Predictor Combination at Test Time
— Supplemental Document —

Kwang In Kim
University of Bath

James Tompkin
Brown University

Christian Richardt
University of Bath

We present additional discussion on the interpretation of our algorithm (Sections 1 and 2), along with complete results that did not fit in the main paper due to limited space (Section 3). To make this supplement self-contained, we have reproduced some content from the main paper.

1. PAC-Bayesian interpretation

Our algorithm is constructed from a geometric intuition of manifold denoising. As an alternative, we present a second interpretation based on the assumption that our references are constructed explicitly as GP predictive distributions, i.e., from a PAC-Bayesian perspective. We follow the PAC-Bayesian analysis by Pentina et al. [10], derived for linear support vector machine (SVM) cases by casting the linear SVM parameter vector into a Gaussian random vector. In Section 3 of the main paper, we extended it to a sample-based, finite-dimensional projection of infinite dimensional GPs.

For a given bounded loss function $C(\cdot,\cdot)\in[0,1]$, the expected error rate R_C and the empirical error rate $\hat{R}_C$ of a deterministic predictor g are defined as [9, 12]:

$$R_C(g)=\int C(y,g(\mathbf{x}))p_{\mathcal{X}\times\mathbb{R}}(\mathbf{x},y)d(\mathbf{x},y), \tag{1}$$

$$\hat{R}_C(g)=\frac{1}{l}\sum_{j=1}^{l}C(y_j,g(\mathbf{x}_j)), \tag{2}$$

where we assume that g is trained on l data points plus labels.

The PAC-Bayesian analysis provides a probabilistic statement on the upper bound of the expected error rate R_C, based on the empirical error rate $\hat{R}_C$ and the difference between the prior GP h^i and the posterior GP f of the function g [9, 12]: For any GPs h^i and f over g, with probability at least $1-\delta$,

$$R_C(g)\leq\hat{R}_C(g)+\sqrt{\frac{1}{2l}\left[\mathrm{KL}(f\,|\,h^i)+\ln\left(\frac{2\sqrt{l}}{\delta}\right)\right]}. \tag{3}$$

In our experiments, this bound was dominated by the KL-divergence when g was constructed as the mean function of a GP predictive distribution f.[1] In this case, our denoising algorithm that uses the single (closest) reference h^i in H minimizes this generalization bound. In general, multiple references can be relevant to f, and therefore we denoise f by minimizing the divergences from a set of (close) $\{h^i\}$, weighted by the confidences $(1-\mathrm{KL}(f\,|\,h^i))$ associated with each h^i.

[1] Originally, McAllester's PAC-Bayesian bound is applicable to bounded loss l [9, 12]. In our experiments, we adopted the approach of using the standard squared loss l' for training while for the evaluation of the bound, l' is replaced by a bounded loss l.

2. Out-of-sample extension

Our experiments focused on refining the evaluation $\mathbf{f}$ of the predictor of interest on a fixed dataset U. However, based on the GP assumption, recovering the explicit functional form of f is possible. Three scenarios exist:

1. When $\mathbf{f}$ is constructed explicitly as a GP predictor (with covariance function k and noise parameter $\delta_{\mathcal{X}}$), the mean function f can be obtained as a combination of kernel expansions:

$$f(\mathbf{x})=\sum_{i=1}^{u}\alpha_i k^f(\mathbf{x}_i,\mathbf{x}), \tag{4}$$

   where $\{\alpha_i\}$ is obtained as the minimizer of the energy functional $\mathcal{E}_{\mathrm{GP}}$ with the labeled training data points augmented with $(U,\mathbf{f}^*)$ (Equation 8).
2. When the underlying model of $\mathbf{f}$ is not provided, but the reference set H is given as GPs, k^f and $\delta^f_{\mathcal{X}}$ can be constructed based on the corresponding covariance functions $\{k^i\}$ and noise parameters $\{\delta^i_{\mathcal{X}}\}$ of H, e.g., k^f can be constructed as a convex combination of $\{k^i\}$:

$$k^f=\sum_{h^i\in H}\overline{w}(h^i)k^i, \tag{5}$$

$$\overline{w}(h^i)=\frac{w(\mathrm{KL}(f\,|\,h^i),\sigma_f^2)}{\sum_{h^j\in H}w(\mathrm{KL}(f\,|\,h^j),\sigma_f^2)}, \tag{6}$$

$$w(x,c)=\exp\left(-\frac{x^2}{c}\right). \tag{7}$$

3. When the explicit representations of $\mathbf{f}$ and H are all unknown, (the hyper-parameter of) the covariance function k^f and the noise $\delta^f_{\mathcal{X}}$ can be estimated based on additional labeled data points, or by performing cross-validation on S.

3. Experiments

As our approach allows combination of predictors of unknown parametric form, existing approaches which require known parametric forms are not applicable. To enable experimental comparison with multi-task or transfer learning, we thus devise a scenario in which existing algorithms are provided with explicit parametric forms while our algorithm is not. To facilitate the objective assessment of our algorithm in this case, we include the training process of f (based on S) in the evaluation of the algorithm.

Our setup. Our algorithm starts with an initial target predictor f and the set of reference predictors H, and produces a denoised target predictor f^*. For each problem, the initial estimate f is obtained as a GP regressor with standard Gaussian covariance kernel $k(\mathbf{x}_1,\mathbf{x}_2)=\exp(-\|\mathbf{x}_1-\mathbf{x}_2\|/\sigma_{\mathcal{X}}^2)$ with scale parameter $\sigma_{\mathcal{X}}$, trained based on a labeled data set $S=\{(\mathbf{x}_{u+1},y_{u+1}),\ldots,(\mathbf{x}_{u+l},y_{u+l})\}$. The mean function f is obtained as a minimizer of the energy functional

$$\mathcal{E}_{\mathrm{GP}}(f)=\left(\sum_{(\mathbf{x},y)\in S}(f(\mathbf{x})-y)^2\right)+\delta_{\mathcal{X}}\|f\|_k^2, \qquad (8)$$

where $\|f\|_k$ is the reproducing kernel Hilbert space (RKHS) norm of f corresponding to the covariance kernel k [12], and $\delta_{\mathcal{X}}$ represents the noise model. Also, we present the predictor combination results, where we use simple linear regressors instead of Gaussian process regressors as the baseline. This shows very similar improvement of our algorithm over existing approaches (Figure 6). As our denoising algorithm uses the unlabeled dataset U, the entire training process, including hyper-parameter tuning (using the labeled dataset S), becomes semi-supervised.

Baseline setup. We adapt Evgeniou and Pontil's graph Laplacian-based algorithm [5] and Pentina et al.'s curriculum learning algorithm (*CL*) [10], plus baseline independent GP predictions (*Ind*). Note that in our predictor combination problem setting, the first approach [5] is equivalent to transfer learning [1, 7, 15], while Pentina et al.'s algorithm corresponds to choosing the best reference in H that minimizes the generalization error bound [10]. We implemented two different versions of Evgeniou and Pontil's algorithm: the first version (GL_1) uses the graph Laplacian ($\mathbf{L}$) with the uniform weight matrix $\mathbf{W}=\mathbf{1}\mathbf{1}^\top$,[2] while the second version (GL_2) estimates the relevance of the reference predictors $\{h^i\}$ to the predictor f to be refined based on Euclidean distance between the corresponding parameter vectors:

$$\mathbf{W}_{if}=w(\|\mathbf{w}_i-\mathbf{w}_f\|,\sigma_{\mathrm{w}}^2), \qquad (9)$$

using the hyper-parameter σ_{w} and for GP (mean) predictors ($h^i(\mathbf{x})=\Phi(\mathbf{x})^\top\mathbf{w}_i$, $f(\mathbf{x})=\Phi(\mathbf{x})^\top\mathbf{w}_f$).

[2] $\mathbf{L}=\mathbf{I}-\mathbf{D}^{-1}\mathbf{W}$ with $\mathbf{D}$ being a diagonal matrix of the row sums of $\mathbf{W}$: $\mathbf{D}_{ii}=\sum_{j=1}\mathbf{W}_{ij}$.

All three baseline algorithms (*CL*, GL_1 and GL_2) can be extended to the semi-supervised learning setting we use by adopting a domain graph Laplacian $L_{\mathcal{X}}$ as a regularizer:

$$\mathcal{R}(\mathbf{f})=\lambda_3\mathbf{f}^\top L_{\mathcal{X}}\mathbf{f}, \qquad (10)$$

where $\mathbf{f}=\{f(\mathbf{x}_1),\ldots,f(\mathbf{x}_u)\}$ and λ_3 is a hyper-parameter. The resulting algorithms are equivalent to Luo et al. [8] and Wang et al. [17] in the predictor combination setting. For each dataset, we only report the performance of the best semi-supervised version (in terms of average test error) as "*SSL*", as all semi-supervised versions produce very similar results to their underlying supervised version. The best semi-supervised extensions are constructed from *CL* (*MOCAP*, *SARCOS*), GL_1 (*School*) and GL_2 (*CAESAR*), respectively.

We compute results on four regression datasets: *MOCAP*, *CAESAR*, *SARCOS*, and *School*. We report performance for all algorithms with varying numbers of labeled training data points. We repeat each experiment 10 times with different training and test set splits, and average the results. We also demonstrate non-parametric predictor combination for facial landmark detection (*Landmarks* dataset), where traditional parametric combination algorithms are futile to apply.

4. MOCAP dataset

Human body poses are captured with an optical marker-based motion capture system [2]. Each of the 50,000 data points describes the 3D location of 62 skeletal body joint locations (i.e., $62\times3=186$ output dimensions). The task is to estimate these body joint locations from the 3D locations of five *end effectors* (left/right hand, left/right foot, and head), i.e., a $5\times3=15$ dimensional mid-level representation as inputs. We removed redundant variables from the original 186-dimensional space, leaving an 87-dimensional output data representation. We randomly sample eight of these as target predictors, and for each use the remaining 86 predictors as references.

For all combination algorithms, we adopt the explicit GP model assumption for the reference predictors H, i.e., we assume that all reference predictors are explicitly constructed as GP predictive distributions. Due to the large size of the *MOCAP* dataset, training the full GP reference models is infeasible, so we adopt Snelson and Ghahramani's sparse GP approximation [14] using 1000 *inducing data points*. This setting facilitates more direct (model-based) comparisons with the baselines GL_1, GL_2 and *CL*. In this setting, our algorithm further benefits from the available predictive variances, which improve the estimation of the KL-divergences (Equation 16 of the main paper): using GP predictive variances reduced the average error rate by 11.34% from the model-free case of using the unit covariance $\delta(\cdot,\cdot)$. However, this reduction is achieved at the expense of making an explicit model assumption, which may restrict the application domain of our algorithm (similar to existing algorithms).

Results. All eight target predictors show improvement (Figure 1). GL_1 did not show noticeable improvement over

Ind, indicating that not all variables are relevant. GL_2, *CL*, and *SSL* show noticeable improvements, but the improvements achieved by our algorithm are much more significant. Figure 6 shows additional experimental results, where linear regressors are used as predictors. Even though linear regressors led to higher average baseline error rates, our test-time combination algorithm significantly improved upon the baselines.

5. CAESAR dataset

This dataset contains 4,258 3D scans of human bodies along with 6 ground-truth measurements: arm length, age, sitting height, weight, shoulder breadth, and foot length [11, 13]. Each body scan is represented as a 20-dimensional feature vector by fitting a statistical body model [11]. Our goal is to refine the initial *target predictor* f of each body measurement by using the remaining 5 measurements as reference predictors H. This constitutes 6 different predictor combination problems.

For our algorithm, each of the 5 observed measurements is used directly as a reference predictor. The corresponding GPs are constructed by using the unit covariance $\delta(\cdot,\cdot)$ (Section 3.1 of the main paper). This corresponds to the simplest and least restrictive application case, where no model assumption on H is imposed. However, this setting is *not applicable* to baselines GL_1, GL_2 and *CL* as they require explicit representations of the reference predictors H. Therefore, for them, the reference predictors $H=\{h^i\}$ are explicitly constructed as GP regressors trained on the observed reference variables.

We also present a comparison with Bonilla et al.'s non-parametric Gaussian process-based multi-task learning (MTL) algorithm (*MTGP*) [4] in Figure 3. We include this comparison as a curiosity to interested readers, because our problem of refining the predictors from a fixed set of references is not translatable to the classical MTL problem. As an MTL method, this algorithm requires training all predictors simultaneously, i.e., requires access to the training process of individual predictors. This is not possible in our application scenario where the reference predictor may be provided by a precompiled software library, or even be a human predictor.

As such, the results of the MTGP comparison are not comparable to the results of our *adaptations* of MTL to our test time combination setting (GL_1, GL_2 and *CL*, Figure 2). Instead, the comparison is performed in the standard MTL setting, for which we use the code shared by the authors. This algorithm requires tuning the rank hyper-parameter, which we performed by picking the best test error rate. The other hyper-parameters including the kernel parameters are automatically tuned by their algorithm.

Results. In realistic applications, not all predictors are relevant. For example, age is not strongly correlated with body length measurements. However, two predictors (arm and foot length) benefit significantly from the combinations obtained by our algorithm (Figure 2). Predictor combinations for the other variables are on par with baseline algorithm *Ind*. The other baseline algorithms GL_1, GL_2, *CL* show no noticeable improvement over *Ind* for any combination.

Table 1. Pairwise 1–KL-divergence values for the *SARCOS* dataset. The target variables 2, 3, 4 and 7 have small KL-divergences leading to mutual improvement by combination.

	1	2	3	4	5	6	7
1	0.00	0.00	0.01	0.26	0.03	0.00	0.17
2	0.00	0.00	0.69	0.33	0.09	0.01	0.41
3	0.01	0.69	0.00	0.47	0.31	0.03	0.54
4	0.26	0.33	0.47	0.00	0.05	0.00	0.93
5	0.03	0.09	0.31	0.05	0.00	0.05	0.09
6	0.00	0.01	0.03	0.00	0.05	0.00	0.01
7	0.17	0.41	0.54	0.93	0.09	0.01	0.00

6. SARCOS dataset

This kinematics dataset contains 44,484 data points collected from a robot arm. The input consists of 7 joint positions, 7 velocities and 7 accelerations, and the output consists of 7 torques [16]. The experimental setting is the same as *CAESAR*: our goal is to refine the predictor of each output attribute given the remaining 6 attributes as references. For our algorithm, the reference predictors are obtained in the same way as *CAESAR*. For GL_1, GL_2 and *CL*, GP regressors are constructed: the reference predictors are constructed based on sparse GP approximation with 1000 inducing data points.

Results. Four out of seven predictors significantly benefit from our predictor combinations (Figure 4). This is in accordance with the measured (inverse) KL-divergences shown in Table 1: target variables 2, 3, 4 and 7 have particularly small KL-divergences (large $1-$KL) with each other, which indicates their mutual relevance. The other algorithms show no significant improvement compared to *Ind*.

7. Landmarks dataset

The task is to detect 6 facial landmarks (the corners of both eyes and the mouth) from 550 face images selected from the BioID Face Database [6]. Three sliding-window-based non-linear SVM detectors are trained (eye inner and outer corners, and mouth corners, exploiting facial symmetry), with detections made at the highest responses. We evaluate the performance of detectors with varying number of training images ($\{10,30,50,100\}$, with the remaining images used for testing). For each detector, 16 training sub-images (patches) are extracted per image: 4 *positive* patches sampled randomly from a 3×3 window centered at the annotated ground-truth position, plus 12 *negative* patches. The size of the sub-images are determined per landmark for *Ind* based on cross-validation. We apply our algorithm to detected (x,y)-coordinate values, representing 12 attributes. Traditional multi-task learning (MTL) cannot be applied in this setting, as it assumes a shared parametric form for the predictors. Hence, we apply MTL at the level of the SVM detectors.

Table 2. Mean squared error (standard deviation in parentheses) on the *School* dataset. All combination approaches improve on independent prediction (*Ind*), although our approach fails to outperform the baselines as the number of unlabeled data points (maximum 251) is too small to reliably estimate KL-divergences. *SSL* is the semi-supervised version of GL_1.

Ind	GL_1	GL_2	*CL*	*SSL*	Ours
11.86 (2.03)	10.80 (1.82)	11.07 (1.95)	11.18 (1.86)	10.81 (1.78)	11.24 (1.86)

Results. Figure 5 summarizes the results. For any number of training images, we can see that traditional MTL does not help. Enforcing similarity of 'eye-corner' detector and 'mouth' detector actually degrades the performance over individual detectors, as these are not anatomically connected. Our algorithm better exploits predictor dependencies through the detected spatial coordinates.

8. School dataset

This dataset consists of examination records of 15,362 students in 139 schools from the Inner London Education Authority [3]. The goal is to predict the exam scores of the students based on 27 input features, such as the year of the exam and gender. Our goal is to estimate the exam scores of each school based on the predictors of the remaining 138 schools as references. This constitutes 139 different combinations of target and reference predictors. We perform experiments on each set trained based on 20 labeled data points, and report the average error rate. Similarly to *MOCAP*, for all combination algorithms, the reference predictors are explicitly constructed as (full) GP predictors.

Results. All four algorithms significantly improved upon *Ind* (Table 2). However, our method shows the least improvement. For this dataset, all tasks are strongly related. All target and reference variables correspond to a single attribute—exam scores—but are sampled from different schools. Thus, only the data sampling distributions are different. This is in contrast to the three other datasets, where each output variable has a different characteristic.

For this dataset, all combination algorithms improve upon independent predictions (*Ind*): Using all parametric references uniformly (GL_1) led to the best results, followed by GL_2, *CL*, and our algorithm. Our algorithm suffered from the lack of data points: the maximum number u of available data points U for each task is 251, with around half of the tasks having less than 100 data points. This demonstrates a limitation of our approach in that data-driven estimation of KL-divergences can be unreliable versus explicit parametric form modeling. However, even in this case, our result still improves over independent predictions without providing explicit parametric forms.

References

[1] Y. Aytar and A. Zisserman. Tabula rasa: model transfer for object category detection. In *ICCV*, 2011.

[2] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*, pages 1092–1099, 2011.

[3] B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *JMLR*, 4, 2003.

[4] E. V. Bonilla, K. M. Chai, and C. Williams. Multi-task Gaussian process prediction. In *NIPS*, pages 153–160, 2008.

[5] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *JMLR*, 6:615–637, 2005.

[6] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the Hausdorff distance. In *Audio- and Video-Based Biometric Person Authentication*, pages 90–95, 2001.

[7] A. Kovashka and K. Grauman. Attribute adaptation for personalized image search. In *ICCV*, pages 3432–3439, 2013.

[8] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank. Manifold regularized multitask learning for semi-supervised multilabel image classification. *IEEE TIP*, 22(2):523–536, 2013.

[9] D. A. McAllester. PAC-Bayesian model averaging. In *COLT*, pages 164–170, 1999.

[10] A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. In *CVPR*, pages 5492–5500, 2015.

[11] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele. Building statistical shape spaces for 3D human modeling. *Pattern Recognition*, 2017.

[12] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[13] K. Robinette, H. Daanen, and E. Paquet. The CAESAR project: a 3-D surface anthropometry survey. In *3-D Digital Imaging and Modeling*, 1999.

[14] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *NIPS*, 2006.

[15] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: learning categories from few examples with multi model knowledge transfer. In *CVPR*, pages 3081–3088, 2010.

[16] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional space. In *ICML*, pages 1079–1086, 2000.

[17] F. Wang, X. Wang, and T. Li. Semi-supervised multi-task learning with task regularizations. In *ICDM*, pages 562–568, 2009.

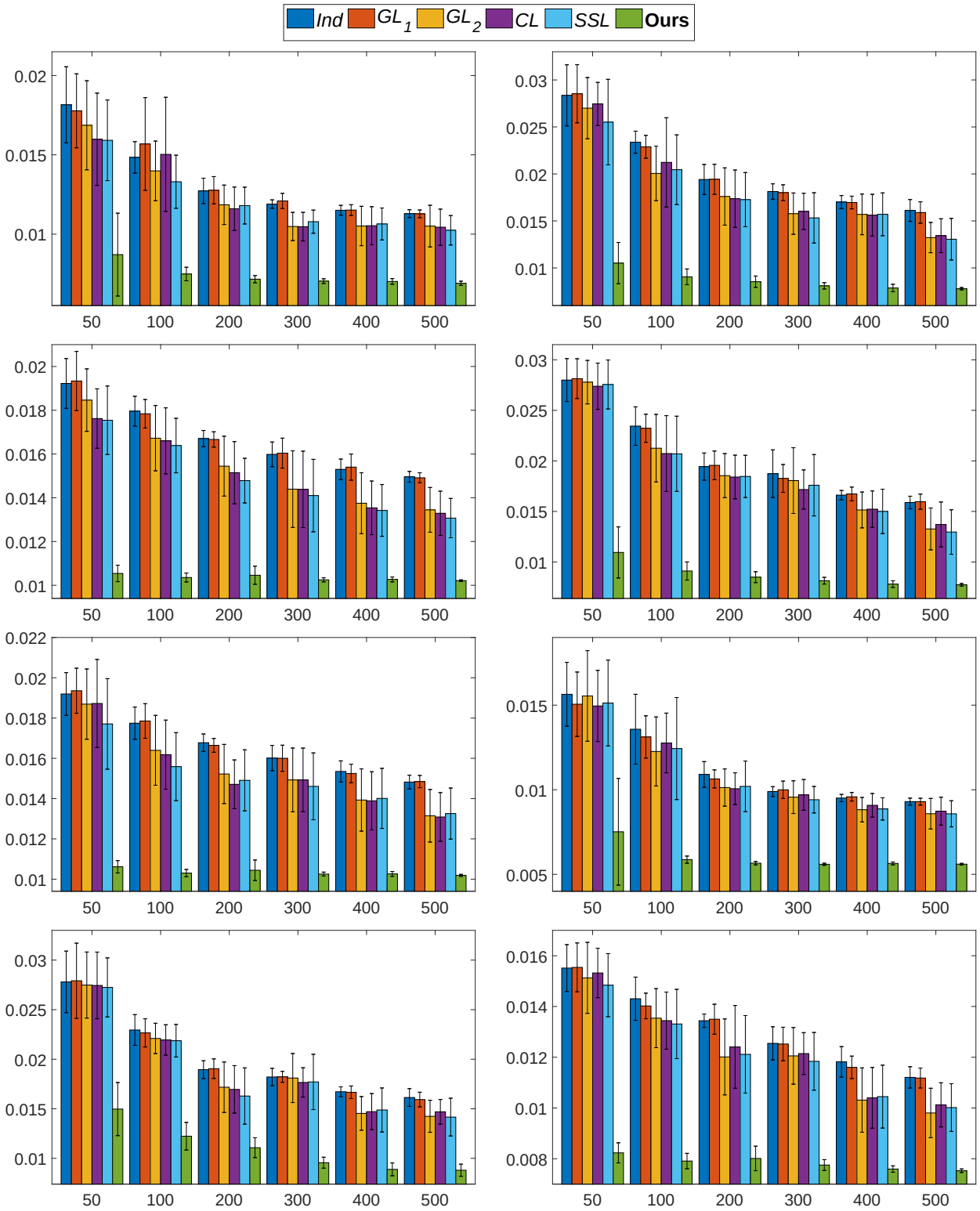

Figure 1. Mean squared error in parameter units from regression on the *MOCAP* dataset (lower is better; error bars are standard deviations). Each plot corresponds to the residual error of learning a target predictor f^i given the remaining reference predictors. The horizontal axis shows the number l of labels used. We compare to: (*Ind*) baseline independent predictions; (GL_1 and GL_2) adaptations of Evgeniou and Pontil [5]; (*CL*) curriculum learning [10]; (*SSL*) semi-supervised extension of *CL*.

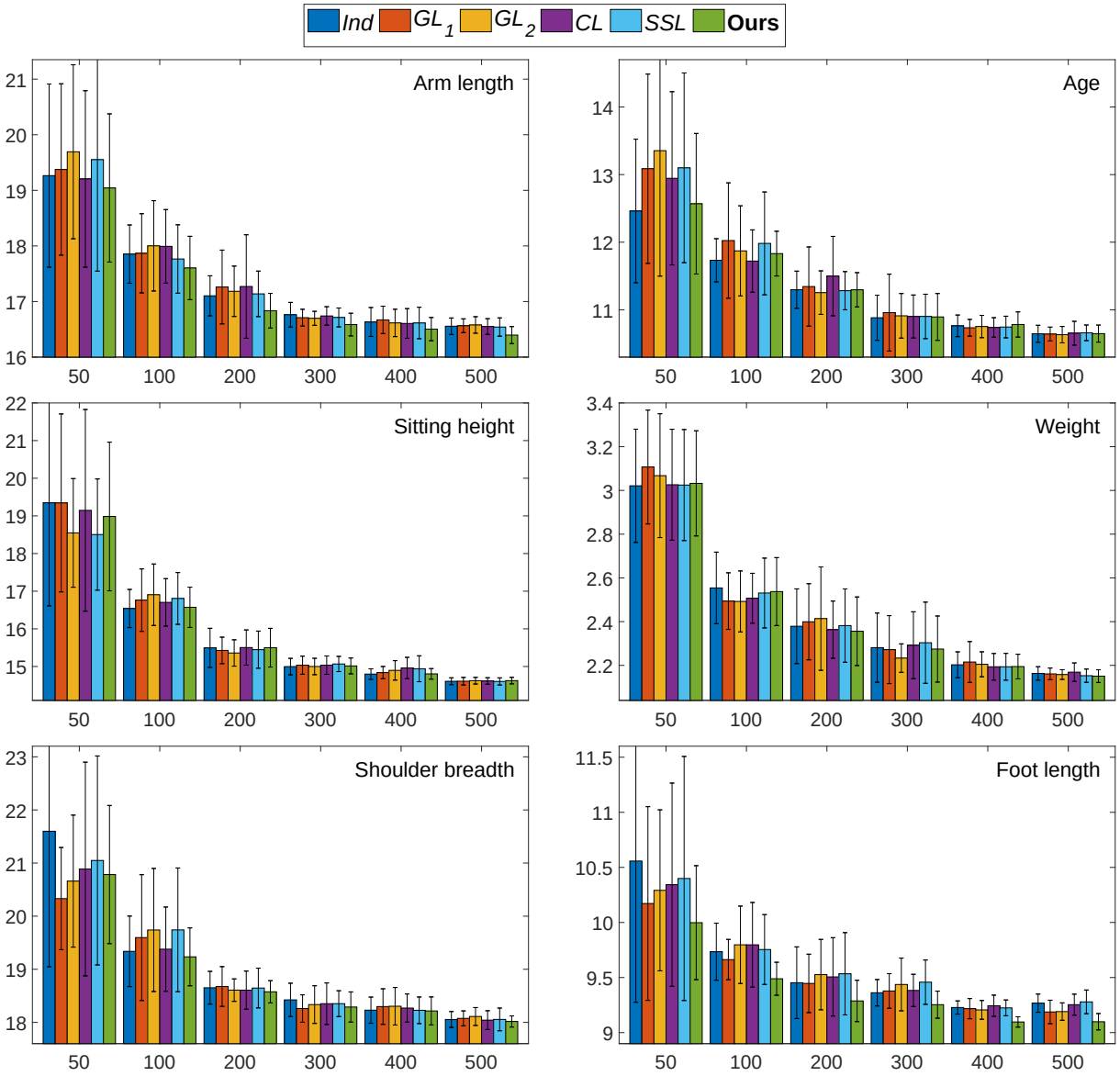

Figure 2. Error rates (lower is better; error bars are standard deviations) for the *CAESAR* dataset. Each plot corresponds to the residual error of learning a target predictor f^i given the remaining reference predictors. The horizontal axis shows the number l of labels used. The prediction results for Arm length and Foot length are shown in the main paper. For the remaining target attributes, no combination algorithm shows a large improvement from *Ind*. *SSL* is the semi-supervised extension of GL_2.

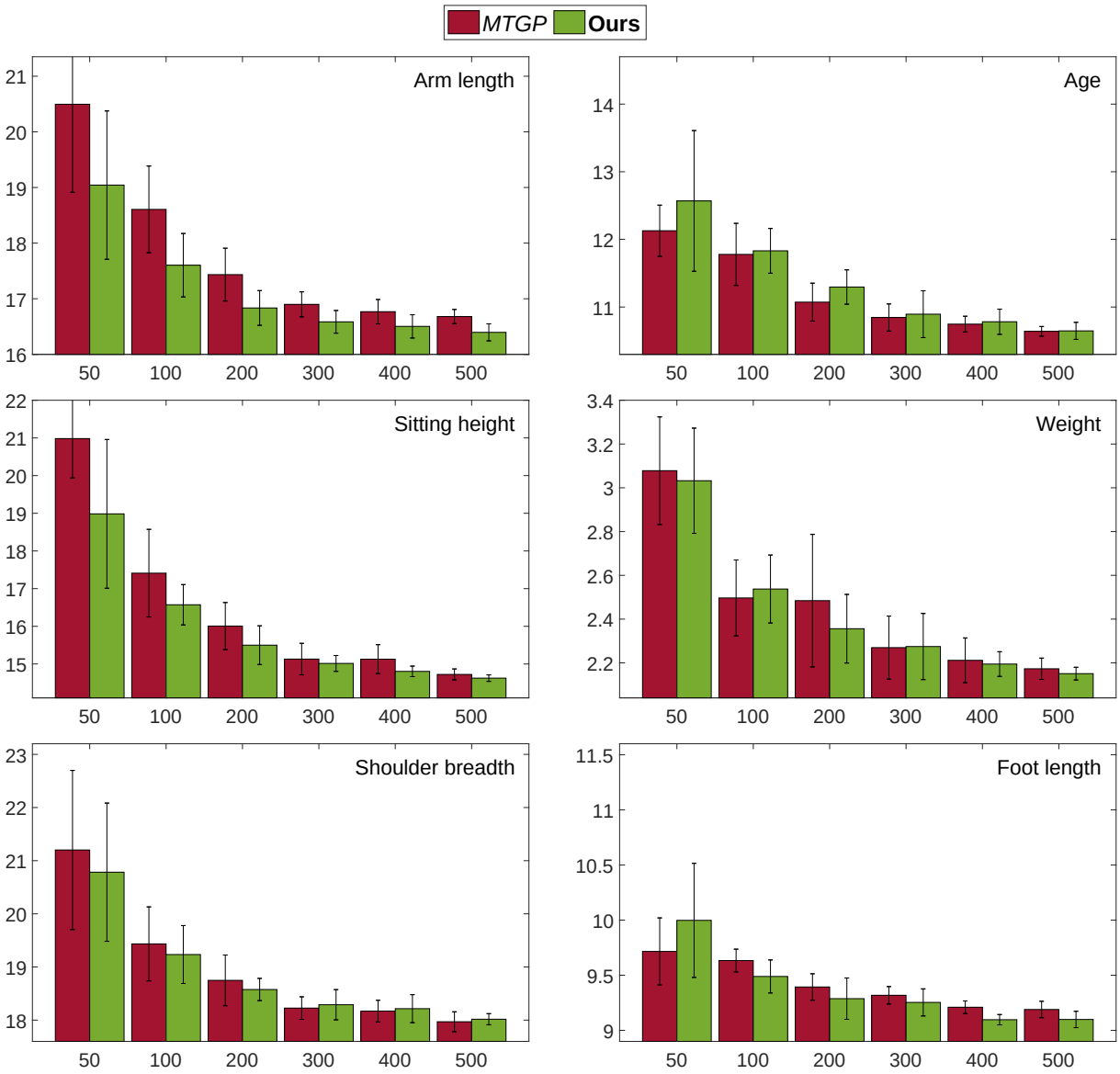

Figure 3. Error rates for the *CAESAR* dataset. For MTGP, the plots show to the residual errors of target attributes which are all learned *simultaneously*. For our method, each plot corresponds to the residual error of learning a target predictor f^i given the remaining reference predictors.

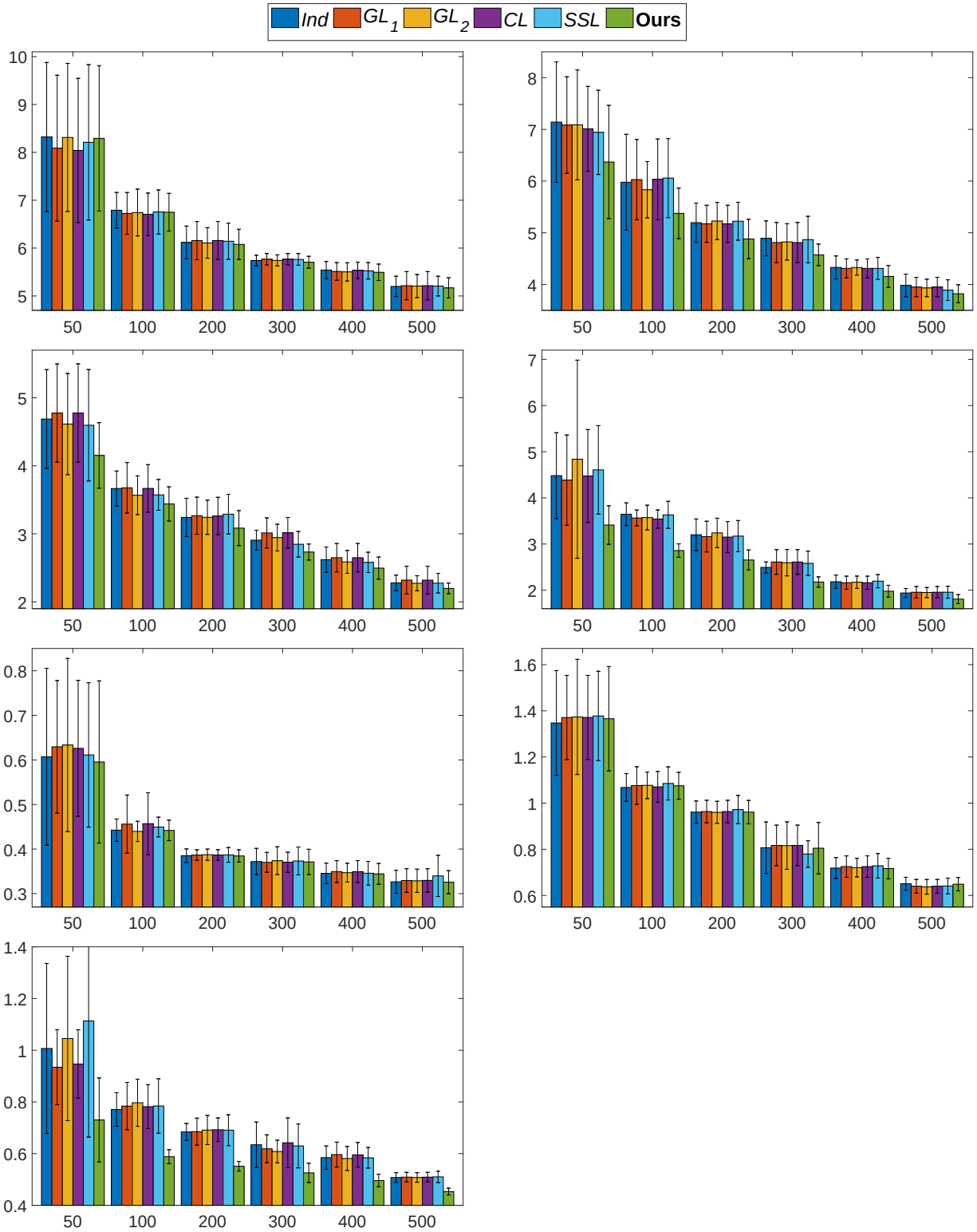

Figure 4. Mean squared error in (lower is better; error bars are standard deviations) for *SARCOS* dataset. Each plot corresponds to the residual error of learning a target predictor f^i given the remaining reference predictors. The horizontal axis shows the number l of labels used. *SSL* is the semi-supervised extension of *CL*.

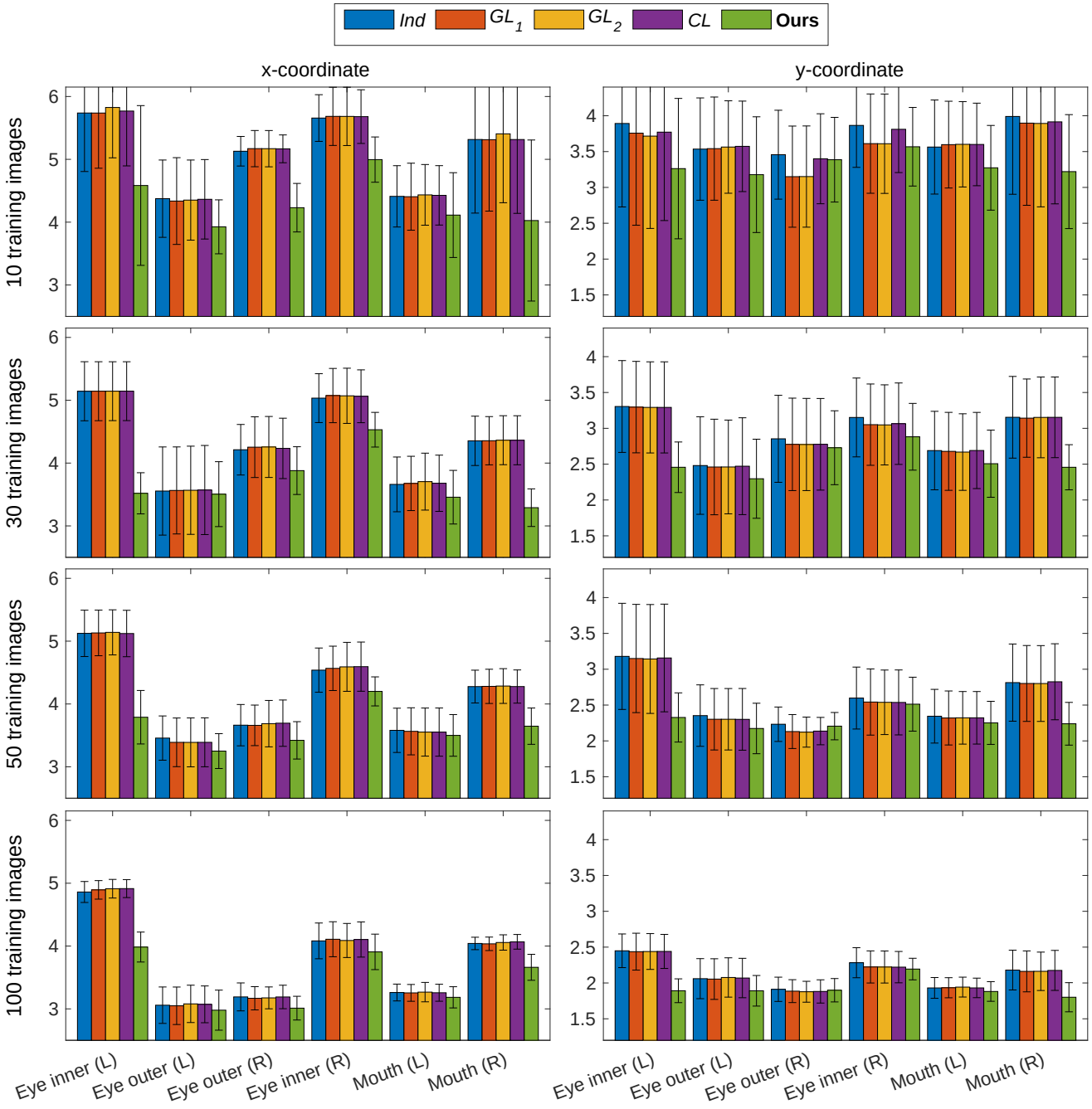

Figure 5. Facial landmark detection error, in pixels vs. annotated ground truth with varying number of traning images (lower is better; error bars are standard deviations). Horizontal axis: indices of six 2D facial landmarks (left: x-coordinates; right: y-coordinates). We compare to: (*Ind*) baseline independent predictions; (GL_1 and GL_2) adaptations of Evgeniou and Pontil [5]; (*CL*) curriculum learning [10].

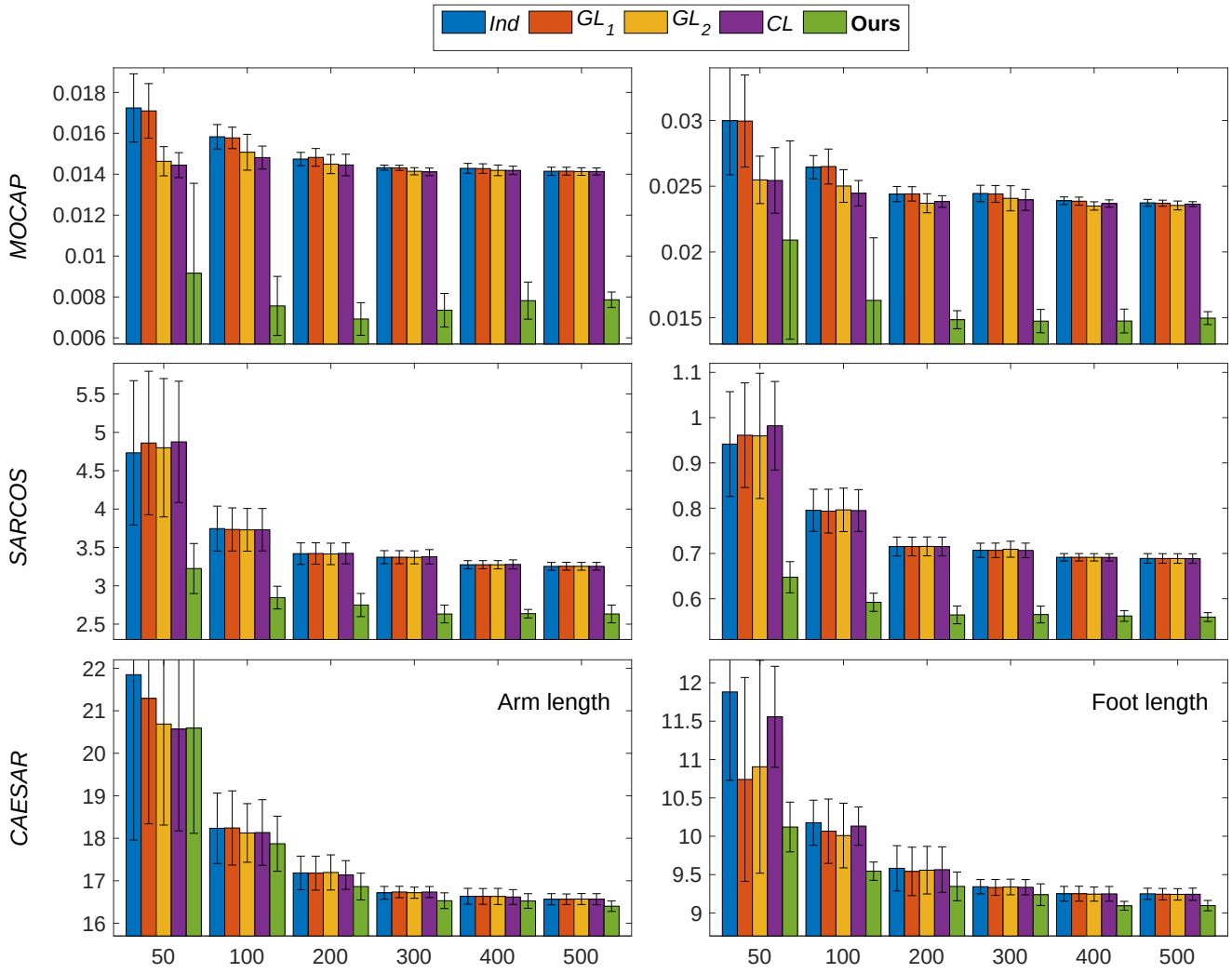

Figure 6. Prediction results with linear regressors as baselines: mean squared error in parameter units (lower is better; error bars are standard deviations). For each dataset, we show two predictor combinations that correspond to the results shown in the first three rows of Figure 1 in the main paper. Our algorithm significantly benefits from the combination even when the baseline predictors are linear.