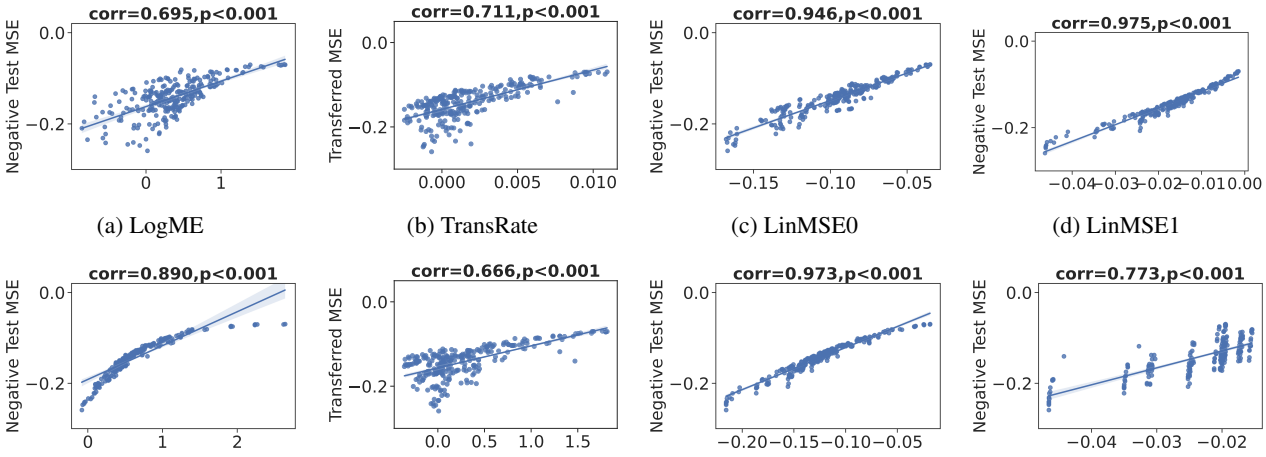


(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1



(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

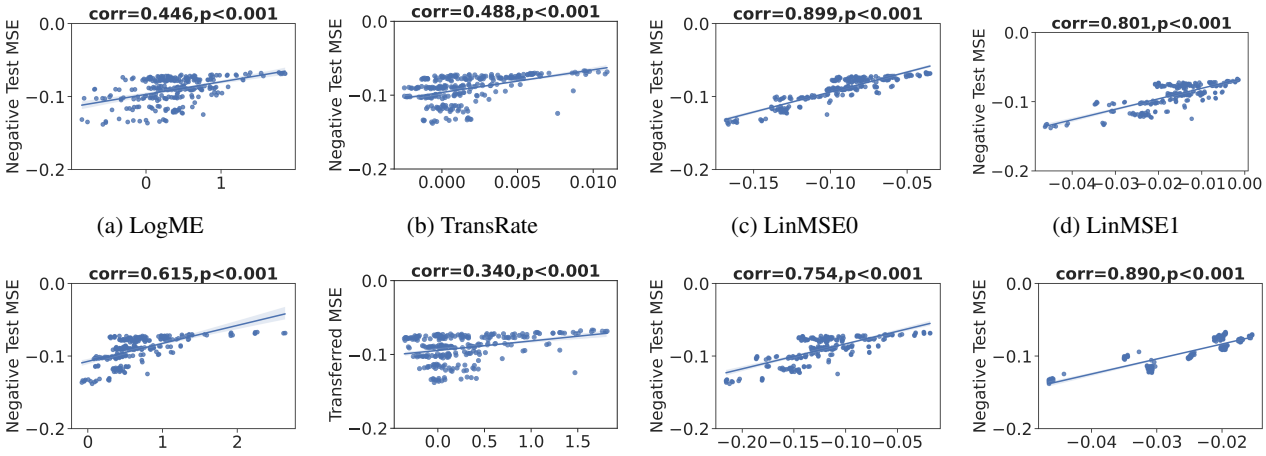
Figure C.6: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with full fine-tuning between any two different keypoints (with shared inputs) on CUB-200-2011.



(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

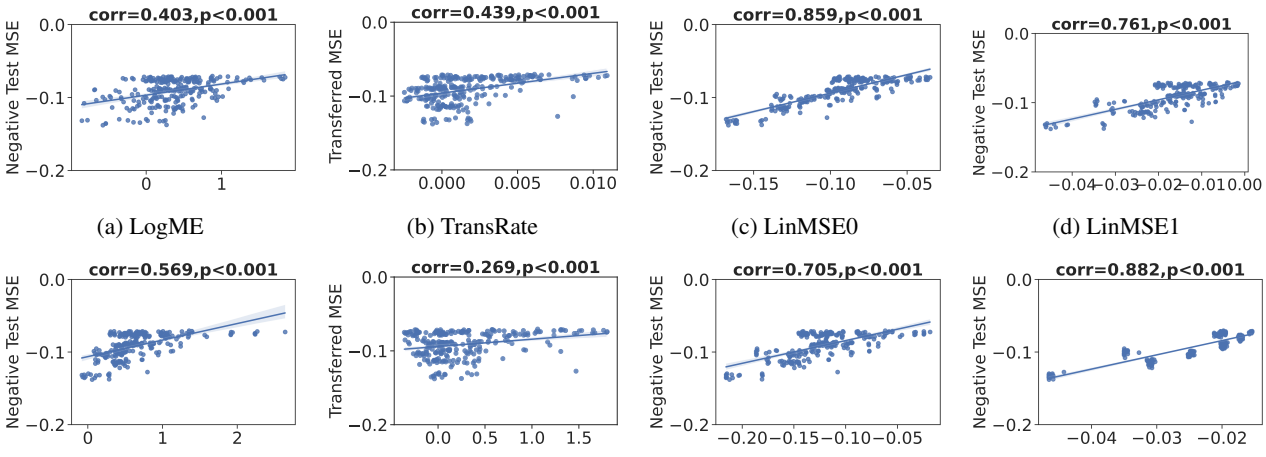
Figure C.7: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with head re-training between any two different keypoints (with shared inputs) on OpenMonkey.



(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

Figure C.8: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with half fine-tuning between any two different keypoints (with shared inputs) on OpenMonkey.



(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

(e) LabLogME (f) LabTransRate (g) LabMSE0 (h) LabMSE1

Figure C.9: **Correlation coefficients and p-values between transferability estimators and negative test MSEs** when transferring with full fine-tuning between any two different keypoints (with shared inputs) on OpenMonkey.

Table C.5: **Correlation coefficients for all source tasks** on CUB-200-2011. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods.

Transfer setting	Source task	Label-based method				Feature-based method			
		LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
Head re-training	Back	0.743	0.116	0.956	**0.966***	0.920	0.273	0.931	0.964*
	Beak	0.863	0.229	0.922*	0.915	0.878	0.158	0.906	**0.945***
	Belly	0.892	0.097	0.970	**0.982***	0.933	0.188	0.932	**0.982***
	Breast	0.915	0.120	0.935	0.945*	0.903	0.279	0.922	**0.961***
	Crown	0.917	0.041	0.962	0.966*	0.913	0.251	0.945	**0.979***
	Forehead	0.888	0.076	0.941*	0.939	0.885	0.221	0.924	**0.966***
	Left eye	0.035	0.076	0.913	0.964*	0.924	0.289	0.945	**0.969***
	Left leg	0.261	0.221	0.935	**0.975***	0.935	0.223	0.953	**0.975***
	Left wing	0.260	0.170	0.964	**0.994***	0.980	0.173	**0.994***	**0.994***
	Nape	0.889	0.085	0.922	0.942*	0.900	0.300	0.929	**0.953***
	Right eye	0.625	0.242	0.904	0.974*	0.921	0.244	0.948	**0.975***
	Right leg	0.508	0.047	0.958	0.989*	0.942	0.217	0.954	**0.990***
	Right wing	0.521	0.167	0.907	0.979*	0.935	0.270	0.946	**0.980***
	Tail	0.591	0.392	0.900	**0.927***	0.872	0.544	0.880	0.890*
	Throat	0.896	0.124	0.938	0.941*	0.890	0.291	0.924	**0.956***
Half fine-tuning	Back	0.714	0.076	0.791	0.814*	0.835	0.168	**0.911***	0.873
	Beak	0.663	0.160	0.831*	0.772	0.765	0.076	0.883	**0.899***
	Belly	0.528	0.233	0.655	0.752*	0.758	0.309	**0.849***	0.764
	Breast	0.730	0.100	0.802*	0.779	0.762	0.152	**0.867***	0.850
	Crown	0.644	0.068	0.752	0.776*	0.714	0.165	**0.832***	0.816
	Forehead	0.654	0.032	0.804*	0.786	0.727	0.120	0.859	**0.873***
	Left eye	0.420	0.046	**0.913***	0.853	0.812	0.227	0.892*	0.865
	Left leg	0.121	0.095	0.721	0.819*	0.845	0.150	**0.893***	0.832
	Left wing	0.352	0.150	**0.949***	0.918	0.859	0.189	0.919*	0.918
	Nape	0.660	0.055	0.705	0.770*	0.751	0.181	**0.863***	0.802
	Right eye	0.561	0.221	**0.911***	0.873	0.786	0.180	0.871	0.890*
	Right leg	0.268	0.125	0.690	0.804*	0.810	0.069	**0.861***	0.820
	Right wing	0.407	0.133	0.495	0.613*	0.516	0.338	0.521	**0.617***
	Tail	0.801	0.117	0.930*	0.812	0.848	0.285	0.924	**0.968***
	Throat	0.767	0.013	0.870*	0.810	0.811	0.253	**0.900***	0.873
Full fine-tuning	Back	0.710	0.085	0.785	0.808*	0.829	0.178	**0.906***	0.868
	Beak	0.659	0.161	0.826*	0.780	0.758	0.073	0.877	**0.899***
	Belly	0.645	0.273	0.782	0.847*	0.862	0.365	**0.926***	0.856
	Breast	0.740	0.104	0.811*	0.791	0.768	0.152	**0.871***	0.859
	Crown	0.647	0.073	0.756	0.784*	0.717	0.157	**0.834***	0.821
	Forehead	0.648	0.037	0.799*	0.783	0.723	0.111	0.855	**0.869***
	Left eye	0.224	**0.456***	0.297	0.347	0.333*	0.246	0.282	0.326
	Left leg	0.057	0.067	0.659	0.769*	0.796	0.146	**0.850***	0.783
	Left wing	0.342	0.159	**0.954***	0.915	0.860	0.195	0.920*	0.914
	Nape	0.667	0.041	0.713	0.779*	0.752	0.177	**0.864***	0.810
	Right eye	0.549	0.213	**0.915***	0.876	0.794	0.199	0.877	0.893*
	Right leg	0.237	0.377	0.673	0.692*	0.755	0.431	**0.766***	0.693
	Right wing	**0.254***	0.046	0.237	0.223	0.225	0.093	0.227*	0.220
	Tail	0.803	0.122	0.930*	0.818	0.846	0.288	0.923	**0.969***
	Throat	0.665	0.027	0.801*	0.779	0.744	0.256	**0.850***	0.834

Table C.6: **Correlation coefficients for all source tasks** on OpenMonkey. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods.

Transfer setting	Source task	Label-based method				Feature-based method			
		LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
Head retraining	Right eye	0.894	0.859	**0.986***	0.835	0.918	0.846	0.978	**0.986***
	Left eye	0.895	0.854	**0.987***	0.838	0.868	0.858	0.981	**0.987***
	Nose	0.908	0.849	0.988*	0.849	0.818	0.837	0.978	**0.989***
	Head	0.941	0.881	**0.992***	0.821	0.897	0.884	0.983*	0.978
	Neck	0.972	0.862	**0.998***	0.887	0.932	0.839	0.982	0.987*
	Right shoulder	0.977	0.837	**0.994***	0.891	0.842	0.811	0.982*	0.980
	Right elbow	0.963	0.529	**0.994***	0.940	0.469	0.564	0.969	0.990*
	Right wrist	0.970	0.753	**0.993***	0.939	0.615	0.446	0.963	0.990*
	Left shoulder	0.972	0.800	**0.997***	0.915	0.823	0.808	0.988*	0.988*
	Left elbow	0.960	0.546	**0.994***	0.948	0.711	0.572	0.969	0.989*
	Left wrist	0.975	0.597	**0.993***	0.951	0.964	0.544	0.963	**0.993***
	Hip	0.922	0.540	0.989*	0.325	0.874	0.557	0.800	**0.991***
	Right knee	0.925	0.080	0.975*	0.850	0.766	0.331	0.945	**0.993***
	Right ankle	0.931	0.411	0.989*	0.770	0.737	0.371	0.930	**0.997***
	Left knee	0.923	0.160	0.978*	0.848	0.692	0.209	0.936	**0.994***
	Left ankle	0.916	0.416	0.986*	0.775	0.852	0.329	0.925	**0.998***
	Tail	0.936	0.712	**0.993***	0.312	0.821	0.662	0.897	0.990*
Half fine-tuning	Right eye	0.795	0.734	0.906*	0.883	0.835	0.709	**0.963***	0.923
	Left eye	0.797	0.731	0.905*	0.879	0.771	0.719	**0.960***	0.918
	Nose	0.829	0.736	0.914*	0.872	0.649	0.721	**0.968***	0.916
	Head	0.835	0.759	0.921*	0.882	0.804	0.751	**0.964***	0.928
	Neck	0.902	0.793	0.929*	0.871	0.745	0.765	**0.969***	0.915
	Right shoulder	0.887	0.725	0.924*	0.890	0.751	0.758	**0.972***	0.924
	Right elbow	0.764	0.250	0.806	0.914*	0.048	0.602	**0.931***	0.821
	Right wrist	0.806	0.501	0.823	0.903*	0.172	0.643	**0.929***	0.819
	Left shoulder	0.893	0.718	0.927*	0.899	0.702	0.774	**0.972***	0.930
	Left elbow	0.782	0.369	0.824	0.919*	0.366	0.594	**0.946***	0.839
	Left wrist	0.822	0.523	0.828	0.902*	0.765	0.663	**0.932***	0.824
	Hip	0.030	0.487	0.233	**0.910***	0.006	0.359	0.800*	0.305
	Right knee	0.481	0.429	0.598	**0.906***	0.186	0.067	0.831*	0.687
	Right ankle	0.357	0.275	0.534	**0.910***	0.286	0.226	0.806*	0.632
	Left knee	0.467	0.355	0.601	**0.899***	0.172	0.215	0.855*	0.692
	Left ankle	0.331	0.242	0.530	**0.904***	0.197	0.303	0.822*	0.632
	Tail	0.231	0.196	0.434	**0.829***	0.160	0.121	0.729*	0.494
Full fine-tuning	Right eye	0.796	0.711	0.905*	0.894	0.821	0.694	**0.959***	0.927
	Left eye	0.790	0.734	0.904*	0.882	0.763	0.714	**0.957***	0.921
	Nose	0.810	0.731	0.912*	0.892	0.642	0.709	**0.960***	0.932
	Head	0.801	0.737	0.900*	0.892	0.772	0.718	**0.947***	0.920
	Neck	0.893	0.782	0.930*	0.886	0.755	0.743	**0.962***	0.926
	Right shoulder	0.896	0.722	0.936*	0.908	0.759	0.750	**0.975***	0.940
	Right elbow	0.689	0.168	0.736	0.878*	0.047	0.562	**0.888***	0.761
	Right wrist	0.796	0.505	0.805	0.876*	0.199	0.644	**0.910***	0.803
	Left shoulder	0.872	0.690	0.901*	0.882	0.670	0.762	**0.955***	0.903
	Left elbow	0.726	0.282	0.774	0.904*	0.326	0.538	**0.914***	0.797
	Left wrist	0.787	0.488	0.787	0.868*	0.725	0.672	**0.903***	0.785
	Hip	0.016	0.518	0.173	**0.894***	0.038	0.382	0.757*	0.238
	Right knee	0.391	0.518	0.516	**0.891***	0.096	0.141	0.763*	0.614
	Right ankle	0.246	0.396	0.437	**0.889***	0.185	0.340	0.726*	0.546
	Left knee	0.381	0.448	0.521	**0.891***	0.149	0.303	0.789*	0.618
	Left ankle	0.244	0.297	0.444	**0.871***	0.098	0.357	0.751*	0.551
	Tail	0.105	0.299	0.309	**0.824***	0.047	0.212	0.628*	0.372