Supplemental Document for Combining Task Predictors via Enhancing Joint Predictability

Kwang In Kim[1][0000−0002−6470−4571], Christian Richardt[2][0000−0001−6716−9845], and Hyung Jin Chang[3][0000−0001−7495−9677]

[1] UNIST, Korea [2] University of Bath, UK [3] University of Birmingham, UK

In this supplemental document, we present:

1. Details of the marginal likelihood calculation used in the automatic determination of relevant predictors Σ_L (Sec. 1.1);
2. A summary of our predictor combination algorithm (Sec. 1.2);
3. A detailed discussion of baseline algorithms, including:
   (a) our adaptation of Mejjati et al.'s multi-task learning (*MTL*) algorithm [6] (Sec. 2.1),
   (b) a derivation of Kim et al.'s original predictor combination (*OPC*) algorithm [5,4] (Sec. 2.2), and
   (c) Evgeniou et al.'s graph Laplacian (*GL*)-based MTL algorithm and its adaptation to predictor combination (Sec. 2.3).

In the main paper, we only presented the ranking results for the first 10 attributes in each dataset. In Section 3, we provide the complete experimental results, including additional results of *GL* and tests of statistical significance of the accuracy improvements made by different algorithms. We reproduce some content from the main paper to make this document self-contained.

1 Details of the main algorithm

1.1 Calculating the marginal likelihood for linear Gaussian process prediction

Suppose that we have the following linear and nonlinear anisotropic covariance functions:

$$k_L(\mathbf{a},\mathbf{b}) = \mathbf{a}^\top \Sigma_L \mathbf{b}, \tag{1}$$

$$k_A(\mathbf{a},\mathbf{b}) = \exp\big(-(\mathbf{a}-\mathbf{b})^\top \Sigma_A (\mathbf{a}-\mathbf{b})\big), \tag{2}$$

where $\Sigma_L = \mathrm{diag}[\boldsymbol{\sigma}]$, the diagonal matrix with elements $\boldsymbol{\sigma} = [\sigma^1,...,\sigma^n]^\top$. Σ_A is defined similarly. Our goal is to maximize the marginal likelihood $p(\mathbf{f}\,|\,G,\Sigma_L)$ of the sampled predictor $\mathbf{f}$ given the reference matrix G with respect to $\boldsymbol{\sigma}$. The log

marginal likelihood $\log(p(\mathbf{f}\,|\,G,\Sigma_L))$ of linear Bayesian regression with Gaussian prior and i.i.d. Gaussian noise model is given as [7]:

$$\begin{aligned}\log(p(\mathbf{f}\,|\,G)) = &-\frac{1}{2}\log\left|G\cdot\mathrm{diag}[\boldsymbol{\sigma}]\cdot G^\top+\lambda I\right|-\frac{N}{2}\log(2\pi)\\ &-\frac{1}{2}\mathbf{f}^\top(G\cdot\mathrm{diag}[\boldsymbol{\sigma}]\cdot G^\top+\lambda I)^{-1}\mathbf{f}.\end{aligned}\tag{3}$$

As maximizing $p(\mathbf{f}\,|\,G,\Sigma_L)$ is equivalent to minimizing $-\log(p(\mathbf{f}\,|\,G,\Sigma_L))$, and the second term in $\log(p(\mathbf{f}\,|\,G,\Sigma_L))$ is independent of $\boldsymbol{\sigma}$, we can find the optimal parameter vector $\boldsymbol{\sigma}^*$ by minimizing the following energy:

$$\begin{aligned}\mathcal{E}(\boldsymbol{\sigma}) &= \log\left|G\cdot\mathrm{diag}[\boldsymbol{\sigma}]\cdot G^\top+\lambda I\right|+\mathbf{f}^\top(G\cdot\mathrm{diag}[\boldsymbol{\sigma}]\cdot G^\top+\lambda I)^{-1}\mathbf{f}\\ &= N\log|\lambda|+\sum_{i=1}^{n}\log(\sigma^i)+\log\left|\mathrm{diag}[1./\boldsymbol{\sigma}]+G^\top G/\lambda\right|\\ &\quad+\mathbf{f}^\top\left(\frac{1}{\lambda}I-\frac{1}{\lambda^2}G\left(\mathrm{diag}[1./\boldsymbol{\sigma}]+\frac{1}{\lambda}G^\top G\right)^{-1}G^\top\right)\mathbf{f},\end{aligned}\tag{4}$$

where the second equation is obtained by applying the Sherman–Morrison–Woodbury formula [9] to both summands of $\mathcal{E}$, and '$1./\boldsymbol{\sigma}$' is the element-wise reciprocal of $\boldsymbol{\sigma}$. Since $\frac{1}{\lambda}\mathbf{f}^\top\mathbf{f}$ and $N\log|\lambda|$ are also independent of $\boldsymbol{\sigma}$, minimizing $\mathcal{E}$ is equivalent to minimizing

$$\begin{aligned}\mathcal{E}'(\boldsymbol{\sigma}) = &\sum_{i=1}^{n}\log(\sigma^i)+\log\left|\mathrm{diag}[1./\boldsymbol{\sigma}]+G^\top G/\lambda\right|\\ &-\frac{1}{\lambda^2}\mathbf{f}^\top\left(G\left(\mathrm{diag}[1./\boldsymbol{\sigma}]+\frac{1}{\lambda}G^\top G\right)^{-1}G^\top\right)\mathbf{f}.\end{aligned}\tag{5}$$

Since this energy $\mathcal{E}'$ is a continuously differentiable function of $\boldsymbol{\sigma}$, it can be minimized by standard gradient descent. Figure 1 shows example parameters $\boldsymbol{\sigma}^*$ optimized for the *Pubfig* dataset. For each of the 11 attributes in *Pubfig* as a target, we optimized the corresponding parameters $\boldsymbol{\sigma}$ using the remaining attributes as references, plus 5 additional randomly generated references. As indicated by small magnitudes and the corresponding standard deviations of the $\boldsymbol{\sigma}^*$ entries, our algorithm successfully disregards these irrelevant references. For comparison, we also show the corresponding parameters optimized for the anisotropic Gaussian kernel k_A (Eq. 2), demonstrating that once normalized, their relative scaling behaviors are similar, i.e.

$$\Sigma_A^* \approx \frac{\Sigma_L^*}{\sigma_k^2}\tag{6}$$

for a global scaling parameter σ_k^2. Our final algorithm uses Σ_L^*/σ_k^2 as a surrogate to Σ_A^*, using σ_k^2 as hyperparameter.

1.2 Algorithm summary

Given the reference matrix G, the initial predictor $\mathbf{f}^0$, and hyperparameters (noise variance σ^2 in Eq. 8; global kernel scaling σ_k^2 in Eq. 6; regularization parameter λ_J

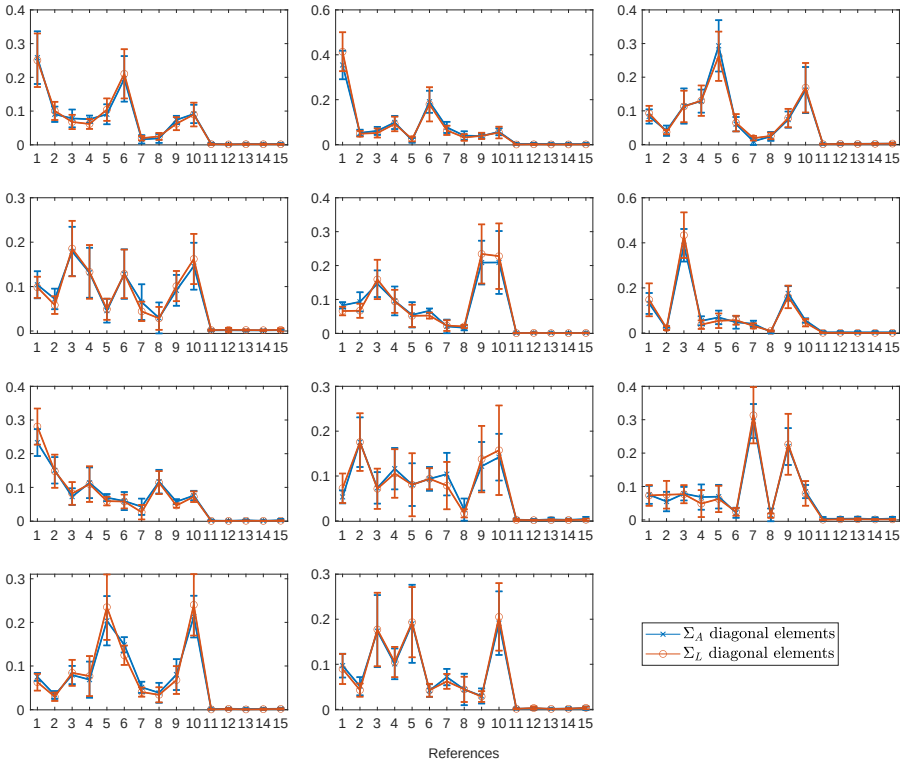

Fig. 1. The average diagonal values of Σ_A and Σ_L optimized for each attribute in the *Pubfig* dataset as the target with the remaining 10 attributes in the same dataset (references 1 to 10), plus 5 additional randomly generated attributes (references 11 to 15) as references. The values of Σ_A and Σ_L are normalized such that the respective sums total to one. Note that the unrelated references (11–15) are correctly detected as irrelevant (small magnitudes and standard deviations) and hence ignored. In addition, the linear kernel Σ_L (orange) is highly correlated to the anisotropic kernel Σ_A (blue), so we use a scaled version of Σ_L as a surrogate to the optimal Σ_A^* (Eq. 6).

in Eq. 7), our algorithm constructs a denoised predictor by iteratively maximizing the objective

$$\mathcal{O}_{\mathrm{N}}(\mathbf{f}) = \frac{\mathbf{f}^\top A \mathbf{f}}{\mathbf{f}^\top C_N \mathbf{f}} \quad \text{with} \tag{7}$$

$$A = (C_N \mathbf{f}^t)(C_N \mathbf{f}^t)^\top + \lambda_{\mathrm{J}} Q' \quad \text{and}$$

$$Q' = C_N(2K(K+\sigma^2 I)^{-1} - (K+\sigma^2 I)^{-1} K K (K+\sigma^2 I)^{-1}) C_N. \tag{8}$$

Algorithm 1 summarizes this process.

Algorithm 1 Nonlinear predictor combination

Input: Initial predictor $\mathbf{f}^0$, references $\{\mathbf{g}_i\}_{i=1}^R$, noise variance σ^2 (Eq. 8), global kernel scaling σ_k^2, regularization parameter λ_J (Eq. 7), and iteration number S.

1: $\mathcal{H}^0=\{\mathbf{h}_0^0,\mathbf{h}_1^0,...,\mathbf{h}_R^0\}\Leftarrow\{\mathbf{f}^0,\mathbf{g}_1,...,\mathbf{g}_R\}$;
2: Calculate the kernel parameter matrix $(\Sigma_L^*)_i$ for each $\mathbf{h}_i^0\in\mathcal{H}^0$; $(\Sigma_A)_i=(\Sigma_L^*)_i/\sigma_k^2$;
3: **for** step $t\in\{0,...,S-1\}$ **do**
4: **for** reference $i\in\{0,...,R\}$ **do**
5: Calculate $\mathbf{h}_i^{t+1}\in\mathcal{H}^{t+1}$ by maximizing $\mathcal{O}_\mathrm{N}$ based on $\mathcal{H}^t$ (Eq. 7);
6: **end for**
7: **end for**

Output: Denoised target predictor $\mathbf{f}^*=\mathbf{h}_0^S$.

2 (Adapting) Existing algorithms

2.1 Mejjati et al.'s *MTL* algorithm

Mejjati et al.'s *MTL* algorithm considers each task-specific predictor as a random variable. Then, the relationships between tasks are modeled based on the statistical dependence estimated by evaluating these predictor random variables on a dataset X [6]. Adopting a nonparametric measure of statistical dependence, the finite set independence criterion (FSIC) [3], *MTL* enables training multiple predictors independently of their parametric forms and, therefore, it can be applied to predictor combination problems.

Applying this algorithm to the predictor combination setting, we construct the initial predictor matrix H^0 by stacking column-wise, the initial target predictor $\mathbf{f}^0$ and the references $\{\mathbf{g}_1,...,\mathbf{g}_R\}$

$$H^0=[\mathbf{f}^0,\mathbf{g}_1,...,\mathbf{g}_R]. \tag{9}$$

MTL then refines the initial predictor matrix H^0 by minimizing the energy

$$\mathcal{E}_\mathrm{M}(H)=\left\|H-H^0\right\|_\mathrm{F}^2-\lambda_1\|\mathrm{vec}(\Phi(H))\|_2^2+\lambda_2\|\mathrm{vec}(\Phi(H))\|_1, \tag{10}$$

where $\mathrm{vec}(A)$ constructs a vector by concatenating columns of matrix A, $\Phi(H)$ is an $(R+1)\times(R+1)$-sized matrix consisting of pairwise FSIC evaluations: $\Phi(H)_{[i,j]}$ takes a large positive value when $H_{[:,i]}$ and $H_{[:,j]}$ exhibit strong statistical dependence and it takes 0 when $H_{[:,i]}$ and $H_{[:,j]}$ are independent as realizations of random variables. Minimizing $\mathcal{E}_\mathrm{M}$ strengthens overall task dependence via (negation of) the L^2 norm of $\mathrm{vec}(\Phi(H))$ and, at the same time, introduces sparsity in the task dependence via the L^1 norm of $\Phi(H)$. Combing these two terms, *MTL* selectively enforces task dependence while suppressing the dependence of weakly related tasks as outliers. As $\mathcal{E}_\mathrm{M}$ is not differentiable, standard gradient-descent type algorithms are not applicable. Instead, it is minimized based on the alternating direction method of multipliers (ADMM) approach. This involves iteratively solving ADMM sub-problems [1], with the number of total iterations S as a hyperparameter. Once the optimal predictor matrix H^* is constructed, the final denoised predictor is obtained by extracting the first column of H^*: $\mathbf{f}^*=H^*_{[:,1]}$. Similarly to our algorithm, S is determined by setting the maximum number of iterations at 50

and selecting the iteration number achieving the highest validation accuracy. The other two hyperparameters, λ_1 and λ_2, are tuned based on validation accuracy.

2.2 Derivation of Kim et al.'s algorithm (*OPC*).

Kim et al.'s original predictor combination (*OPC*) approach iteratively minimizes the following energy (Eq. 1 in the main paper):

$$\mathcal{E}_{\mathrm{O}}(f) = D_{\mathrm{KL}}(f \,|\, f^t)^2 + \lambda_{\mathrm{O}} \sum_{i=1}^{R} w_i D_{\mathrm{KL}}(f \,|\, g_i)^2, \tag{11}$$

$$w_i = \exp\left(- \frac{D_{\mathrm{KL}}(f^t \,|\, g_i)^2}{\sigma_{\mathrm{O}}^2} \right), \tag{12}$$

with $\lambda_{\mathrm{O}}, \sigma_{\mathrm{O}}^2 > 0$ being hyperparameters. We present how this algorithm is obtained as an instance of Hein and Maier's *Manifold Denoising* algorithm [2] by discretizing a diffusion process on a predictor manifold $\mathcal{M}$.

Manifold denoising [2]. Suppose that we have a set of data points $\mathcal{H}^0 = \{\mathbf{h}_i^0\}_{i=1}^n$ presented as a sample from a Euclidean space $\mathbb{R}^d$ and, further, that the points in $\mathcal{H}^0$ are sampled from an underlying data-generating manifold $\mathcal{M}$ embedded in $\mathbb{R}^d$ ($\imath(\mathcal{M}) \subset \mathbb{R}^d$ with $\imath$ being the embedding), and they are observed as a subset of $\mathbb{R}^d$ contaminated with i.i.d. Gaussian noise $\boldsymbol{\epsilon}$ in $\mathbb{R}^d$:

$$\mathbf{h}_i^0 = \imath(\widetilde{\mathbf{h}}_i^0) + \boldsymbol{\epsilon} \in \mathbb{R}^d \quad \text{for} \quad \widetilde{\mathbf{h}}_i^0 \in \mathcal{M}. \tag{13}$$

The manifold denoising algorithm denoises $\mathcal{H}^0$ by simulating diffusion on a graph G that discretizes $\mathcal{M}$ (each point $\mathbf{h}_i^0 \in \mathcal{H}^0$ forms a vertex of G):

$$\frac{\partial H}{\partial t} = -\delta L H, \tag{14}$$

where $H = [\mathbf{h}_1, ..., \mathbf{h}_n]^\top$ and L is the graph Laplacian:

$$L = I - D^{-1} W, \tag{15}$$

$$W_{[i,j]} = \exp\left(- \frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{\sigma^2} \right), \tag{16}$$

and D is a diagonal matrix consisting of row sums of W, such that $D_{ii} = \sum_{j=1} W_{[i,j]}$. Now discretizing Eq. 14 using the implicit Euler method, we obtain

$$H^{t+1} - H^t = -\delta L H^{t+1}. \tag{17}$$

At each time step t, the solution H^{t+1} of Eq. 17 is obtained as the minimizer of the following energy:

$$\mathcal{E}_{\mathrm{D}}(H) = \left\| H - H^t \right\|_{\mathrm{F}}^2 + \delta \mathrm{tr}[H^\top L H], \tag{18}$$

where $\|A\|_{\mathrm{F}}$ and $\mathrm{tr}[A]$ are the Frobenius norm and trace of matrix A, respectively. As the number of data points n grows to infinity, G becomes a precise representation of $\mathcal{M}$ embedded in $\mathbb{R}^d$, and L converges to the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ on $\mathcal{M}$ casting Eq. 17 into a diffusion process on a continuous manifold

$\mathcal{M}$ [2]. It should be noted that the graph Laplacian was constructed based on the *ambient* L^2 distance in $\mathbb{R}^d$ rather than the intrinsic metric on $\mathcal{M}$. This facilitates building a practical, still consistent algorithm: Equation 18 only requires the ambient Euclidean distance (via L) without having to access $\mathcal{M}$ directly, but it guarantees the statistical consistency of L as proven by Hein and Maier [2].

Now applying this algorithm to the predictor combination setting and, therefore, assuming that only the first point $\mathbf{h}_1^0 \in \mathcal{H}^0$ is noisy (i.e. $\mathbf{h}_i = \imath(\tilde{\mathbf{h}}_i)$ for $i = \{2,...,n\}$), we obtain an iterative update rule of $\mathbf{h}_1^t$ given fixed *references* $\{\mathbf{h}_i\}_{i=2}^n$:

$$\mathcal{E}_{\mathrm{D}}(\mathbf{h}) = \left\|\mathbf{h} - \mathbf{h}_1^t\right\|^2 + \delta \sum_{i=2}^{n} W_{[1,i]} \|\mathbf{h} - \mathbf{h}_i\|^2. \tag{19}$$

Finally, $\mathcal{E}_{\mathrm{O}}$ in Eq. 11 is obtained by replacing each point in $\mathcal{H}^t$ and the corresponding L^2 distances in $\mathcal{E}_{\mathrm{D}}$ with a Gaussian process predictor and Kullback–Leibler divergences, respectively: $\mathbf{h}_1^t$ and $\{\mathbf{h}_i\}_{i=2}^n$ (with $n = R+1$) are considered as the target predictor $\mathbf{f}$ and the corresponding references $\{\mathbf{g}_i\}$, respectively.

2.3 Evgeniou et al.'s graph Laplacian (*GL*)-based MTL algorithm.

Evgeniou et al.'s graph Laplacian (*GL*)-based algorithm learns predictors $\mathcal{H} = \{h_i\}_{i=1}^n$ of multiple tasks by enforcing pairwise parameter similarities: Assuming that all predictors are linear, i.e. $h_i(\mathbf{x}) = \mathbf{w}_i^\top \mathbf{x}$, their algorithm estimates the predictor parameters $W = \{\mathbf{w}_1,...,\mathbf{w}_n\}$ by minimizing the energy

$$\mathcal{E}_{\mathrm{GL}}(W) = \sum_{i=1}^{n} l_i(h_i) + \lambda_1 \sum_{i=1}^{n} \|\mathbf{w}_i\|^2 + \lambda_2 \sum_{i=1}^{n} \sum_{j \neq i} U_{[i,j]} \|\mathbf{w}_i - \mathbf{w}_j\|^2, \tag{20}$$

where $\{l_i(\cdot)\}_{i=1}^n$ are task-specific loss functions and $U_{[i,j]} \geq 0$ represents the relationship between tasks i and j. Now adapting this algorithm to the predictor combination setting, we assume that the initial predictor $f^0 = h_1$ and the corresponding references $g_i = h_{i+1}$ for $i \in \{1,...,R\}$ are given ($n = R+1$). Then, f^0 is refined by minimizing the energy

$$\mathcal{E}_{\mathrm{GL}}(\mathbf{w}) = \left\|\mathbf{w} - \mathbf{w}_1^0\right\|^2 + \lambda_{\mathrm{GL}} \sum_{j=2}^{n} U_{[1,j]} \|\mathbf{w} - \mathbf{w}_j\|^2. \tag{21}$$

In general, determining the task relationship parameters $\{U_{[1,j]}\}$ is a challenging problem. Here, we determine them by adopting Kim et al.'s approach: We iteratively update $\{U_{[1,j]}\}$ by minimizing $\mathcal{E}_{\mathrm{GL}}$ at each time step t with

$$U_{[1,j]} = \exp\left(-\frac{\|\mathbf{w}_i - \mathbf{w}_1^t\|^2}{\sigma_{\mathrm{GL}}^2}\right). \tag{22}$$

Further, adopting Kim and Chang's approach [4], we explicitly constrain all predictor parameter vectors to have unit norm: $\|\mathbf{w}_i\| = 1$, enabling the comparison of task predictor parameters independently of their scales. The two hyperparameters σ_{GL}^2 and λ_{GL} are determined based on validation accuracy. As often, ranking problems are nonlinear, we extend this framework by adopting

the linear-in-parameter model:

$$f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}_f, \tag{23}$$

where $\phi\colon \mathcal{X} \to \mathcal{F}_k$ with $\mathcal{F}_k$ being the reproducing kernel Hilbert space (RKHS) corresponding to a Gaussian kernel with hyperparameter σ_k^2 [8]:

$$k(\mathbf{a},\mathbf{b}) = \exp\left(-\frac{\|\mathbf{a}-\mathbf{b}\|^2}{\sigma_k^2}\right). \tag{24}$$

In this case, the target predictor f is represented based the original parameter vector $\mathbf{w}_f$ as well as its dual parameter vector $\mathbf{a}_f = [a_f^1,\ldots,a_f^{N'}]^\top$:

$$f(\mathbf{x}) := \phi(\mathbf{x})^\top \mathbf{w}_f = \sum_{j=1}^{N'} a_f^j k(\mathbf{b}_j,\mathbf{x}), \tag{25}$$

with $\{\mathbf{b}_i\}_{i=1}^{N'}$ being a set of *basis vectors*. The reference predictors $\{g_i\}_{i=1}^R$ are represented similarly:

$$g_i(\mathbf{x}) := \phi(\mathbf{x})^\top \mathbf{w}_i = \sum_{j=1}^{N'} a_i^j k(\mathbf{b}_j,\mathbf{x}). \tag{26}$$

Under this setting, the parameter similarity $\|\mathbf{w}_f - \mathbf{w}_i\|$ can be calculated using the standard kernel trick [8] as

$$\|\mathbf{w}_f - \mathbf{w}_j\| = \mathbf{a}_f^\top K \mathbf{a}_f^\top + \mathbf{a}_i^\top K \mathbf{a}_i^\top - 2\mathbf{a}_f^\top K \mathbf{a}_i^\top, \tag{27}$$

with $K_{[i,j]} = k(\mathbf{b}_i,\mathbf{b}_j)$. It should be noted that efficient[1] calculation of $\|\mathbf{w}_f - \mathbf{w}_i\|$ based on Eq. 27 requires that all predictors should share the same RKHS determined by the kernel parameter σ_k. To facilitate this, in our experiments, we first determine f^0 as nonlinear rank support vector machine that minimizes the regularized energy

$$\mathcal{E}_\mathrm{S}(f) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in U} \mathcal{L}(f,(\mathbf{x}_i,\mathbf{x}_j)) + C_f \|\mathbf{w}\|^2, \tag{28}$$

$$\mathcal{L}(f,(\mathbf{a},\mathbf{b})) = \max(1-(f(\mathbf{a})-f(\mathbf{b})),0)^2 \tag{29}$$

for the rank loss $\mathcal{L}$ defined on ground-truth ranked pairs $U \subset X \times X$ and tune the hyperparameters σ_k^2 and C_f based on validation accuracy. Once f^0 is fixed in this way, the reference predictors $\{g_i\}_{i=1}^R$ are determined by minimizing $\mathcal{E}_\mathrm{S}$ for the respective rank labels. However, for these references, only the respective regularization hyperparameters $\{C_i\}_{i=1}^R$ are tuned while the corresponding kernel parameters are all fixed as σ_k^2 (optimized for f^0), to facilitate the computation of $\|\mathbf{w} - \mathbf{w}_i\|$ (Eq. 27). We fixed N' at 500 and selected the basis vectors $\{\mathbf{b}_i\}_{i=1}^{N'}$ as the cluster centers of input data points X, estimated using k-means clustering.

Note that this setting violates the application conditions of predictor combination: It requires access to the forms of all predictors $\{f,g_1,\ldots,g_R\}$ and, further,

[1] The RKHS $\mathcal{F}_k$ corresponding to a Gaussian kernel k is infinite-dimensional. Therefore, each parameter vector $\mathbf{w} \in \mathcal{F}_k$ is an infinite-dimensional object, making the direct evaluation of $\|\mathbf{w}_f - \mathbf{w}_j\|$ infeasible.

it assumes that all predictors share the same form (Eqs. 25 and 26). We show in Sec. 3 that the latter homogeneity requirement poses a severe limitation on predictor combination performance. Even when *GL* took advantage of known predictor forms, except for a few cases, the other predictor combination algorithms significantly outperformed *GL*. Often, the results of *GL* are even worse than the initial predictors $\mathbf{f}^0$ that are obtained by selecting the best predictors (via validation) from the heterogeneous predictor pools.

3 Complete ranking results

Table 1 summarizes the results for the relative attributes ranking experiments (see Tables 2–6 for complete results). Our algorithm *NPC* performs best for 87% (162/186) of attributes. In particular, it showed statistically significant improvement on 74 out of 80 *AWA2* attributes, while the baselines *OPC* and *MTL* achieved significant performance gains only on 8 and 33 attributes, respectively.[2]

Overall, our algorithm *NPC* is often statistically significantly better than these methods and – apart from only one attribute (for *CUB*) out of 186 – ours is not statistically significantly worse than the other methods. This demonstrates that the baselines *OPC* and *MTL* are limited in that they can only capture pairwise dependence between the target predictor and each reference. Taking into account the dependence present among the references, and thereby *jointly* exploiting them in improving the target predictor, our algorithm *NPC (Ours)* significantly improves the performance.

References

1. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1–122 (2010). https://doi.org/10.1561/2200000016 4
2. Hein, M., Maier, M.: Manifold denoising. In: NIPS. pp. 561–568 (2007) 5, 6
3. Jitkrittum, W., Szabó, Z., Gretton, A.: An adaptive test of independence with analytic kernel embeddings. In: PMLR (Proc. ICML). pp. 1742–1751 (2017) 4
4. Kim, K.I., Chang, H.J.: Joint manifold diffusion for combining predictions on decoupled observations. In: CVPR. pp. 7549–7557 (2019) 1, 6
5. Kim, K.I., Tompkin, J., Richardt, C.: Predictor combination at test time. In: ICCV. pp. 3553–3561 (2017) 1
6. Mejjati, Y.A., Cosker, D., Kim, K.I.: Multi-task learning by maximizing statistical dependence. In: CVPR. pp. 3465–3473 (2018) 1, 4

[2] We used a t-test with $\alpha = 0.95$. Note that statistical significance tests do not necessarily evaluate how significant the improvements are in an absolute scale: Even when the improvements are marginal, if they are consistent, the result of statistical significant tests can be positive. For instance, for *AWA2* attribute 4, *MTL* achieved rather moderate improvements (with mean 0.05) but the test of statistical significance is positive as the results consistently improved the performance from the baseline, as indicated by the small standard deviation.

Table 1. A summary of the results of statistical significance tests of our method NPC compared to baseline $\mathbf{f}^0$, *GL*, *OPC* and *MTL*, based on a t–test with $\alpha = 0.95$. For each method, we show #attributes where our *NPC* is statistically significantly better (first column), on par with (second column), and statistically significantly worse (third column).

Dataset	vs. baseline $\mathbf{f}^0$			vs. *GL*			vs. *OPC*			vs. *MTL*			# total attr.
Shoes	9	1	0	10	0	0	8	2	0	9	1	0	10
Pubfig	11	0	0	11	0	0	11	0	0	11	0	0	11
OSR	6	0	0	2	4	0	4	2	0	5	1	0	6
OSR (ResNet)	6	0	0	6	0	0	6	0	0	6	0	0	6
aPascal	25	4	0	16	13	0	9	20	0	19	10	0	29
CUB	31	9	0	23	17	0	26	14	0	12	27	1	40
AWA2	74	6	0	73	7	0	71	9	0	72	8	0	80
Zap50K	0	4	0	0	4	0	0	4	0	0	4	0	4
# total attr.	162	24	0	141	45	0	135	51	0	134	51	1	186

7. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA (2006) 2
8. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge, MA (2002) 7
9. Sherman, J., Morrison, W.J.: Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. The Annals of Mathematical Statistics **21**(1), 124–127 (1950) 2

Table 2. Ranking accuracies of different predictor combination algorithms on the *Shoes*, *Pubfig*, *OSR*, and *OSR (ResNet)* datasets. For each dataset, we repeated experiments 10 times with different training, validation, and test set splits. For baseline $\mathbf{f}^0$ (second column), Kendall's Tau correlations×100 (standard deviations in parentheses) are presented. For the remaining algorithms (third to sixth columns), the accuracy offsets from $\mathbf{f}^0$ are presented. The best and second best results are highlighted with **bold** and *italic* fonts, respectively. The results of statistical significance test based on a t–test with $\alpha = 0.95$ are highlighted in green (significantly positive) and orange (significantly negative). The last three columns show the results of statistical significance test of our algorithm with *GL*, *OPC*, and *MTL*, respectively (+/−: significantly positive/negative).

Shoes

Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	72.09 (1.71)	-0.36 (1.44)	*2.36 (0.79)*	2.03 (0.49)	**3.21 (0.86)**	+	+	+
2	63.84 (1.87)	-2.04 (2.94)	*1.57 (1.25)*	0.70 (0.39)	**2.26 (1.38)**	+	0	+
3	38.07 (2.11)	-1.38 (2.43)	-0.24 (0.65)	*0.16 (0.70)*	**4.58 (2.45)**	+	+	+
4	*50.10 (2.75)*	-2.45 (2.59)	-0.88 (3.29)	-0.08 (0.51)	**1.63 (2.25)**	+	+	+
5	65.76 (1.20)	-1.11 (1.84)	-0.05 (0.16)	*0.13 (0.34)*	**0.92 (2.46)**	+	0	0
6	65.02 (1.83)	-0.86 (1.13)	0.68 (0.87)	*0.81 (0.86)*	**4.18 (1.54)**	+	+	+
7	59.38 (2.06)	-3.31 (1.59)	*0.78 (1.16)*	0.45 (0.42)	**4.14 (3.20)**	+	+	+
8	56.85 (2.04)	-2.57 (1.46)	0.19 (0.46)	*0.40 (0.51)*	**2.62 (0.87)**	+	+	+
9	65.15 (1.94)	0.35 (1.94)	*2.49 (1.27)*	1.40 (0.72)	**4.58 (1.72)**	+	+	+
10	72.10 (1.24)	-1.26 (1.55)	*1.71 (0.77)*	1.47 (0.77)	**2.75 (1.05)**	+	+	+

Pubfig

Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	67.13 (2.75)	4.89 (2.48)	8.37 (3.84)	*9.37 (2.94)*	**15.45 (2.59)**	+	+	+
2	62.49 (2.41)	-0.96 (2.62)	-0.31 (0.91)	*2.24 (1.55)*	**13.78 (3.23)**	+	+	+
3	68.31 (2.33)	2.27 (3.27)	3.06 (2.52)	*6.25 (3.04)*	**11.33 (3.10)**	+	+	+
4	63.98 (3.46)	*8.44 (4.26)*	4.88 (3.14)	7.84 (3.49)	**17.80 (4.26)**	+	+	+
5	61.27 (2.96)	*6.15 (2.92)*	3.33 (4.23)	3.26 (3.70)	**16.62 (6.02)**	+	+	+
6	81.60 (1.26)	-1.09 (2.67)	-0.03 (1.25)	*0.44 (1.58)*	**6.17 (2.03)**	+	+	+
7	64.23 (2.88)	2.87 (3.13)	1.68 (2.67)	*3.14 (4.61)*	**15.66 (3.14)**	+	+	+
8	66.10 (3.53)	*0.38 (2.66)*	0.19 (0.21)	0.10 (0.49)	**12.16 (3.17)**	+	+	+
9	59.73 (4.79)	*3.58 (3.78)*	-0.11 (1.89)	1.96 (3.20)	**17.74 (4.68)**	+	+	+
10	63.58 (3.48)	*5.79 (3.55)*	3.24 (1.92)	4.06 (2.23)	**14.16 (2.58)**	+	+	+
11	69.12 (2.87)	7.76 (2.73)	9.30 (2.48)	*9.49 (2.73)*	**15.20 (2.87)**	+	+	+

OSR

Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	88.57 (0.93)	**3.16 (1.07)**	2.06 (0.83)	2.19 (1.03)	*2.72 (1.39)*	0	0	0
2	87.52 (0.89)	-1.17 (1.06)	-0.00 (0.19)	*0.03 (0.12)*	**0.93 (0.69)**	+	+	+
3	76.12 (0.95)	0.50 (1.32)	1.31 (0.92)	*1.99 (1.26)*	**3.25 (1.49)**	+	+	+
4	77.67 (0.92)	*1.29 (1.11)*	0.69 (0.70)	0.90 (0.68)	**2.10 (1.29)**	0	+	+
5	79.58 (0.65)	*2.50 (0.72)*	2.26 (0.68)	1.43 (0.86)	**2.89 (1.04)**	0	0	+
6	80.49 (1.22)	*0.70 (1.00)*	0.09 (0.52)	0.03 (0.43)	**1.46 (0.84)**	0	+	+

OSR (ResNet)

Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	96.12 (0.59)	-0.07 (0.14)	*0.39 (0.33)*	0.26 (0.37)	**1.33 (0.48)**	+	+	+
2	84.73 (0.87)	0.00 (0.27)	-0.11 (0.25)	*0.10 (0.25)*	**2.51 (1.14)**	+	+	+
3	84.46 (1.08)	-0.01 (0.03)	0.43 (0.30)	*0.89 (0.66)*	**2.56 (1.21)**	+	+	+
4	85.14 (1.27)	-0.09 (0.44)	-0.03 (0.10)	*0.57 (0.89)*	**2.45 (0.78)**	+	+	+
5	88.00 (0.78)	0.03 (0.24)	0.55 (0.82)	*0.62 (0.93)*	**3.52 (1.66)**	+	+	+
6	90.88 (0.88)	-0.08 (0.17)	0.06 (0.25)	*0.71 (0.57)*	**1.56 (1.08)**	+	+	+

Table 3. Ranking accuracies of different predictor combination algorithms on the *aPascal* and *Zap50K* datasets. For each dataset, we repeated experiments 10 times with different training, validation, and test set splits. For baseline $\mathbf{f}^0$ (second column), Kendall's Tau correlations×100 (standard deviations in parentheses) are presented. For the remaining algorithms (third to sixth columns), the accuracy offsets from $\mathbf{f}^0$ are presented. The best and second best results are highlighted with **bold** and *italic* fonts, respectively. The results of statistical significance test based on a t–test with $\alpha = 0.95$ are highlighted in green (significantly positive) and orange (significantly negative). The last three columns show the results of statistical significance test of our algorithm with *GL*, *OPC*, and *MTL*, respectively (+/−: significantly positive/negative).

aPascal								
Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	59.44 (4.19)	3.11 (2.88)	*3.38 (2.31)*	0.87 (0.64)	**6.04 (3.43)**	+	+	+
2	68.21 (4.50)	-0.02 (0.12)	*0.42 (0.87)*	0.27 (0.13)	**1.19 (2.17)**	0	0	0
3	14.45 (4.62)	1.15 (2.40)	*2.09 (5.79)*	0.69 (0.33)	**4.24 (4.67)**	0	0	+
4	65.20 (3.20)	0.23 (0.57)	**1.82 (1.38)**	0.07 (0.04)	*1.19 (2.67)*	0	0	0
5	57.87 (4.38)	2.08 (1.69)	*4.10 (2.03)*	1.19 (0.67)	**4.10 (2.26)**	+	0	+
6	57.37 (5.76)	3.21 (1.49)	**4.18 (1.23)**	1.36 (0.70)	*3.90 (2.28)*	0	0	+
7	71.34 (2.69)	0.05 (0.36)	*2.29 (2.27)*	1.05 (0.53)	**2.91 (1.85)**	+	0	+
8	67.08 (4.56)	2.25 (1.79)	*3.78 (3.30)*	1.65 (0.51)	**3.79 (3.11)**	+	0	+
9	62.36 (4.83)	2.54 (1.51)	*3.91 (2.27)*	2.03 (0.55)	**4.69 (1.34)**	+	0	+
10	57.09 (4.36)	4.79 (2.25)	*5.38 (2.86)*	2.92 (1.55)	**7.00 (2.74)**	+	+	+
11	62.25 (3.59)	2.05 (1.54)	*2.61 (1.39)*	1.76 (0.75)	**4.12 (3.01)**	0	0	+
12	60.58 (4.76)	*3.48 (2.13)*	**4.25 (2.38)**	1.54 (0.67)	3.43 (2.63)	0	0	+
13	46.71 (4.95)	3.41 (3.54)	*4.20 (4.16)*	1.47 (0.60)	**4.86 (3.99)**	0	0	+
14	52.31 (3.52)	3.52 (2.89)	*4.37 (1.68)*	2.29 (1.07)	**6.75 (2.27)**	+	+	+
15	52.07 (6.42)	2.14 (2.77)	*2.34 (2.77)*	1.46 (0.77)	**5.65 (3.71)**	+	+	+
16	47.33 (4.15)	0.46 (0.72)	*1.55 (1.22)*	1.16 (0.41)	**2.76 (2.55)**	+	0	0
17	49.53 (4.57)	-0.18 (0.96)	0.42 (1.70)	*0.95 (0.45)*	**2.49 (2.44)**	+	0	0
18	82.22 (3.20)	0.60 (0.81)	*1.37 (0.70)*	1.10 (0.36)	**1.53 (0.88)**	+	0	0
19	70.22 (3.21)	1.04 (1.05)	1.01 (1.37)	*1.42 (0.61)*	**2.40 (2.19)**	0	0	0
20	79.62 (2.60)	1.87 (0.88)	**2.10 (1.19)**	1.44 (0.58)	*2.09 (2.31)*	0	0	0
21	53.01 (5.18)	*0.69 (1.34)*	0.02 (0.37)	0.37 (0.25)	**3.70 (4.07)**	+	+	+
22	55.52 (3.98)	2.83 (2.30)	*3.04 (1.78)*	2.22 (0.71)	**6.41 (3.92)**	+	+	+
23	70.53 (5.23)	2.31 (2.23)	2.03 (1.14)	*2.34 (1.11)*	**3.82 (2.24)**	+	+	+
24	38.48 (5.51)	*1.12 (2.63)*	0.95 (3.22)	0.91 (0.43)	**1.81 (4.71)**	0	0	0
25	48.45 (3.52)	*0.89 (2.29)*	0.76 (1.65)	0.58 (0.51)	**2.48 (2.88)**	0	+	+
26	49.93 (3.91)	2.53 (2.56)	*3.60 (3.00)*	2.31 (0.55)	**6.00 (2.57)**	+	+	+
27	72.87 (3.05)	0.04 (0.31)	*1.07 (1.49)*	0.91 (0.29)	**1.65 (1.55)**	+	0	0
28	64.35 (2.27)	0.30 (0.66)	**1.39 (1.33)**	0.42 (0.42)	*0.43 (3.09)*	0	0	0
29	53.84 (3.47)	3.97 (2.36)	*4.16 (2.19)*	2.53 (0.78)	**5.17 (2.20)**	0	0	+

Zap50K								
Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	87.97 (0.99)	-0.00 (0.38)	**0.27 (0.60)**	-0.20 (0.67)	*0.27 (0.75)*	0	0	0
2	89.43 (1.56)	*0.13 (0.82)*	-0.27 (1.10)	**0.40 (0.86)**	0.03 (0.95)	0	0	0
3	90.67 (1.29)	0.43 (0.80)	**0.80 (0.83)**	*0.70 (0.84)*	0.67 (1.23)	0	0	0
4	90.33 (1.56)	*0.23 (1.14)*	0.17 (0.98)	0.03 (0.82)	**0.37 (0.87)**	0	0	0

Table 4. Ranking accuracies of different predictor combination algorithms on the *CUB* dataset. We repeated experiments 10 times with different training, validation, and test set splits. For baseline $\mathbf{f}^0$ (second column), Kendall's Tau correlations×100 (standard deviations in parentheses) are presented. For the remaining algorithms (third to sixth columns), the accuracy offsets from $\mathbf{f}^0$ are presented. The best and second best results are highlighted with **bold** and *italic* fonts, respectively. The results of statistical significance test based on a t–test with $\alpha = 0.95$ are highlighted in green (significantly positive) and orange (significantly negative). The last three columns show the results of statistical significance test of our algorithm with *GL*, *OPC*, and *MTL*, respectively (+/−: significantly positive/negative).

Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	68.80 (3.98)	-0.02 (0.07)	-0.07 (0.28)	*0.16 (0.24)*	**1.47 (1.05)**	+	+	+
2	74.83 (3.89)	0.64 (0.66)	*1.12 (0.64)*	0.87 (1.53)	**2.00 (1.10)**	+	+	+
3	78.59 (2.36)	0.86 (0.66)	1.49 (1.10)	*1.93 (1.23)*	**2.22 (1.10)**	+	+	0
4	73.92 (2.51)	-0.17 (0.39)	0.27 (0.42)	**1.82 (1.72)**	*1.38 (2.03)*	+	0	0
5	74.61 (3.37)	1.30 (1.35)	0.98 (1.35)	*2.36 (1.46)*	**2.73 (2.01)**	+	+	0
6	63.86 (5.24)	-0.00 (0.41)	0.58 (0.91)	**1.48 (1.99)**	*0.89 (1.34)*	0	0	0
7	76.97 (2.21)	1.02 (0.70)	0.54 (0.34)	**1.18 (0.66)**	*1.06 (0.79)*	0	+	0
8	62.97 (3.05)	-0.00 (0.04)	*0.26 (0.45)*	0.20 (0.49)	**0.76 (1.11)**	0	0	0
9	72.52 (2.57)	1.05 (0.99)	1.06 (0.57)	*1.53 (1.19)*	**3.08 (1.84)**	+	+	+
10	63.62 (2.99)	0.09 (0.35)	0.50 (1.69)	*0.75 (0.86)*	**2.30 (1.47)**	+	0	+
11	59.70 (3.69)	0.02 (0.30)	0.02 (0.29)	**0.66 (0.98)**	*0.54 (1.30)*	0	0	0
12	71.08 (2.09)	0.17 (0.40)	-0.04 (0.74)	**0.87 (0.95)**	*0.87 (0.72)*	+	+	0
13	78.10 (2.31)	0.25 (0.31)	0.11 (0.41)	**1.88 (1.09)**	*1.31 (1.27)*	+	+	−
14	74.13 (1.90)	1.08 (0.89)	0.48 (0.39)	*1.59 (1.32)*	**1.85 (1.93)**	0	+	0
15	72.23 (3.07)	0.02 (0.62)	0.04 (0.30)	*1.25 (0.92)*	**1.84 (1.28)**	+	+	+
16	73.32 (1.97)	1.02 (1.13)	0.67 (0.67)	**1.65 (1.81)**	*1.54 (2.30)*	0	0	0
17	58.11 (4.61)	0.09 (0.15)	0.18 (0.39)	*0.51 (0.61)*	**1.16 (0.95)**	+	+	+
18	57.35 (4.92)	-0.04 (0.23)	0.29 (0.65)	*0.55 (0.92)*	**0.71 (1.85)**	0	0	0
19	76.67 (3.06)	1.28 (1.15)	0.43 (0.98)	*1.30 (0.94)*	**1.92 (1.47)**	0	+	0
20	76.31 (2.10)	0.28 (0.47)	-0.06 (0.48)	**0.81 (1.37)**	*0.72 (1.53)*	0	0	0
21	75.45 (3.03)	1.21 (0.93)	1.35 (1.54)	*1.88 (1.55)*	**2.28 (1.50)**	0	0	0
22	75.28 (4.13)	1.01 (0.64)	0.49 (0.84)	**1.66 (1.08)**	*1.30 (1.45)*	0	+	0
23	69.67 (3.00)	0.12 (0.43)	0.02 (0.72)	*1.99 (1.36)*	**2.07 (1.74)**	+	+	0
24	76.24 (2.55)	0.87 (0.57)	*1.17 (0.87)*	0.77 (0.74)	**2.16 (1.49)**	+	0	+
25	70.57 (1.93)	-0.04 (0.08)	0.43 (0.76)	*0.94 (1.60)*	**1.21 (1.20)**	+	+	0
26	63.59 (2.97)	-0.09 (0.19)	0.07 (0.42)	*0.52 (1.15)*	**0.82 (0.53)**	+	+	0
27	72.15 (3.19)	*0.39 (0.56)*	0.19 (0.49)	-0.06 (0.41)	**0.41 (0.49)**	0	0	+
28	64.40 (3.29)	0.00 (0.54)	0.29 (0.92)	*1.21 (0.97)*	**1.27 (1.27)**	+	+	0
29	57.52 (2.78)	0.09 (0.24)	0.76 (0.96)	*2.20 (2.42)*	**2.50 (2.58)**	+	+	0
30	56.73 (2.86)	0.41 (0.72)	0.78 (0.93)	**1.98 (1.72)**	*1.89 (2.30)*	0	0	0
31	73.27 (2.96)	1.13 (1.11)	1.14 (0.67)	*1.91 (0.97)*	**2.00 (1.09)**	+	+	0
32	*52.82 (3.71)*	-0.12 (0.53)	-0.12 (0.55)	-0.00 (0.47)	**0.64 (0.81)**	+	+	+
33	69.13 (2.63)	0.14 (0.34)	0.21 (0.54)	**0.52 (0.66)**	*0.27 (0.98)*	0	0	0
34	58.15 (4.74)	-0.07 (0.48)	0.06 (0.44)	*0.46 (0.55)*	**0.90 (1.10)**	+	+	0
35	58.61 (3.93)	*0.52 (1.35)*	0.48 (1.58)	-0.10 (1.16)	**1.64 (1.26)**	0	+	+
36	58.91 (3.40)	0.35 (0.59)	0.15 (0.79)	*0.78 (1.84)*	**1.73 (1.16)**	+	+	0
37	52.60 (4.12)	0.14 (0.42)	0.08 (0.26)	*0.27 (0.38)*	**0.67 (1.01)**	0	0	0
38	67.73 (3.67)	2.08 (2.28)	1.64 (0.93)	**3.37 (2.68)**	*3.34 (2.47)*	+	+	0
39	76.39 (2.22)	-0.25 (1.44)	0.29 (0.41)	*0.67 (0.72)*	**1.19 (0.92)**	0	+	+
40	70.58 (3.12)	-0.04 (0.28)	0.09 (0.38)	*1.06 (0.64)*	**1.95 (0.80)**	+	+	+

Table 5. Ranking accuracies of different predictor combination algorithms on the first 40 attributes of *AWA2* dataset. We repeated experiments 10 times with different training, validation, and test set splits. For baseline $\mathbf{f}^0$ (second column), Kendall's Tau correlations×100 (standard deviations in parentheses) are presented. For the remaining algorithms (third to sixth columns), the accuracy offsets from $\mathbf{f}^0$ are presented. The best and second best results are highlighted with **bold** and *italic* fonts, respectively. The results of statistical significance test based on a t–test with α=0.95 are highlighted in green (significantly positive) and orange (significantly negative). The last three columns show the results of statistical significance test of our algorithm with *GL*, *OPC*, and *MTL*, respectively (+/−: significantly positive/negative).

Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
1	77.86 (3.70)	*0.45 (0.92)*	0.12 (0.26)	0.15 (0.29)	**7.25 (2.73)**	+	+	+
2	83.79 (3.18)	0.05 (0.11)	-0.17 (0.41)	*0.33 (0.60)*	**6.22 (2.14)**	+	+	+
3	98.55 (0.65)	*0.02 (0.11)*	**0.04 (0.07)**	-0.01 (0.06)	0.02 (0.46)	0	0	0
4	88.21 (3.47)	0.03 (0.32)	*0.13 (0.29)*	0.05 (0.05)	**5.22 (2.27)**	+	+	+
5	88.53 (1.90)	*0.21 (0.49)*	0.12 (0.21)	0.04 (0.29)	**3.55 (2.17)**	+	+	+
6	*97.94 (1.07)*	-0.07 (0.17)	-0.02 (0.07)	-0.12 (0.29)	**0.69 (0.65)**	+	+	+
7	*99.22 (0.34)*	-0.05 (0.11)	-0.02 (0.06)	-0.00 (0.09)	**0.24 (0.23)**	+	+	+
8	82.30 (1.69)	-0.03 (0.12)	0.12 (0.39)	*0.13 (0.21)*	**4.32 (1.88)**	+	+	+
9	79.33 (4.37)	-0.01 (0.35)	*0.17 (0.42)*	0.04 (0.14)	**7.05 (1.65)**	+	+	+
10	98.58 (0.85)	*0.08 (0.31)*	0.03 (0.13)	0.01 (0.03)	**0.26 (0.40)**	0	0	0
11	97.44 (0.97)	-0.03 (0.23)	0.01 (0.10)	*0.06 (0.17)*	**0.90 (0.35)**	+	+	+
12	94.46 (1.91)	0.00 (0.25)	0.04 (0.31)	*0.47 (0.52)*	**1.13 (0.70)**	+	+	+
13	93.52 (1.05)	-0.14 (0.24)	*0.08 (0.31)*	0.05 (0.19)	**2.43 (0.68)**	+	+	+
14	94.50 (1.64)	0.04 (0.15)	0.33 (0.49)	*0.55 (0.43)*	**2.29 (0.64)**	+	+	+
15	95.04 (1.21)	0.00 (0.07)	*0.25 (0.40)*	0.13 (0.19)	**1.70 (0.68)**	+	+	+
16	85.91 (2.85)	-0.02 (0.03)	0.05 (0.23)	*1.39 (1.03)*	**5.88 (2.32)**	+	+	+
17	87.00 (2.60)	0.48 (0.64)	0.01 (0.36)	*1.64 (0.76)*	**4.33 (1.84)**	+	+	+
18	99.25 (0.55)	0.01 (0.03)	*0.08 (0.18)*	0.02 (0.10)	**0.18 (0.26)**	0	0	0
19	**99.75 (0.22)**	*-0.00 (0.01)*	-0.02 (0.04)	-0.02 (0.05)	-0.05 (0.31)	0	0	0
20	97.88 (0.99)	0.03 (0.15)	0.07 (0.26)	*0.27 (0.40)*	**0.79 (0.59)**	+	+	+
21	92.36 (2.33)	0.08 (0.13)	*0.30 (0.62)*	0.13 (0.37)	**2.44 (1.35)**	+	+	+
22	96.81 (1.03)	0.01 (0.06)	*0.20 (0.35)*	0.18 (0.18)	**1.40 (0.77)**	+	+	+
23	91.60 (2.67)	0.11 (0.36)	*0.15 (0.23)*	0.07 (0.14)	**3.57 (0.88)**	+	+	+
24	95.46 (1.41)	0.02 (0.30)	*0.06 (0.23)*	0.01 (0.14)	**1.43 (1.24)**	+	+	+
25	89.94 (2.52)	-0.02 (0.08)	0.07 (0.43)	*0.14 (0.45)*	**3.55 (1.68)**	+	+	+
26	84.78 (3.62)	0.03 (0.28)	*0.61 (1.16)*	0.23 (0.34)	**3.77 (1.15)**	+	+	+
27	90.67 (2.26)	-0.27 (1.30)	0.92 (1.09)	*1.09 (1.05)*	**3.91 (1.80)**	+	+	+
28	90.67 (1.67)	0.00 (0.03)	*0.05 (0.23)*	-0.00 (0.10)	**4.51 (1.59)**	+	+	+
29	95.73 (1.49)	**0.16 (0.30)**	-0.04 (0.16)	-0.11 (0.34)	*0.15 (0.80)*	0	0	0
30	94.97 (1.58)	0.04 (0.11)	*0.12 (0.42)*	0.06 (0.17)	**1.21 (1.10)**	+	+	+
31	94.39 (2.23)	*0.41 (0.45)*	0.40 (0.27)	0.41 (0.48)	**2.52 (0.99)**	+	+	+
32	96.92 (1.17)	-0.02 (0.13)	*0.14 (0.33)*	0.04 (0.11)	**0.86 (0.86)**	+	0	+
33	85.97 (4.63)	0.08 (0.21)	*0.16 (0.53)*	0.02 (0.20)	**6.68 (3.30)**	+	+	+
34	98.58 (0.83)	0.00 (0.01)	*0.07 (0.14)*	-0.07 (0.12)	**0.25 (0.60)**	0	0	0
35	98.69 (0.47)	0.03 (0.07)	*0.30 (0.34)*	0.26 (0.28)	**0.45 (0.45)**	+	0	0
36	*97.71 (0.82)*	-0.01 (0.14)	-0.02 (0.15)	-0.07 (0.13)	**0.44 (0.27)**	+	+	+
37	97.19 (1.48)	-0.04 (0.12)	0.22 (0.59)	*0.34 (0.42)*	**1.17 (1.18)**	+	+	+
38	95.21 (0.97)	*0.05 (0.10)*	-0.00 (0.16)	0.04 (0.17)	**1.95 (0.75)**	+	+	+
39	89.66 (1.83)	-0.02 (0.05)	0.13 (0.75)	*1.17 (0.94)*	**4.01 (1.53)**	+	+	+
40	95.33 (1.58)	-0.04 (0.17)	0.20 (0.26)	*0.64 (0.45)*	**2.60 (1.27)**	+	+	+

Table 6. Ranking accuracies of different predictor combination algorithms on the last 40 attributes of *AWA2* dataset. For each dataset, we repeated experiments 10 times with different training, validation, and test set splits. For baseline $\mathbf{f}^0$ (second column), Kendall's Tau correlations×100 (standard deviations in parentheses) are presented. For the remaining algorithms (third to sixth columns), the accuracy offsets from $\mathbf{f}^0$ are presented. The best and second best results are highlighted with **bold** and *italic* fonts, respectively. The results of statistical significance test based on a t–test with $\alpha=0.95$ are highlighted in green (significantly positive) and orange (significantly negative). The last three columns show the results of statistical significance test of our algorithm with *GL*, *OPC*, and *MTL*, respectively (+/−: significantly positive/negative).

Attr.	Baseline $\mathbf{f}^0$	*GL*	*OPC*	*MTL*	*NPC (ours)*	vs. *GL*	vs. *OPC*	vs. *MTL*
41	95.21 (1.26)	-0.06 (0.16)	0.01 (0.13)	*0.07 (0.19)*	**1.03 (0.60)**	+	+	+
42	84.32 (3.88)	*0.21 (0.49)*	-0.06 (0.17)	-0.07 (0.27)	**2.86 (1.72)**	+	+	+
43	95.78 (1.49)	0.06 (0.12)	-0.10 (0.19)	*0.13 (0.20)*	**2.57 (1.12)**	+	+	+
44	*98.22 (0.78)*	-0.05 (0.25)	-0.00 (0.04)	-0.00 (0.17)	**0.62 (0.50)**	+	+	+
45	88.71 (2.33)	0.07 (0.20)	0.42 (0.65)	*0.65 (0.83)*	**2.92 (2.09)**	+	+	+
46	83.63 (2.01)	0.01 (0.04)	0.43 (0.74)	*0.77 (0.41)*	**8.15 (1.08)**	+	+	+
47	89.57 (2.34)	*0.39 (0.59)*	0.16 (0.33)	0.13 (0.34)	**3.66 (2.15)**	+	+	+
48	91.92 (2.16)	0.18 (0.24)	0.51 (0.83)	*0.51 (0.43)*	**2.70 (1.81)**	+	+	+
49	89.56 (3.03)	0.05 (0.27)	-0.03 (0.25)	*0.34 (0.40)*	**3.65 (1.79)**	+	+	+
50	95.20 (1.24)	-0.01 (0.05)	-0.06 (0.25)	*0.11 (0.21)*	**2.23 (0.82)**	+	+	+
51	90.78 (2.72)	-0.34 (1.72)	*1.01 (1.61)*	0.91 (0.81)	**3.16 (1.46)**	+	+	+
52	94.00 (1.48)	-0.14 (0.19)	0.08 (0.70)	*0.28 (0.29)*	**3.11 (0.97)**	+	+	+
53	94.51 (1.71)	*0.10 (0.17)*	-0.00 (0.17)	-0.02 (0.18)	**1.25 (0.78)**	+	+	+
54	91.17 (1.73)	0.15 (0.39)	*0.22 (0.51)*	0.16 (0.47)	**4.19 (1.20)**	+	+	+
55	93.07 (1.87)	0.07 (0.29)	-0.00 (0.38)	*0.30 (0.33)*	**2.95 (0.86)**	+	+	+
56	88.30 (3.12)	0.29 (0.51)	0.57 (1.21)	*1.26 (1.36)*	**3.32 (2.47)**	+	+	+
57	92.34 (2.15)	*0.34 (0.61)*	0.03 (0.30)	0.05 (0.18)	**2.59 (0.92)**	+	+	+
58	96.10 (1.78)	0.16 (0.18)	0.11 (0.17)	*0.43 (0.75)*	**1.13 (1.02)**	+	+	+
59	92.12 (3.24)	*0.30 (0.42)*	0.16 (0.25)	0.23 (0.23)	**3.07 (1.47)**	+	+	+
60	89.37 (1.95)	0.12 (0.25)	*0.44 (0.75)*	0.14 (0.23)	**2.37 (1.45)**	+	+	+
61	92.43 (1.97)	-0.10 (0.22)	0.04 (0.27)	*0.17 (0.28)*	**2.37 (0.88)**	+	+	+
62	98.16 (1.08)	0.08 (0.27)	0.07 (0.24)	*0.18 (0.23)*	**0.39 (0.53)**	0	0	0
63	90.95 (3.52)	-0.04 (0.10)	*0.33 (0.51)*	0.15 (0.49)	**3.91 (2.55)**	+	+	+
64	92.31 (2.46)	-0.06 (0.13)	0.19 (0.24)	*0.39 (0.52)*	**2.80 (1.80)**	+	+	+
65	90.41 (3.04)	0.14 (0.30)	-0.01 (0.29)	*0.57 (0.49)*	**3.53 (1.25)**	+	+	+
66	89.81 (2.71)	*0.80 (0.89)*	0.58 (1.30)	0.00 (0.27)	**3.12 (1.38)**	+	+	+
67	94.72 (2.26)	-0.02 (0.13)	*0.11 (0.77)*	0.07 (0.15)	**2.38 (0.83)**	+	+	+
68	85.60 (2.43)	0.18 (0.83)	*0.47 (1.06)*	0.11 (0.14)	**6.41 (1.36)**	+	+	+
69	98.79 (0.41)	0.02 (0.05)	0.14 (0.27)	*0.14 (0.22)*	**0.50 (0.41)**	+	+	+
70	97.09 (1.19)	-0.04 (0.12)	0.14 (0.18)	*0.27 (0.35)*	**0.93 (0.83)**	+	+	+
71	99.02 (0.47)	0.01 (0.02)	*0.26 (0.20)*	0.16 (0.11)	**0.49 (0.32)**	+	+	+
72	96.69 (0.57)	-0.11 (0.29)	*0.23 (0.44)*	0.18 (0.19)	**1.79 (0.72)**	+	+	+
73	94.18 (1.34)	*0.09 (0.28)*	-0.00 (0.06)	0.00 (0.13)	**2.48 (0.98)**	+	+	+
74	83.62 (3.96)	-0.02 (0.41)	0.16 (0.84)	*0.56 (0.62)*	**5.06 (3.19)**	+	+	+
75	82.00 (3.99)	-0.06 (0.11)	0.54 (0.64)	*1.43 (1.62)*	**6.32 (2.62)**	+	+	+
76	83.13 (3.21)	0.08 (0.45)	0.23 (0.64)	*0.29 (0.65)*	**6.38 (2.76)**	+	+	+
77	85.16 (1.97)	0.09 (0.29)	0.08 (0.40)	*0.76 (0.67)*	**5.58 (2.22)**	+	+	+
78	85.83 (2.74)	0.14 (0.48)	-0.11 (0.50)	*0.43 (0.81)*	**4.48 (2.19)**	+	+	+
79	87.37 (2.77)	0.08 (0.34)	0.05 (0.29)	*0.50 (0.30)*	**5.25 (1.86)**	+	+	+
80	85.30 (2.93)	0.04 (0.21)	0.18 (0.53)	*0.44 (0.30)*	**6.05 (2.39)**	+	+	+