Covariate Shift Adaptation for Discriminative 3D Pose Estimation

Makoto Yamada, *Member, IEEE*, Leonid Sigal, *Member, IEEE* and Michalis Raptis,

Abstract—Discriminative, or (structured) prediction, methods have proved effective for variety of problems in computer vision; a notable example is 3D monocular pose estimation. All methods to date, however, relied on an assumption that training (source) and test (target) data come from the same underlying joint distribution. In many real cases, including standard datasets, this assumption is flawed. In presence of training set bias, the learning results in a biased model whose performance degrades on the (target) test set. Under the assumption of covariate shift we propose an unsupervised domain adaptation approach to address this problem. The approach takes the form of training instance re-weighting, where the weights are assigned based on the ratio of training and test marginals evaluated at the samples. Learning with the resulting *weighted* training samples, alleviates the bias in the learned models. We show the efficacy of our approach by proposing weighted variants of Kernel Regression (KR) and Twin Gaussian Processes (TGP). We show that our weighted variants outperform their un-weighted counterparts and improve on the state-of-the-art performance in the public (HUMANEVA) dataset.

Index Terms—3D pose estimation, covariate shift adaptation, importance weight estimation, twin Gaussian processes.

✦

1 INTRODUCTION

Many problems in computer vision can be expressed in the form of (structured) predictions of real-valued multivariate output, $y \in \mathbb{R}^{d_y}$, from a high-dimensional multivariate input, $x \in \mathbb{R}^{d_x}$. In this paper, we focus on such models in the context of articulated 3D pose estimation.

Articulated 3D pose estimation, particularly from monocular images and/or video, is a challenging problem due to variability in person appearance, pose, body shape, lighting, and motion. Despite these challenges, discriminative methods, have proved to be effective in recovering the 3D pose [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17] in variety of scenarios. In these methods the goal is to learn a direct (and often multi-modal) mapping, $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$, from image features (e.g., bag-of-words of HoG or SIFT descriptors) $x \in \mathbb{R}^{d_x}$ to 3D poses $y \in \mathbb{R}^{d_y}$, typically expressed as joint positions or angles. Probabilistic formulations do so by learning the conditional distribution $p(y|x)$ based on the training dataset of n_{tr} image-pose pairs – $\{(x_i^{tr}, y_i^{tr})\}_{i=1}^{n_{tr}}$ (assumed to be independent and identically distributed (i.i.d.) samples from the underlying joint density $p_{tr}(x, y)$). A number of methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17] of this form have been proposed over the last decade that explore a gamut of models,

image features and learning architectures. However, in *all* cases it has been assumed that the training and test distributions are one and the same, i.e., $p_{tr}(x, y) = p_{te}(x, y)$, and hence the model learned using the training feature-pose pairs can be directly applied to the test image features, x^{te}, to infer the output 3D pose y^{te}.

The problem of *dataset bias* is starting to emerge as very prominent issue in object categorization [18], [19], [20], [21], [22], where even large datasets (e.g., LabelMe or ImageNet) have shown to exhibit significant (and often unexpected) biases [22] in the form of lighting, object appearance and viewpoint to name a few. We argue that similar issues exist in 3D pose estimation and need to be addressed if one is to build a system that works outside of well calibrated laboratory setups and datasets. The issues of dataset bias and overfitting to the training set, in 3D pose estimation, are evident from poor generalization that one often sees when applying such models to novel data. In addition, we argue that *within* dataset bias is, at least in certain cases, as prevalent as the *between* dataset bias. While in dataset creation an effort is typically made to make the training and test sets as similar as possible, this is difficult to achieve precisely. For example, Urtasun and Darrell in [16] show that performance decreases dramatically (to as low as 25% of the baseline) when training and test sequences are disjoint[1]. Note that this is despite the fact that, even in the disjoint case, training and test sequences were captured by the same static cameras and with no appreciable difference in subject appearance and

- M. Yamada is with the NTT Communication Science Laboratories in Japan.
  E-mail: myamada0321@gmail.com
- L. Sigal and M. Raptis are with the Disney Research Pittsburgh in USA.
  E-mail: {lsigal,mraptis}@disneyresearch.com

---

1. The baseline sampled training/test sets on per-frame bases from the full HUMANEVA-I [23] dataset.

lighting. Similar degradation of performance is often observed when a subject is not included in the training set, or when training data comes from multiple subjects (and/or motions) and at the test time only a single subject (and/or motion) is observed [5].

Unfortunately, the domain adaptation approaches proposed in [18], [19], [20], [21], [24], [25], [26] are not adequate for addressing the bias in this case. First, they typically assume categorical classification, as opposed to multi-valued (structured) predictions. Second, and more importantly, they are supervised and assume existence of one or more labeled instances from the test set to allow the transfer learning to fine tune the source model to a target test set. In many scenarios, such as, 3D pose estimation, obtaining 3D pose for a test image is infeasible. To this end, we formulate a novel training instance re-weighting mechanism for addressing the bias in (structured) prediction problems under the assumption of a *covariate shift* [27] in an unsupervised manner; where we assume that $p(\boldsymbol{y}|\boldsymbol{x}) = p_{\mathrm{tr}}(\boldsymbol{y}|\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{y}|\boldsymbol{x})$, but the marginals are different ($p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$).

Contributions: The key contributions of this work is to shed some light on the potential issues of dataset bias in the structured prediction problems, mainly, 3D pose estimation, and to propose a simple, yet effective, solution for handling such bias through training instance re-weighting in a covariate shift adaptation formulation. We illustrate the efficacy of our approach by proposing weighted variants of Kernel Regression and Twin Gaussian Processes and showing that they outperform their non-weighted counterparts in various setups and with different image features. As a consequence we achieve state-of-the-art performance on HUMANEVA-I dataset. The proposed training instance re-weighting, however, is general and is amenable to most popular formulations (e.g., Linear Regression, Mixture of Experts, GPLVM, Kernel Information Embeddings), as well as to other (structured) prediction problems in computer vision.

Preliminary version of this work appeared in [28]; here we provide a more comprehensive derivation and discussion of Importance Weighted Twin Gaussian Processes (IWTGP), as well as include a more detailed experimental section that addresses model sensitivity to parameters.

2 RELATED WORK

Discriminative models are popular in vision for various tasks, including 3D human pose [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], human shape [29], hand pose [30], [31] and face pose [32] estimation. The focus of this paper is on 3D pose estimation which we discuss next; we also discuss transfer learning techniques that motivated our approach.

A variety of (structured) prediction methods have been proposed for 3D pose estimation in the literature, including Nearest Neighbor regression (NN) [13], linear Locally-Weighted Regression (LWR) [13], Linear Regression (LR) [2], Relevance Vector Regression (RVR) [2], Kernel Regression (KR) [13] and Gaussian Process Regression (GPR) [17]. The observation that the mapping from image features to 3D pose is typically multi-modal, due to inherent imaging ambiguities, has led to introduction of multi-modal alternatives and mixture models, including Mixture of Linear Regressors (MoLR) [1], conditional Bayesian Mixture of Experts (cMoE) [4], [8], [14], [15], Local GP Regression (LGPR) [16] and Twin Gaussian Processes (TGP) [5], to name a few. Mixture models, such as MoLR and cMoE, can produce multiple solutions (one for each expert) with the hope that ambiguities can be resolved by an oracle [4], [8] or over time [14]; alternatively, optimization can be used to ascend to the most prominent mode of the conditional distribution [5]. We leverage these prior methods and propose an Importance Weighted Twin Gaussian Processes (IWTGP) model, based on TGP [5], where importance weights adopt the model to the test data at hand in an un-supervised fashion.

The methods outlined above differ significantly in learning and inference. The issue of learning from large datasets was addressed in [4] using a forward feature selection and bound optimization, allowing training of cMoE models from upward of 100,000 input-output samples. A competing issue of learning from small datasets has also received much attention, with most methods converging on intermediate shared low-dimensional latent representations (e.g., shared GPLVM (sGPLVM) [6], [9] or shared Kernel Information Embeddings (sKIE) [12]) to address overfitting with few input-output samples; some formulations were shown to be amenable semi-supervised learning settings [8], [9], [12] where a large number of *unpaired* marginal samples, which are drawn from the training distribution (not test distribution), are available. We deal with training from large datasets, as in [16] and [5], by first selecting an active set of input-output pairs (k Nearest Neighbors to the test input feature vector $\boldsymbol{x}$) and then learning an IWTGP model for this reduced set. This results in a fixed model and inference complexity regardless of training set size (apart from the initial kNN lookup).

Our method is also motivated by recent works that study effects of dataset biases in vision. The issue of dataset bias has recently emerged as a serious problem in object categorization, with Torralba and Efros [22] showing that significant biases exist in all current datasets. As a result, techniques for *domain adaptation* in object categorization are starting to emerge [18], [19], [20], [21], [24], [25], [26]. However, unlike our method, the focus of such techniques, so far, has been on a supervised setting where one or more labeled

examples are available at test time (in the target domain). This allows the source models, obtained using training data, to be adopted to the target test domain explicitly. A more recent variant by Kulis *et al*. [19] introduces a method for doing this in a cross-domain setting, where the representation of the features at train and test time may, in itself, be different. Our setting, here, is substantially different, however, as we assume that no labeled instances are present at test time. This makes the problem more challenging, but at the same time more realistic for our target application, as it is unreasonable to assume that accurate 3D pose can be annotated for monocular test images. This setting is a special case of domain adaptation known as *covariate shift* [27], where the training distribution $p_{\text{tr}}(\boldsymbol{x})$ and test distribution $p_{\text{te}}(\boldsymbol{x})$ over the inputs are different (i.e., $p_{\text{tr}}(\boldsymbol{x}) \neq p_{\text{te}}(\boldsymbol{x})$) but the conditional distribution of output values, $p(\boldsymbol{y}|\boldsymbol{x})$, remains same.

The influence of the covariate shift could be mitigated by re-weighting of the log likelihood terms according to their importance within the test set. Since the importance is generally unknown, the key issue of covariate shift adaptation is to estimate these importance weights accurately. Following this idea, several direct importance weight estimation methods have been recently proposed [33], [34], [35], [36]. In this paper, we adopt an importance weight estimation method called *relative unconstrained least-squares importance fitting* (RuLSIF) [36], since it holds practical advantage over competing methods. Mainly, it is computationally efficient and can naturally control the *adaptiveness* to the test distribution. In contrast to [36], however, we adopt RuLSIF for (structured) real-valued predictions and illustrate it's efficacy on a real-world vision problem.

3 COVARIATE SHIFT IN 3D POSE ESTIMATION

At first glance, it may not be evident why dataset bias plays a role in discriminative models, considering that discriminative methods are trying to model the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ and it seems reasonable to assume that $p(\boldsymbol{y}|\boldsymbol{x}) = p_{\text{tr}}(\boldsymbol{y}|\boldsymbol{x}) = p_{\text{te}}(\boldsymbol{y}|\boldsymbol{x})$ (even if $p_{\text{tr}}(\boldsymbol{x}, \boldsymbol{y}) \neq p_{\text{te}}(\boldsymbol{x}, \boldsymbol{y})$). In other words, how can the fact that $p_{\text{tr}}(\boldsymbol{x}) \neq p_{\text{te}}(\boldsymbol{x})$ effect the conditional distribution? The issue is that the conditional models assume a certain functional form and typically choose the optimal parameters (within this functional form) by minimizing the average regression error (i.e., average discrepancy between the predicted and true values on the *training* set). Intuitively this means that the learned model performs more accurately in the denser regions than in the sparser regions of $p_{\text{tr}}(\boldsymbol{x})$, because the denser regions dominate the average regression error. Hence, if $p_{\text{tr}}(\boldsymbol{x}) \neq p_{\text{te}}(\boldsymbol{x})$, the learned model may no longer be *optimal* for the test set.

The formulation outlined in the previous paragraph is known in the transfer learning community as one of *covariate shift* [27]. Under covariate shift setup it is assumed that labeled training image-pose pairs $\{(\boldsymbol{x}_i^{\text{tr}}, \boldsymbol{y}_i^{\text{tr}})\}_{i=1}^{n_{\text{tr}}}$ drawn i.i.d. from $p(\boldsymbol{y}|\boldsymbol{x})p_{\text{tr}}(\boldsymbol{x})$ and unlabeled test image features $\{\boldsymbol{x}_j^{\text{te}}\}_{j=1}^{n_{\text{te}}}$ drawn i.i.d. from $p_{\text{te}}(\boldsymbol{x})$ (which is usually different from $p_{\text{tr}}(\boldsymbol{x})$) are available. The goal of (structured) prediction is to learn a mapping, $\boldsymbol{f} : \mathbb{R}^{d_{\text{x}}} \rightarrow \mathbb{R}^{d_{\text{y}}}$, which in the most general form can be expressed as:

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{e} \in \mathbb{R}^{d_{\text{y}}}$ is the noise. Under covariate shift this mapping is learned based on a *weighted* set of training image-pose pairs $\{(w_i, \boldsymbol{x}_i^{\text{tr}}, \boldsymbol{y}_i^{\text{tr}})\}_{i=1}^{n_{\text{tr}}}$. Re-weighting each training instances by the ratio (a.k.a., importance weight), $w_i = w_1(\boldsymbol{x}_i^{\text{tr}}; \boldsymbol{\theta}) = \frac{p_{\text{te}}(\boldsymbol{x}_i^{\text{tr}})}{p_{\text{tr}}(\boldsymbol{x}_i^{\text{tr}})}$, removes the training set bias producing an unbiased model under assumption of covariate shift [27]. Note $\{\boldsymbol{x}_j^{\text{te}}\}_{j=1}^{n_{\text{te}}}$ are necessary to estimate the numerator. The main challenge, however, is estimation of the importance weight; we discuss this in detail in Section 4.

Before proceeding, however, we would like to illustrate the effect of *covariate shift* on a synthetic toy example. In Figure 1, we illustrate the efficacy of our re-weighting scheme under covariate shift by incorporating it into Kernel Regression (KR) and Twin Gaussian Process (TGP); for details see Section 5. As can be seen, Importance Weighted KR (IWKR) and Importance Weighted TGP (IWTGP) can predict the true test output well, while standard KR and TGP fails to predict the true test output, in particular, around $x = 0.5$ in the TGP case (see Figure 1-(e)). Note that in this specific case the mean squared error (MSE) is improved by a large margin (from 0.021 to 0.002 for IWKR and from 0.038 to 0.002 for IWTGP) by incorporating the importance weight into the learning.

4 IMPORTANCE WEIGHT ESTIMATION

The importance weight may be computed by separately estimating densities $p_{\text{tr}}(\boldsymbol{x})$ and $p_{\text{te}}(\boldsymbol{x})$ from training and test feature vectors and then taking their ratio. However, density estimation is known to be a hard problem and taking the ratio of estimated densities tends to increase the estimation error [34]. Thus, this two step approach is not appropriate in practice.

To avoid density estimation, direct density-ratio estimation approaches have recently been proposed [37]. Specifically, they estimate the ratio of probability distributions $w(\boldsymbol{x})$ directly without going through density estimation. Direct density-ratio estimation has been actively studied in the machine learning community such as kernel mean matching (KMM) [33], the Kullback-Leibler importance estimation procedure (KLIEP) [37], WKV [38], and unconstrained least-squares importance fitting (uLSIF) [35]. These direct
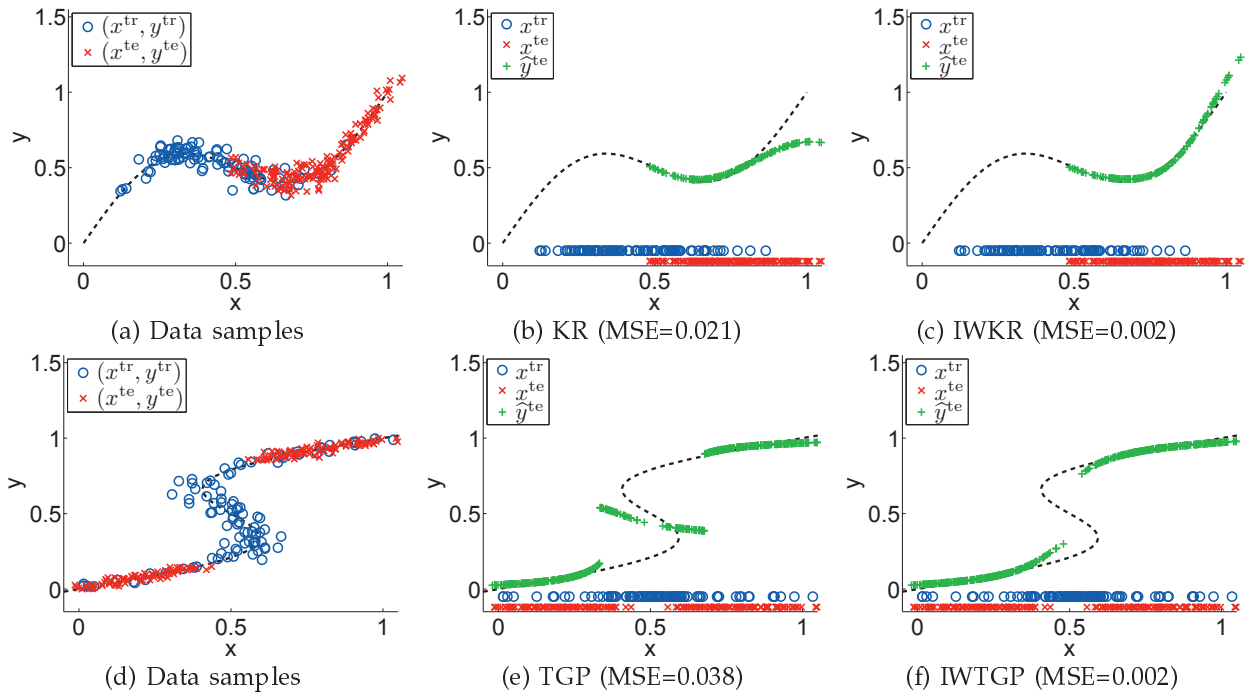
Fig. 1. Predicted outputs y by KR (b), IWKR (c), TGP (e), and IWTGP (f) under covariate shift in green. (a) *Uni-modal setting*. Samples from the model $y = x + 0.3\sin(2\pi x) + e$ where $e \sim \mathcal{N}(0, 0.05^2)$; ∘ and × are 100 training and 300 test samples respectively (for clarity we also illustrate marginals $p_{\mathrm{tr}}(x)$ and $p_{\mathrm{te}}(x)$ in (b) and (c) bottom). (d) *Multi-modal setting*. Samples from the model $x = y + 0.3\sin(2\pi y) + e$ where $e \sim \mathcal{N}(0, 0.05^2)$; ∘ and × are training and test samples respectively (for clarity we also illustrate marginals $p_{\mathrm{tr}}(x)$ and $p_{\mathrm{te}}(x)$ in (e) and (f) bottom). Note that the input-output test samples are not used in the training of KR and TGP and the output test samples are not used in the training of IWKR and IWTGP, they are plotted in the figure for illustration purposes.

density-ratio estimation methods have shown the optimal convergence rate for non-parametric density-ratio estimation.

However, since the importance weight $w(\boldsymbol{x})$ can diverge to infinity under a rather simple setting, e.g., when the ratio of two Gaussian functions is considered [39], the estimation of $w(\boldsymbol{x})$ is unstable and the covariate shift adaptation tends to be unstable [27]. To cope with this instability issue, slightly flattening the importance weights is practically useful for stabilizing the estimator (i.e., $w(\boldsymbol{x})^{\alpha}, \ 0 \le \alpha \le 1$), even though it cannot give an unbiased model under covariate shift [27]. However, since estimation of $w(\boldsymbol{x})$ is rather hard [39], all flattened weights $w(\boldsymbol{x})^{\alpha}$ are also unreliable when $w(\boldsymbol{x})$ is estimated poorly and, as a result, covariate shift adaptation does not work well in practice.

To deal with this problem, we adopt a method that allows us to directly learn the flattened importance weight function without going through density estimation and an additional flattening operation. The method is called the *relative unconstrained least-squares importance fitting* (RuLSIF) [36].

Let us first define the *relative importance weight* [36]:

$$w_\alpha(\boldsymbol{x}) = \frac{p_{\mathrm{te}}(\boldsymbol{x})}{(1-\alpha)p_{\mathrm{te}}(\boldsymbol{x}) + \alpha p_{\mathrm{tr}}(\boldsymbol{x})}, \quad 0 \le \alpha \le 1, \quad (2)$$

where α is the tuning parameter to control the

adaptiveness to the test distribution. If $\alpha = 0$ (i.e., $w_0(\boldsymbol{x}) = 1$) gives no adaptation, while $\alpha = 1$ (i.e., $w_1(\boldsymbol{x}) = \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}$) gives the full adaptation from $p_{\mathrm{tr}}(\boldsymbol{x})$ to $p_{\mathrm{te}}(\boldsymbol{x})$; $0 < \alpha < 1$ will give an intermediate estimator[2].

Let $\mathcal{X}^{\mathrm{tr}}(\subseteq \mathbb{R}^{d_{\mathrm{x}}})$ be the domain of training image feature vector $\boldsymbol{x}^{\mathrm{tr}}$ and $\mathcal{X}^{\mathrm{te}}(\subseteq \mathbb{R}^{d_{\mathrm{x}}})$ be the domain of test image feature vector $\boldsymbol{x}^{\mathrm{te}}$. Suppose we are given n_{tr} and n_{te} i.i.d. training and test image feature vectors, $\{\boldsymbol{x}_i^{\mathrm{tr}} \mid \boldsymbol{x}_i^{\mathrm{tr}} \in \mathcal{X}^{\mathrm{tr}}, \ i = 1, \ldots, n_{\mathrm{tr}}\}$, $\{\boldsymbol{x}_j^{\mathrm{te}} \mid \boldsymbol{x}_j^{\mathrm{te}} \in \mathcal{X}^{\mathrm{te}}, \ j = 1, \ldots, n_{\mathrm{te}}\}$, drawn from distributions with densities $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$, respectively.

The final goal of relative importance weight estimation is to estimate the relative importance weight based on the training and test image features. Let us model the relative importance weight $w_\alpha(\boldsymbol{x})$ by the following kernel model:

$$w_\alpha(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\ell=1}^{n_{\mathrm{te}}} \theta_\ell \kappa(\boldsymbol{x}, \boldsymbol{x}_\ell^{\mathrm{te}}) = \sum_{\ell=1}^{n_{\mathrm{te}}} \theta_\ell \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_\ell^{\mathrm{te}}\|^2}{2\tau^2}\right),$$
(3)

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n_{\mathrm{te}}})^\top$ are parameters to be learned from data samples, $^\top$ denotes the transpose, $\kappa(\cdot, \cdot)$

2. Setting α to $0 < \alpha < 1$ is practically useful for stabilizing the covariate shift adaptation, even though it cannot give an unbiased model under covariate shift [36].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. XX, NO. X, XXXX
5

is the Gaussian kernel and τ (> 0) is the kernel bandwidth.

The reason we use $\{\boldsymbol{x}_j^{\text{te}}\}_{j=1}^{n_{\text{te}}}$ as the basis centers is as follows. The importance $w_\alpha(\boldsymbol{x})$ tends to have large values if the denominator $(1-\alpha)p_{\text{te}}(\boldsymbol{x}) + \alpha p_{\text{tr}}(\boldsymbol{x})$ is small and the numerator $p_{\text{te}}(\boldsymbol{x})$ is large. On the other hand, $w_\alpha(\boldsymbol{x})$ tends to have small values if $(1-\alpha)p_{\text{te}}(\boldsymbol{x}) + \alpha p_{\text{tr}}(\boldsymbol{x})$ is large and $p_{\text{te}}(\boldsymbol{x})$ is small. If we approximate density-ratio $w_\alpha(\boldsymbol{x})$ using Gaussian kernels, many kernels may be needed in the region where $w_\alpha(\boldsymbol{x})$ is large, while we may not need to have large number of kernels in the region where $w_\alpha(\boldsymbol{x})$ is close to zero. Following this observation, we place kernels at high test input density regions, which is achieved by using test points as the Gaussian centers.

The parameters $\boldsymbol{\theta}$ in the model $w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})$ are determined so that the following expected squared-error J is minimized:

$$
\begin{aligned}
J(\boldsymbol{\theta}) &= \frac{1}{2}\mathbb{E}_{q_\alpha(\boldsymbol{x})}\left[(w_\alpha(\boldsymbol{x}; \boldsymbol{\theta}) - w_\alpha(\boldsymbol{x}))^2\right] \\
&= \frac{1}{2}\mathbb{E}_{q_\alpha(\boldsymbol{x})}\left[w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})^2 - 2w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})w_\alpha(\boldsymbol{x}) + w_\alpha(\boldsymbol{x})^2\right] \\
&= \frac{1-\alpha}{2}\mathbb{E}_{p_{\text{te}}(\boldsymbol{x})}\left[w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})^2\right] + \frac{\alpha}{2}\mathbb{E}_{p_{\text{tr}}(\boldsymbol{x})}\left[w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})^2\right] \\
&\quad - \mathbb{E}_{p_{\text{te}}(\boldsymbol{x})}[w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})] + \text{Const.},
\end{aligned}
$$

where $q_\alpha(\boldsymbol{x}) = (1-\alpha)p_{\text{te}}(\boldsymbol{x}) + \alpha p_{\text{tr}}(\boldsymbol{x})$, and we used $w_\alpha(\boldsymbol{x})q_\alpha(\boldsymbol{x}) = p_{\text{te}}(\boldsymbol{x})$.

Approximating the expectations by empirical averages and ignore the constant, we have

$$
\begin{aligned}
\widehat{J}(\boldsymbol{\theta}) = {}& \frac{1-\alpha}{2n_{\text{te}}}\sum_{j=1}^{n_{\text{te}}} w_\alpha(\boldsymbol{x}_j^{\text{te}}; \boldsymbol{\theta})^2 + \frac{\alpha}{2n_{\text{tr}}}\sum_{i=1}^{n_{\text{tr}}} w_\alpha(\boldsymbol{x}_i^{\text{tr}}; \boldsymbol{\theta})^2 \\
& - \frac{1}{n_{\text{te}}}\sum_{j=1}^{n_{\text{te}}} w_\alpha(\boldsymbol{x}_j^{\text{te}}; \boldsymbol{\theta}).
\end{aligned}
$$

By substituting the kernel model Eq.(3) into $\widehat{J}(\boldsymbol{\theta})$, we obtain the following optimization problem:

$$
\widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^{n_{\text{te}}}}\left[\frac{1}{2}\boldsymbol{\theta}^\top \widehat{\boldsymbol{H}}\boldsymbol{\theta} - \widehat{\boldsymbol{h}}^\top\boldsymbol{\theta} + \frac{\nu}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta}\right], \qquad (4)
$$

where $\nu\boldsymbol{\theta}^\top\boldsymbol{\theta}/2$ is included to avoid overfitting, and ν (≥ 0) denotes the regularization parameter. $\widehat{\boldsymbol{H}}$ is the $n_{\text{te}} \times n_{\text{te}}$ matrix with the (ℓ, ℓ')-th element

$$
\begin{aligned}
\widehat{H}_{\ell, \ell'} = {}& \frac{1-\alpha}{n_{\text{te}}}\sum_{i=1}^{n_{\text{te}}} \kappa(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{x}_\ell^{\text{te}})\kappa(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{x}_{\ell'}^{\text{te}}) \\
& + \frac{\alpha}{n_{\text{tr}}}\sum_{j=1}^{n_{\text{tr}}} \kappa(\boldsymbol{x}_j^{\text{tr}}, \boldsymbol{x}_\ell^{\text{te}})\kappa(\boldsymbol{x}_j^{\text{tr}}, \boldsymbol{x}_{\ell'}^{\text{te}}),
\end{aligned}
$$

$$
\widehat{h}_\ell = \frac{1}{n_{\text{te}}}\sum_{i=1}^{n_{\text{te}}} \kappa(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{x}_\ell^{\text{te}}).
$$

Then the solution to Eq. (4) can be *analytically* obtained as

$$
\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{H}} + \nu\boldsymbol{I})^{-1}\widehat{\boldsymbol{h}}, \qquad (5)
$$

where $\boldsymbol{I}$ is the $n_{\text{te}} \times n_{\text{te}}$-dimensional identity matrix.

The performance of RuLSIF depends on the choice of the kernel bandwidth τ and the regularization parameter ν. Model selection of RuLSIF is possible based on cross-validation with respect to the squared-error criterion J [36].

Computational complexity: Learning RuLSIF has complexity $O(n_{\text{te}}^3)$ due to the matrix inversion. However, when the number of test samples is large, we may reduce the number of kernels in Eq.(3) to $b_{\text{te}}(< n_{\text{te}})$. Then, the inverse matrix in Eq.(5) can be efficiently computed with complexity $O(b_{\text{te}}^3)$.

5 IMPORTANCE WEIGHTED 3D HUMAN POSE ESTIMATION

Given the derivation of the importance weight estimator, in previous section, we now formulate two regression-based methods that take these weights into account. We start by formulating Importance Weighted Kernel Regression (IWKR), which has a particularly simple form and allows learning of non-linear mapping between the image features and the 3D pose. IWKR, similar to standard KR, is well suited for uni-modal predictions. However, in 3D pose estimation, the mapping from image features to 3D pose has been shown to be multi-modal, due to the inherent imaging ambiguities [14]. To address this, we also introduce an Importance Weighted Twin Gaussian Processes model, based on [5], which in addition imposes structure on the output 3D poses. As a result, IWTGP is able to estimate the most prominent mode, corresponding to the most likely 3D pose, as opposed to averaging across modes as is the case with KR and IWKR.

5.1 Importance Weighted Kernel Regression

In kernel regression vector-valued regression function $\boldsymbol{f}$, in Eq.(1), takes the following form:

$$
\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{A}) = \boldsymbol{A}^\top\boldsymbol{k}(\boldsymbol{x}), \qquad (6)
$$

where $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{d_y}] \in \mathbb{R}^{(n_{\text{tr}}+1)\times d_y}$ is a model parameter, d_y is the dimensionality of pose $\boldsymbol{y}$, $\boldsymbol{k}(\boldsymbol{x}) = [1, K(\boldsymbol{x}, \boldsymbol{x}_1^{\text{tr}}), K(\boldsymbol{x}, \boldsymbol{x}_2^{\text{tr}}), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_{n_{\text{tr}}}^{\text{tr}})]^\top$, and $K(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function. We use the Gaussian kernel [40] in our experiments: $K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\rho_x^2}\right)$, where ρ_x is the kernel bandwidth.

Under covariate shift setup, the use of *relative importance weighted* risk minimization was shown to be useful for adaptation from $p_{\text{tr}}(\boldsymbol{x})$ to $p_{\text{te}}(\boldsymbol{x})$ [36]:

$$
\min_{\boldsymbol{A}}\left[\sum_{i=1}^{n_{\text{tr}}} w_\alpha(\boldsymbol{x}_i^{\text{tr}})\|\boldsymbol{y}_i^{\text{tr}} - \boldsymbol{A}^\top\boldsymbol{k}(\boldsymbol{x}_i^{\text{tr}})\|^2 + \frac{\gamma}{2}\sum_{j=1}^{d_y}\|\boldsymbol{\alpha}_j\|^2\right],
$$
(7)

where $w_\alpha(\boldsymbol{x})$ is the *relative importance weight* in Eq.(2), α is the tuning parameter to control the *adaptiveness*

to the test distribution, and $\gamma \geq 0$ is the regularization parameter; we call this *importance weighted kernel regression* (IWKR).

The solution to Eq.(7) can be obtained analytically:

$$\widehat{A} = (\widetilde{K}^{\mathrm{tr}} W (\widetilde{K}^{\mathrm{tr}})^\top + \gamma I)^{-1} \widetilde{K}^{\mathrm{tr}} W (Y^{tr})^\top, \quad (8)$$

where $\widetilde{K}^{\mathrm{tr}} = [k(x_1^{\mathrm{tr}}), \ldots, k(x_{n_{\mathrm{tr}}}^{\mathrm{tr}})] \in \mathbb{R}^{(n_{\mathrm{tr}}+1) \times n_{\mathrm{tr}}}$, $Y^{tr} = [y_1^{\mathrm{tr}}, \ldots, y_{n_{\mathrm{tr}}}^{\mathrm{tr}}] \in \mathbb{R}^{d_y \times n_{\mathrm{tr}}}$, and W is the $n_{\mathrm{tr}} \times n_{\mathrm{tr}}$-dimensional diagonal matrix with (i,i)-th diagonal element defined by $W_{i,i} = w_\alpha(x_i^{\mathrm{tr}})$.

The above IWKR method includes two tuning parameters: kernel parameter ρ_{x} and the regularization parameter γ. These parameters can be selected using importance-weighted variant of cross-validation (IWCV) [41].

Computational complexity: Learning IWKR has complexity $O((n_{\mathrm{tr}} + 1)^3)$. Similar to RuLSIF, when the number of training data is large, we may reduce the number of kernels in Eq.(6) to $b_{\mathrm{tr}}(< n_{\mathrm{tr}} + 1)$. Then, the inverse matrix in Eq.(8) can be efficiently computed with complexity $O(b_{\mathrm{tr}}^3)$. Since IWKR also includes the estimation of relative importance weight and its complexity is $O(b_{\mathrm{te}}^3)$. Thus, the total complexity of IWKR is $O(b_{\mathrm{tr}}^3) + O(b_{\mathrm{te}}^3)$.

5.2 Importance Weighted Twin Gaussian Processes

We now propose the importance-weighted variant of twin Gaussian processes [5] called IWTGP. The benefit of IWTGP over IWKR is that it can naturally take into account the multi-modality present in the human pose estimation, by incorporating structure over the output poses into the regression.

5.2.1 Gaussian Processes Regression

The Gaussian Process (GP) regression assumes a linear model in the function space with Gaussian noise for the k-th dimension (e.g., joint position):

$$y_k = f_k(x) + e_k, \ e_k \sim \mathcal{N}(0, \sigma^2), \qquad f_k(x) = \beta_k^\top \phi(x), \tag{9}$$

where there is a zero mean Gaussian prior over the parameters $\beta_k \sim \mathcal{N}(0_p, \Sigma_p)$; 0_p is the p-dimensional zero vector and Σ_p is the p-dimensional covariance matrix, $\phi(x)$ is the function which maps a d_{x} dimensional input vector x into an p dimensional feature space. To make prediction for the test sample, one needs to average over all possible parameter values, weighted by their posterior, resulting in a Gaussian predictive distribution. GP has similar problems with multi-modality as KR. To address this limitation, TGP encodes the relations between both inputs and outputs using GP priors. This is achieved by minimizing the Kullback-Leibler divergence between the marginal GP of outputs (poses) and observations (features).

5.2.2 Twin Gaussian Process Regression

For completeness, we include the TGP formulation here, following closely the original derivation in [5]. To do so, we take an alternative view to GP regression, where a joint distribution over all training outputs, $Y^{tr} = [y_1^{\mathrm{tr}}, \ldots, y_{n_{\mathrm{tr}}}^{\mathrm{tr}}] \in \mathbb{R}^{d_y \times n_{\mathrm{tr}}}$, and an unknown test output y for a given corresponding input x takes the form of a joint Gaussian:

$$\left[\begin{array}{c} Y_k^\top \\ y_k \end{array} \right] \sim \mathcal{N}_X \left(0, \ \left[\begin{array}{cc} K & k(x) \\ k(x)^\top & K(x,x) \end{array} \right] \right), \tag{10}$$

where Y_k is the k-th row of Y, y_k is the k-th entry of y, K is an $n_{\mathrm{tr}} \times n_{\mathrm{tr}}$ matrix with each element $K_{ij} = cov(\phi(x_i), \phi(x_j))$ being a covariance function encoding correlations between pairs of random variables x_i and x_j; similarly $k_i(x) = cov(\phi(x_i), \phi(x))$ is a column vector of size $n_{\mathrm{tr}} \times 1$ and $K(x,x) = cov(\phi(x), \phi(x))$. A popular choice for a covariance function is a Gaussian with noise:

$$\begin{aligned} cov(\phi(x_i), \phi(x_j)) &= K(x_i, x_j) \\ &= \exp\left(-\frac{||x_i - x_j||^2}{2\rho_{\mathrm{x}}^2} \right) + \lambda_{\mathrm{x}} \delta_{ij}, \end{aligned}$$

where, ρ_{x} is the kernel bandwidth parameter and λ_{x} is the noise variance; δ_{ij} is the Kronecker delta function.

Because the joint distribution is Gaussian, the predictive distribution is also Gaussian and can be obtained by conditioning on the observed training outputs Y_k. The mean and variance of the predictive distribution can be derived in closed form:

$$m(y_k) = Y_k K^{-1} k(x),$$
$$\sigma^2(y_k) = K(x,x) - k(x)^\top K^{-1} k(x).$$

Unlike GP regression, TGP also defines the covariance function over outputs (not just inputs), which allows to model correlations in outputs. In addition, by minimizing KL divergence between the two Gaussian Processes (one going in the forward and one in the backwards direction) it is possible to focus on the most prominent mode in the potentially multi-modal mapping between image features and 3D pose.

To derive the backwards GP process, note that $\left[Y_k^\top, y_k \right]^\top$, in Eq. (10), can also be thought of as a sample from a Gaussian distribution over the outputs,

$$\left[\begin{array}{c} Y_k^\top \\ y_k \end{array} \right] \sim \mathcal{N}_{Y_k \cup y_k} \left(0, \ \left[\begin{array}{cc} L & l(y_k) \\ l(y_k)^\top & L(y_k, y_k) \end{array} \right] \right), \tag{11}$$

where we can empirically estimate covariance matrix as:

$$L_{Y_k \cup y_k} = \left[\begin{array}{c} Y_k^\top \\ y_k \end{array} \right] \left[\begin{array}{cc} Y_k & y_k \end{array} \right].$$

For a multivariate output case, we can treat each dimension k as an independent samples from the Gaussian distribution (see Eq. (10)) since covariance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. XX, NO. X, XXXX   7

is independent of k. Hence, we can instead estimate covariance matrix as:

$$\boldsymbol{L}_{\boldsymbol{Y} \cup \boldsymbol{y}} = \frac{1}{d_{\mathrm{y}}} \begin{bmatrix} \boldsymbol{Y}^\top \\ \boldsymbol{y}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{y} \end{bmatrix}.$$

This, however, assumes that outputs along each dimension are i.i.d.; to account for the correlations between outputs, we can define a more general covariance function over the outputs, resulting in:

$$\begin{bmatrix} \boldsymbol{Y}^\top \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}_{\boldsymbol{Y} \cup \boldsymbol{y}} \left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{L} & \boldsymbol{l}(\boldsymbol{y}) \\ \boldsymbol{l}(\boldsymbol{y})^\top & L(\boldsymbol{y}, \boldsymbol{y}) \end{bmatrix} \right), \quad (12)$$

which bares close similarity to the original GP in Eq. (10).

TGP measures the offset between the true Gaussian distribution of the inputs, $\mathcal{N}_{\boldsymbol{X}}$, and measured Gaussian distribution of the outputs, $\mathcal{N}_{\boldsymbol{Y} \cup \boldsymbol{y}}$, using Kullback-Leibler divergence. However, the output, $\boldsymbol{y}$, is unknown in this measure. To match the estimated output distribution and fully observed input one as much as possible, one is required to estimate output, $\hat{\boldsymbol{y}}$, by minimizing the Kullback-Leibler divergence:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathbb{R}^{d_{\mathrm{y}}}}{\operatorname{argmin}} \, D_{KL}(\mathcal{N}_{\boldsymbol{X}} \, || \, \mathcal{N}_{\boldsymbol{Y} \cup \boldsymbol{y}}),$$

where $\mathcal{N}_{\boldsymbol{X}}$ and $\mathcal{N}_{\boldsymbol{Y} \cup \boldsymbol{y}}$ are defined in Eq. (10) and Eq. (12) respectively.

As a result, inference in TGP is given as the solution to the following optimization problem [5]:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathbb{R}^{d_{\mathrm{y}}}}{\operatorname{argmin}} \Big[1 + \lambda_{\mathrm{y}} - 2\boldsymbol{l}(\boldsymbol{y})^\top \boldsymbol{u}$$
$$- \eta \log \big[1 + \lambda_{\mathrm{y}} - \boldsymbol{l}(\boldsymbol{y})^\top \boldsymbol{L}^{-1} \boldsymbol{l}(\boldsymbol{y}) \big] \Big], \quad (13)$$

where $\boldsymbol{u} = \boldsymbol{K}^{-1} \boldsymbol{k}(\boldsymbol{x})$, $\eta = K(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^\top \boldsymbol{u}$, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\rho_{\mathrm{x}}^2}\right) + \lambda_{\mathrm{x}} \delta_{ij}$ and $L(\boldsymbol{y}_i, \boldsymbol{y}_j) = \exp\left(-\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2}{2\rho_{\mathrm{y}}^2}\right) + \lambda_{\mathrm{y}} \delta_{ij}$ are the Gaussian kernel function for image feature vector $\boldsymbol{x}$ and pose feature vector $\boldsymbol{y}$, ρ_{x} and ρ_{y} are the kernel bandwidth, $\boldsymbol{l}(\boldsymbol{y}) = [L(\boldsymbol{y}, \boldsymbol{y}_1), \dots, L(\boldsymbol{y}, \boldsymbol{y}_{n_{\mathrm{tr}}})]^\top$, $\boldsymbol{k}(\boldsymbol{x}) = [K(\boldsymbol{x}, \boldsymbol{x}_1), \dots, K(\boldsymbol{x}, \boldsymbol{x}_{n_{\mathrm{tr}}})]^\top$, and λ_{y} and λ_{x} are regularization parameters to avoid overfitting. This nonlinear optimization problem can be solved using a second order BFGS quasi-Newton optimizer with cubic polynomial line search for optimal step size selection [5].

5.2.3   Importance Weighting in TGP

Under covariate shift, the likelihood of Gaussian Process can be given as [27]:

$$\prod_{i=1}^{n_{\mathrm{tr}}} p(y_i^{tr} | \boldsymbol{x}_i^{\mathrm{tr}}, \boldsymbol{\beta})^{w_\alpha(\boldsymbol{x}_i^{\mathrm{tr}})}$$
$$\propto \prod_{i=1}^{n_{\mathrm{tr}}} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}}) y_i^{tr} - w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}}) \boldsymbol{\phi}(\boldsymbol{x}_i^{\mathrm{tr}})^\top \boldsymbol{\beta}\|^2}{2\sigma^2}\right), \quad (14)$$

where $w_\alpha(\boldsymbol{x})$ is the relative importance weight function. Note, if we consider the MAP estimate for

Eq. (14) with a prior distribution over $\boldsymbol{\beta}$, then we can show that IWKR and Eq. (14) are one and the same.

Thus, the GP regression model under covariate shift can be represented by

$$w_\alpha^{\frac{1}{2}}(\boldsymbol{x}) y_k = w_\alpha^{\frac{1}{2}}(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\beta}_k + e_k, \; e_k \sim \mathcal{N}(0, \sigma^2). \quad (15)$$

That is, to achieve covariate shift adaptation in TGP, we need to simply re-weight each input and output by $w_\alpha^{\frac{1}{2}}(\boldsymbol{x})$. Therefore, the output of the importance weighted TGP (IWTGP) is given by

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathbb{R}^{d_{\mathrm{y}}}}{\operatorname{argmin}} \Big[1 + \lambda_{\mathrm{y}} - 2\boldsymbol{l}_w(\boldsymbol{y})^\top \boldsymbol{u}_w$$
$$- \eta_w \log \big[1 + \lambda_{\mathrm{y}} - \boldsymbol{l}_w(\boldsymbol{y})^\top \boldsymbol{L}_w^{-1} \boldsymbol{l}_w(\boldsymbol{y}) \big] \Big], \quad (16)$$

where $\eta_w = 1 + \lambda_{\mathrm{x}} - \boldsymbol{k}_w(\boldsymbol{x})^\top \boldsymbol{u}_w$, $\boldsymbol{u}_w = \boldsymbol{K}_w^{-1} \boldsymbol{k}_w(\boldsymbol{x}^{\mathrm{tr}})$, $[\boldsymbol{K}_w]_{i,j} = w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}}) w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_j^{\mathrm{tr}}) \exp\left(-\frac{\|\boldsymbol{x}_i^{\mathrm{tr}} - \boldsymbol{x}_j^{\mathrm{tr}}\|^2}{2\rho_{\mathrm{x}}^2}\right) + \lambda_{\mathrm{x}} \delta_{ij}$ and $[\boldsymbol{L}_w]_{i,j} = w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}}) w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_j^{\mathrm{tr}}) \exp\left(-\frac{\|\boldsymbol{y}_i^{\mathrm{tr}} - \boldsymbol{y}_j^{\mathrm{tr}}\|^2}{2\rho_{\mathrm{y}}^2}\right) + \lambda_{\mathrm{y}} \delta_{ij}$, $[\boldsymbol{k}_w(\boldsymbol{x})]_i = w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}}) \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i^{\mathrm{tr}}\|^2}{2\rho_{\mathrm{x}}^2}\right)$, and $[\boldsymbol{l}_w(\boldsymbol{y})]_i = w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}}) \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{y}_i^{\mathrm{tr}}\|^2}{2\rho_{\mathrm{y}}^2}\right)$. IWTGP can also be solved using a second order, BFGS quasi-Newton optimizer with cubic polynomial line search for optimal step size selection. We ignore the weighting for certain terms that are independent of $\boldsymbol{y}$, and hence do not effect the optimization, for simplicity.

Computational complexity: IWTGP requires matrix inversions of $n_{\mathrm{tr}} \times n_{\mathrm{tr}}$ matrices, the complexity of solving Eq.(16) is $O(n_{\mathrm{tr}}^3)$, which is impractical when n_{tr} is large. To deal with this issue, we first find the M nearest neighbors of a test input and estimate IWTGP on the reduced set of training paired samples. Then, the inverse matrix in Eq.(16) can be efficiently computed with complexity $O(M^3)$. IWTGP also includes the estimation of relative importance weight, thus the total complexity of IWTGP is $O(M^3) + O(b_{\mathrm{te}}^3)$.

5.3   Importance Weighting for Other Methods

The proposed weighting methodology is amenable to most popular formulations (e.g., Linear Regression, conditional Bayesian Mixture of Experts (cMoE), GPLVM, KIE), as well as to other (structured) prediction problems in computer vision. For example, in Linear Regression importance weighting can be incorporate via Weighted Linear Regression. Incorporating importance weighting into cMoE would amount to secondary weighting on top of expert assignment; for cMoE models with soft expert assignments this would require very minor changes to the learning procedure. Latent variants like GPLVM and KIE can also make use of the importance weighting, for example, in KIE the importance weighted version of Mutual Information can be used to learn an IWKIE model.

6 EXPERIMENTS

We compare the performance of the proposed methods IWKR and IWTGP with their un-weighted counterparts, KR [1] and TGP (we use public implementation from [5]), and weighted k-Nearest Neighbors approach (WkNN) [13]. We report performance on two publicly available datasets: Poser [2] and HUMANEVA-I [23].

Parameters: For Poser dataset, we experimentally (through grid search) set the TGP and IWTGP parameters to $\lambda_x = \lambda_y = 10^{-4}$, $2\rho_x^2 = 5$, and $2\rho_y^2 = 5000$. For HUMANEVA-I dataset, we used the original parameter setting of [5]: $\lambda_x = \lambda_y = 10^{-3}$, $2\rho_x^2 = 5$, and $2\rho_y^2 = 5 \times 10^5$. The number of M nearest neighbors in TGP and IWTGP is set to $min(800, n_{tr})$. In RuLSIF, we set the $\alpha = 0.5$, since it performs favorably (see Figures 4-(a), (b), and (c) for the quantitative analysis with respect to the α parameter), and $b_{te} = min(500, n_{te})$. For KR and IWKR, we set $b_{tr} = min(500, n_{tr})$, and all the parameters are chosen by cross-validation (CV) and importance weighted CV; in WkNN we set the number of nearest neighbors to 25. In addition, instead of using the entire test set to adopt the model, we use a temporal window of 20 frames (feature vectors) around the current test sample to compute the importance weight for IWTGP and IWKR. This is more efficient and is also more realistic, as one will typically not see the full set of test examples all at once. Note, the temporal window size corresponds to the number of test samples n_{te}. In addition, we carried out the quantitative analysis with respect to the temporal window size of 1, 10, 20, 30 and all frames, and we found their performance are not significantly different. Thus, we experimentally choose 20 frames in this paper. (see Figures 4-(d), (e), and (f) for the quantitative analysis).

Computational speed: The overhead for importance weighting is small compared to the base methods; for example, IWTGP is about 4% slower than TGP when entire training set is used. Moreover, experimentally we observed that IWTGP can be faster than TGP with few samples (see Figure 2). We attribute this to the fact that weighting in TGP can lead to an easier optimization problem, offsetting the coast of the weight estimation itself.

6.1 Poser Dataset

Poser dataset [2] consists of 1927 training and 418 test images, which are synthetically generated, using Poser software package, from motion capture (Mocap) data (54 joint angles per frame). The image features, corresponding to bag-of-words representation with silhouette-based shape context features, and error metric are provided with the dataset [2]. In this experiment, we use the entire 418 test images for covariate shift adaptation. Since the Poser data is synthetically
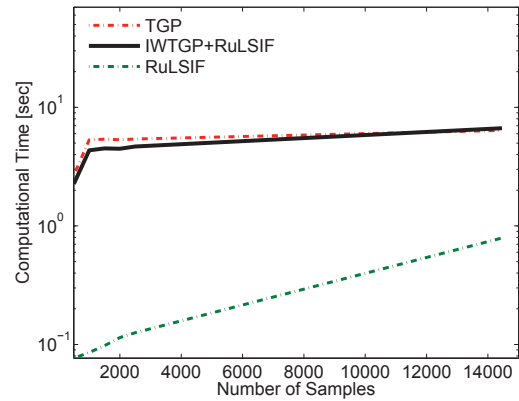

Fig. 2. **Computational time comparison**. Mean computational time of TGP, IWTGP + RuLSIF, and RuLSIF with respect to the number of training samples. The horizontal axis denotes the number of training samples, and the vertical axis denotes the computational time (sec). In this experiment, we use "Boxing" sequence of subject S1 of HUMANEVA-I dataset for testing, where the number of frames is 251. The number of test samples (window size) is set to 20.

generated and was tuned to uni-modal predictions [2], there exists only a small bias between training and test images/features.

Error metric: The proposed error measure amounts to the root mean square error (in degrees), averaged over all joint angles, and is given by: $Error_{pose}(\widehat{y}, y^*) = \frac{1}{54}\sum_{m=1}^{54} \|(\widehat{y}^{(m)} - y^{*(m)}) \mod 360°\|$, where $\widehat{y} \in \mathbb{R}^{54}$ is an estimated and $y^* \in \mathbb{R}^{54}$ is a true pose vector.

Performance: Table 1 shows the pose estimation result averaged across the test set. Proposed IWKR and IWTGP outperform their un-weighted counterparts, reducing error by 5% and 2% respectively. IWKR and IWTGP also compare favorably with other existing methods reported elsewhere. It is worth mentioning that Shared KIE required a local model computed using a small neighborhood of 25 training samples to achieve comparable performance (with the global model the performance drops from 5.77 to 5.95 degrees on average). In contrast, the IWKR and IWTGP models are more global; IWTGP takes 800 neighbors into account and IWKR uses all the training data[3].

6.2 HumanEva-I Dataset

HUMANEVA-I contains synchronized multi-view video and Mocap data. It consists of 3 subjects performing multiple activities: walking, jogging, boxing, throw and catch, and gesturing. We use the histogram of oriented gradient (HoG) features ($\in \mathbb{R}^{270}$) proposed in [5] (we refer to [5] for

---

3. While all the data is used it is dynamically re-weighed based on the importance weight so not all of it is *active* at all times.

TABLE 1
Performance of IWKR and IWTGP on Poser dataset.

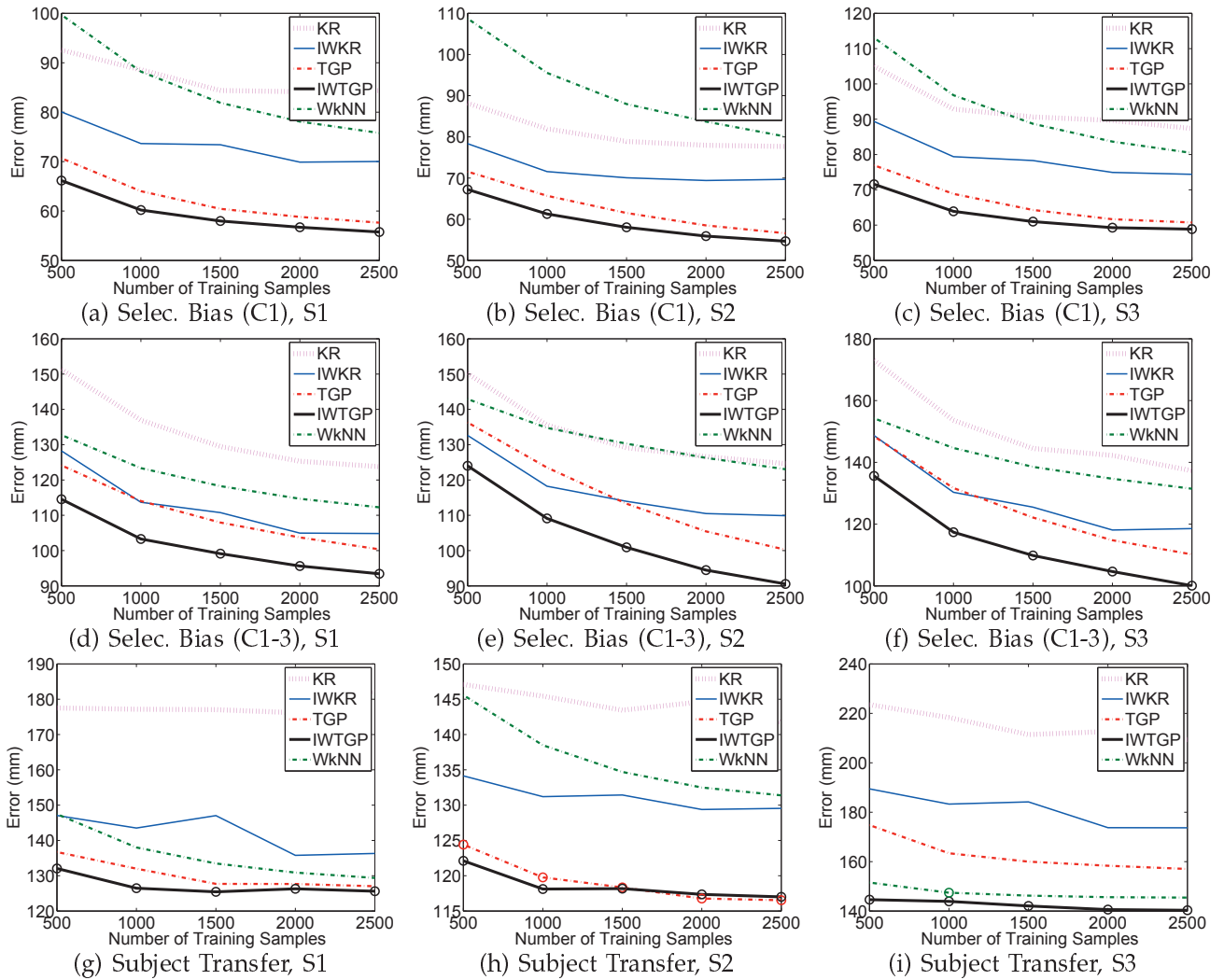|	IWTGP	TGP	IWKR	KR	NN [12]	GPLVM [6]	sKIE [12]
Error (deg)	**5.75**	5.83	**5.72**	6.04	6.87	6.50	5.77/5.95

Fig. 3. Performance on HUMANEVA-I dataset illustrated as a function of the number of training samples; we averaged the error over all motions for each subject. Comparable methods according to the paired *t-test* at the significance level 5% are specified by '○'.

details[4]). We use training and validations sub-sets of HUMANEVA-I and only utilize data from 3 color cameras with a total of 9630 image-pose frames for each camera. This is consistent with experiments in [5]. We use half of the data (4815 frames) for training and half (4815 frames) for testing; the test and training data is disjoint. Where fewer, e.g., $n_{tr} = 500$, training samples are necessary (as in Figure 3) we randomly sub-sample n_{tr} from the full training set; to alleviate the sampling bias we sample 10 times and average the resulting errors.

The bias in pose estimation can come in (at least)

two forms: (1) the training data may simply be biased and, for example, not contain the subject present in the test set (we call this *subject transfer*), or (2) the training data may contain data from variety of subjects, motions and cameras, where as at test time only a sub-set of that data is presented at any given time (we call this *selection bias*). To evaluate our methods under such scenarios we propose 3 experiments of interest:

Selection bias (C1): Only camera 1 data is used for training and testing.

Selection bias (C1-3): All camera data is used for training and testing ($3 \times 4815 = 14445$ frames of training and 14445 frames of test data).

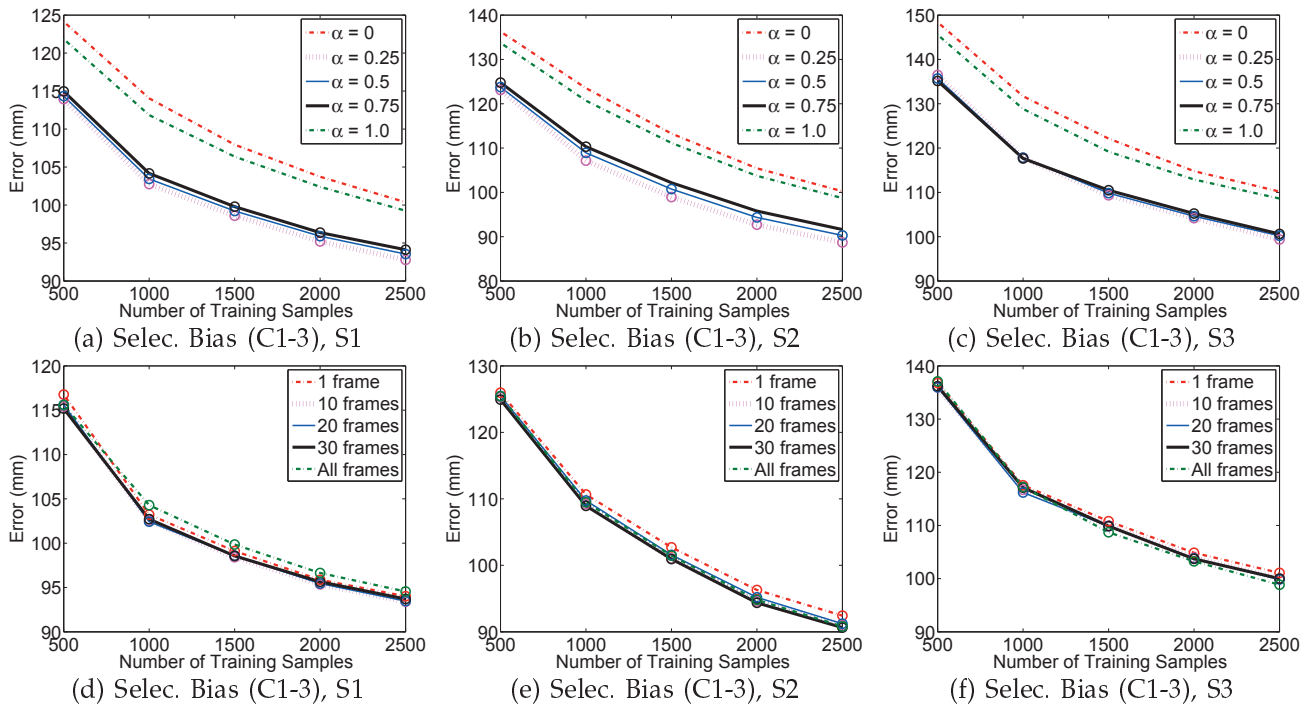Subject transfer (C1): Test subject is not included in training phase.

Fig. 4. (a)-(c): Performance comparison of IWTGP with respect to the α parameter on HUMANEVA-I dataset illustrated as a function of the number of training samples; we averaged the error over all motions for each subject. In these experiments, we fix the temporal window size to 20 frames, where it corresponds to the number of test samples used for RuLSIF computation. (d)-(f): Performance comparison of IWTGP with respect to the temporal window size on HUMANEVA-I dataset illustrated as a function of the number of training samples; we averaged the error over all motions for each subject. Here, we fix the α parameter to 0.5. The best method achieving the smallest mean error and comparable methods according to the paired *t-test* at the significance level 5% are specified by '∘'.

Error metric: In HUMANEVA-I pose is encoded by (20) 3D joint markers defined relative to the 'torsoDistal' joint in camera-centric coordinate frame, so $\boldsymbol{y} = [\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(20)}]^\top \in \mathbb{R}^{60}$ and $\boldsymbol{y}^{(i)} \in \mathbb{R}^3$. Error (in mm) for each pose is measured as average Euclidean distance: $Error_{pose}(\widehat{\boldsymbol{y}}, \boldsymbol{y}^*) = \frac{1}{20} \sum_{m=1}^{20} \|\widehat{\boldsymbol{y}}^{(m)} - \boldsymbol{y}^{*(m)}\|$, where $\widehat{\boldsymbol{y}}$ is an estimated pose vector, and $\boldsymbol{y}^*$ is a true pose vector.

Performance: Figure 3 shows the average mean pose estimation error as a function of training set size (averaged over all motions and 10 runs). The graphs clearly show that IWTGP and IWKR outperform their un-weighted counterparts. Moreover, IWTGP overall compares favorably with existing methods in terms of the overall performance.

Table 2 shows performance using the entire training set. IWTGP tends to have smaller error compared to all other methods. Note that both the weighted and their un-weighted counterparts use the same parameters and inference procedures; the key difference is in the interest weighting that alters the learning. Moreover, paired t-tests were conducted for all experiments, we observe that about 80% cases the importance weighted methods, IWTGP and IWKR, statistically outperform their non-weighted counterparts at

p=0.05 (5%) significance. In certain settings, we see more drastic improvements, e.g., 14% reduction in error in subject transfer with S3 using IWTGP (and 19% using IWKR), or over 10% reduction in error in selection bias (C1-3) with S3. We also see significant improvements on certain specific motions (see Table 3 and Figures 6-7), where, for example, on gesture motion under selection bias we observe improvement by 22.6 mm (reducing error by 20%) or under subject transfer by 64.5 mm (reducing error by 33%).

We observed that WkNN performs well (better than TGP) in subject transfer for S3, while global methods such as KR perform poorly. This implies that local information tends to be more important than global information in this case. We believe IWTGP and IWKR work well here, because while dealing with more global representation, they can emphasize locally important information through importance weighting. Note, for sample selection bias cases, the difference between TGP and IWTGP is smaller compared to cases with less training samples (see Figure 3). This is because selection bias tends to be small if the number of training samples is large. However, in subject transfer case, the prediction performance of TGP cannot be improved simply by increasing the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. XX, NO. X, XXXX 11

TABLE 2
Performance on the entire HUMANEVA-I dataset averaged over all motions.

| Transfer Type | Subject | | **IWTGP** | TGP | **IWKR** | KR | WkNN |
	Train	Test					
Selection Bias (C1)	S1,S2,S3	S1	**54.2**	55.1	**71.1**	80.1	70.2
	S1,S2,S3	S2	**52.5**	53.2	**67.6**	75.5	71.5
	S1,S2,S3	S3	**57.5**	57.9	**75.1**	86.0	72.5
Selection Bias (C1-3)	S1,S2,S3	S1	**81.9**	83.9	**101.9**	119.9	94.3
	S1,S2,S3	S2	**72.7**	75.1	**102.7**	120.0	100.7
	S1,S2,S3	S3	**77.2**	86.1	**111.7**	134.8	110.4
Subject Transfer (C1)	S2,S3	S1	**126.2**	126.9	**137.3**	168.4	128.2
	S1,S3	S2	116.7	**116.6**	**130.5**	141.5	130.6
	S1,S2	S3	**140.0**	159.7	**168.4**	209.1	145.5

number of training samples, since the training and test distributions are inherently different. In contrast, IWTGP can improve the prediction accuracy in this case by properly weighting the training samples. We find IWTGP is best suited for cases where training and test distributions are different but do overlap to a certain extent.

Analysis of importance weights: To get additional insights into the effect of relative importance weighting we plot a histogram of importance weights for Selection Bias (C1-3) experiment, subject S3, Gesture action in Figure 5 (window size is set to 20). Note that the number of importance weights for *each* test sample is 800, one for each training nearest neighbor in the active set. As can be seen from the figure, in general, the importance weighting mostly down weights irrelevant training samples (large fraction of importance weights fall below 1.0) and up weights (assigns importance weight ≥ 1) few training samples that are most relevant with respect to the test sample.

Role of parameters: The two parameters that must be set by hand in the importance weighting are the α and the temporal window size. We carried out a set of experiments to illustrate the sensitivity of the method to those parameters in Figure 4. In particular, as can be seen from Figure 4 (a)-(c) the method is not sensitive to the exact value of α so long as it's within a moderate range, $0.25 \leq \alpha \leq 0.75$. Note that performance, as expected, degrades as α goes to 0 (no importance weighting) or 1 (full adaptation, which can make weights diverge to infinity).

We also test the effect of temporal window size (number of test samples). The experiments in Figure 4 (d)-(f) again illustrate insensitivity to exact value of the parameter. As a result we are able to use shorter window of 20 frames (or even fewer) without degrading the performance. Note that performance degrades only very slightly as window is decreased to only 1 frame. We attribute this, in part, to the core property of our model, that estimates the importance weight ratio directly and, in part, to the fact that the resulting importance estimate is always centered, and hence is most accurate, at the current test point regardless of the window size.
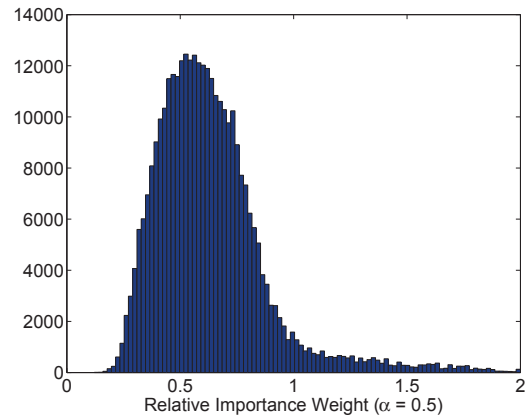

Fig. 5. **Histogram of importance weights**. Horizontal axis denotes relative importance weight and vertical axis – count of the training samples to which weights within the specified range were assigned. We indiscriminately plot histogram over all importance weights obtained during inference of "Gesture" sequence of subject S3 in the Selection Bias (C1-3) experiment of HUMANEVA-I dataset. The number of test samples (window size) is set to 20. Most importance weights down weight irrelevant training samples, while few emphasize most relevant training exemplars with respect to the test samples.

7 CONCLUSIONS

We propose a simple, yet effective, unsupervised method for addressing training set bias through covariate shift adaptation in (structured) prediction problems. As part of our formulation, we also introduce importance weighted variants of kernel regression (IWKR) and twin Gaussian processes (IWTGP) which produce state-of-the-art 3D pose estimation performance on standard datasets (HUMANEVA-I and Poser [2]). We view our approach as the first step towards eliminating bias in structured prediction problems in vision. That said, additional work is needed to address more extreme biases, e.g., biases that lead to entirely disjoint training and test dataset. We leave such extreme biases as subject of future work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. XX, NO. X, XXXX 12

TABLE 3
Performance on the entire HUMANEVA-I dataset; we average errors over all motions and frames.

Transfer Type	Subject Train	Test	Motion	IWTGP	TGP	IWKR	KR	WkNN
Selection Bias (C1)	S1,S2,S3	S1	Walking	**40.4**	41.1	63.6	69.6	57.2
			Jogging	**51.7**	52.0	74.6	85.3	75.6
			Gestures	**10.5**	11.1	15.9	18.3	11.8
			Box	**39.1**	40.0	55.5	62.4	50.1
			ThrowCatch	**129.1**	131.5	145.7	165.2	156.6
			Average	**54.2**	55.1	71.1	80.1	70.2
	S1,S2,S3	S2	Walking	**24.8**	26.5	44.5	55.1	41.7
			Jogging	**39.3**	40.2	55.6	63.1	60.0
			Gestures	65.2	**65.0**	77.8	83.0	72.4
			Box	**59.9**	60.5	76.2	85.6	91.7
			ThrowCatch	74.2	**74.0**	83.9	90.6	91.9
			Average	**52.5**	53.2	67.6	75.5	71.5
	S1,S2,S3	S3	Walking	**57.7**	59.2	80.6	91.0	80.9
			Jogging	**34.2**	35.2	53.9	64.7	58.7
			Gestures	72.7	**70.6**	91.9	109.5	82.7
			Box	**65.3**	66.8	73.8	78.6	67.6
			ThrowCatch	—	—	—	—	—
			Average	**57.5**	57.9	75.1	86.0	72.5
Selection Bias (C1-3)	S1,S2,S3	S1	Walking	**92.4**	96.3	120.4	132.9	114.5
			Jogging	**122.9**	123.5	130.8	142.6	130.4
			Gestures	**11.2**	12.0	26.3	43.7	12.0
			Box	**42.0**	43.4	68.5	91.0	54.2
			ThrowCatch	**141.2**	144.1	163.5	189.1	160.3
			Average	**81.9**	83.9	101.9	119.9	94.3
	S1,S2,S3	S2	Walking	**36.3**	39.9	85.9	111.4	68.6
			Jogging	**94.3**	97.1	127.0	141.4	146.3
			Gestures	**89.6**	91.0	103.7	115.4	97.8
			Box	**60.8**	62.7	89.8	108.4	83.0
			ThrowCatch	**82.4**	84.9	107.3	123.2	107.6
			Average	**72.7**	75.1	102.7	120.0	100.7
	S1,S2,S3	S3	Walking	**98.6**	104.5	137.2	154.2	132.8
			Jogging	**64.4**	70.5	121.4	144.9	127.7
			Gestures	**87.9**	110.4	102.6	139.8	106.0
			Box	**58.0**	59.1	85.5	100.3	75.2
			ThrowCatch	—	—	—	—	—
			Average	**77.2**	86.1	111.7	134.8	110.4
Subject Transfer (C1)	S2,S3	S1	Walking	**94.9**	98.3	115.3	143.4	95.6
			Jogging	106.0	110.6	115.6	135.6	**104.6**
			Gestures	**86.3**	87.9	121.4	187.0	100.9
			Box	151.6	148.5	**142.5**	156.8	144.0
			ThrowCatch	191.9	**189.4**	191.8	219.0	196.0
			Average	**126.2**	126.9	137.3	168.4	128.2
	S1,S3	S2	Walking	**108.9**	**108.9**	124.5	127.2	126.9
			Jogging	92.1	**91.1**	117.4	121.3	141.8
			Gestures	96.3	**94.8**	110.1	128.5	100.8
			Box	177.6	178.1	168.5	184.1	**168.8**
			ThrowCatch	**108.4**	109.9	132.2	146.5	114.6
			Average	116.7	**116.6**	130.5	141.5	130.6
	S1,S2	S3	Walking	**130.7**	132.7	150.1	192.7	133.6
			Jogging	**116.9**	120.2	147.3	193.4	139.6
			Gestures	**133.4**	197.9	186.5	191.2	140.1
			Box	179.2	188.0	189.7	259.2	**168.5**
			ThrowCatch	—	—	—	—	—
			Average	**140.0**	159.7	168.4	209.1	145.5

REFERENCES

[1] A. Agarwal and B. Triggs, "Monocular human motion capture with a mixture of regressors," in *CVPR Workshop*, 2005.

[2] A. Agarwal and B.Triggs, "Recovering 3D human pose from monocular images," *IEEE Trans. on PAMI*, vol. 28, no. 1, pp. 44—58, 2006.

[3] A. Bissacco, M. Yang, and S. Soatto, "Fast human pose estimation using appearance and motion via multi-dimensional boosting regression," in *CVPR*, 2007, pp. 1–8.

[4] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Fast algorithms for large scale conditional 3d prediction," in *CVPR*, 2008.

[5] L. Bo and C. Sminchisescu, "Twin gaussian processes for structured prediction," *Int. J. Comput. Vision*, vol. 87, no. 1-2, pp. 28–52, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-008-0204-y

[6] C. Ek, P. Torr, and N. Lawrence, "Gaussian process latent variable models for human pose estimation," in *Workshp on ML for Mult. Inter.*, vol. 4892. LNCS, 2007.

[7] C. Ionescu, L. Bo, and C. Sminchisescu, "Structural svm for visual localization and continuous state estimation," in *ICCV*, 2009.

[8] A. Kanaujia, C. Sminchisescu, and D. Metaxas, "Semi-supervised hierarchical models for 3D human pose reconstruction," in *CVPR*, 2007.

[9] R. Navaratnam, A. Fitzgibbon, and R. Cipolla, "The joint manifold model for semi-supervised multi-valued regression,"

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. XX, NO. X, XXXX   13
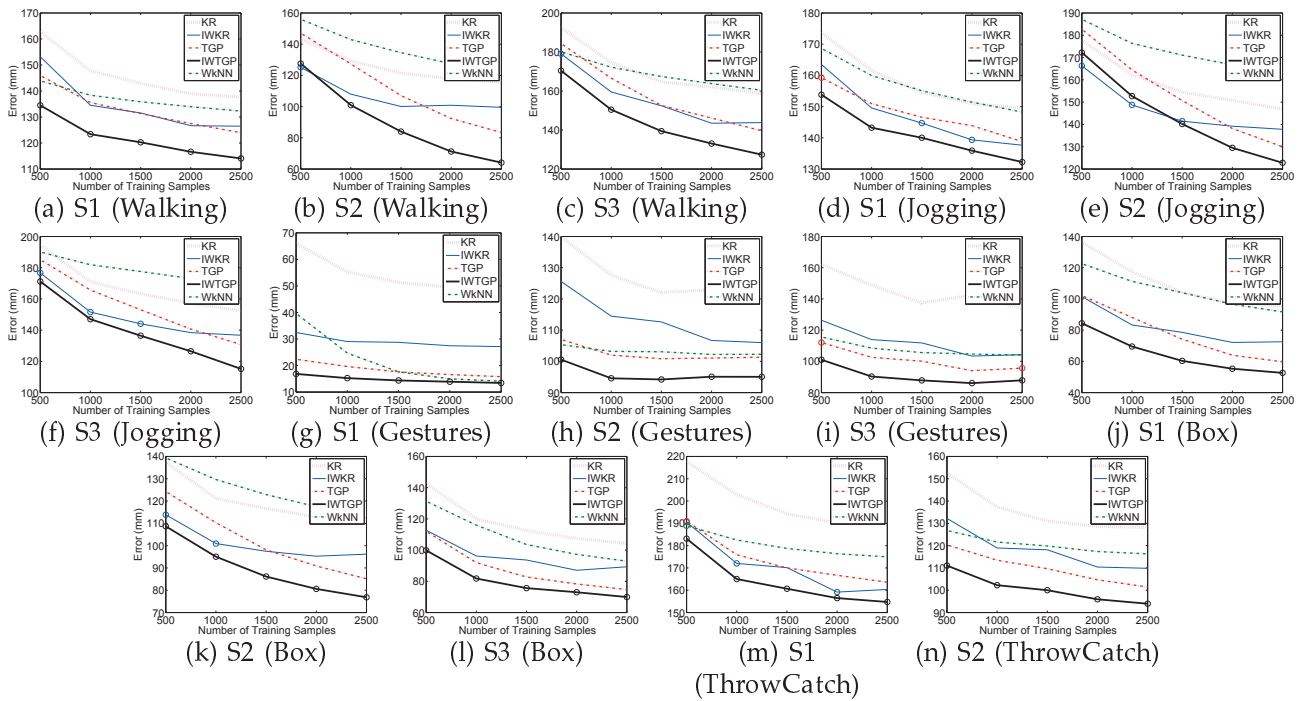

Fig. 6. **Results of selection bias (C1-3)**. Mean error for HUMANEVA-I data with respect to the number of training samples for each subject and each motion. The horizontal axis denotes the number of training samples, and the vertical axis denotes the mean error. The best method achieving the smallest mean error and comparable methods according to the paired *t-test* at the significance level 5% are specified by '○'.
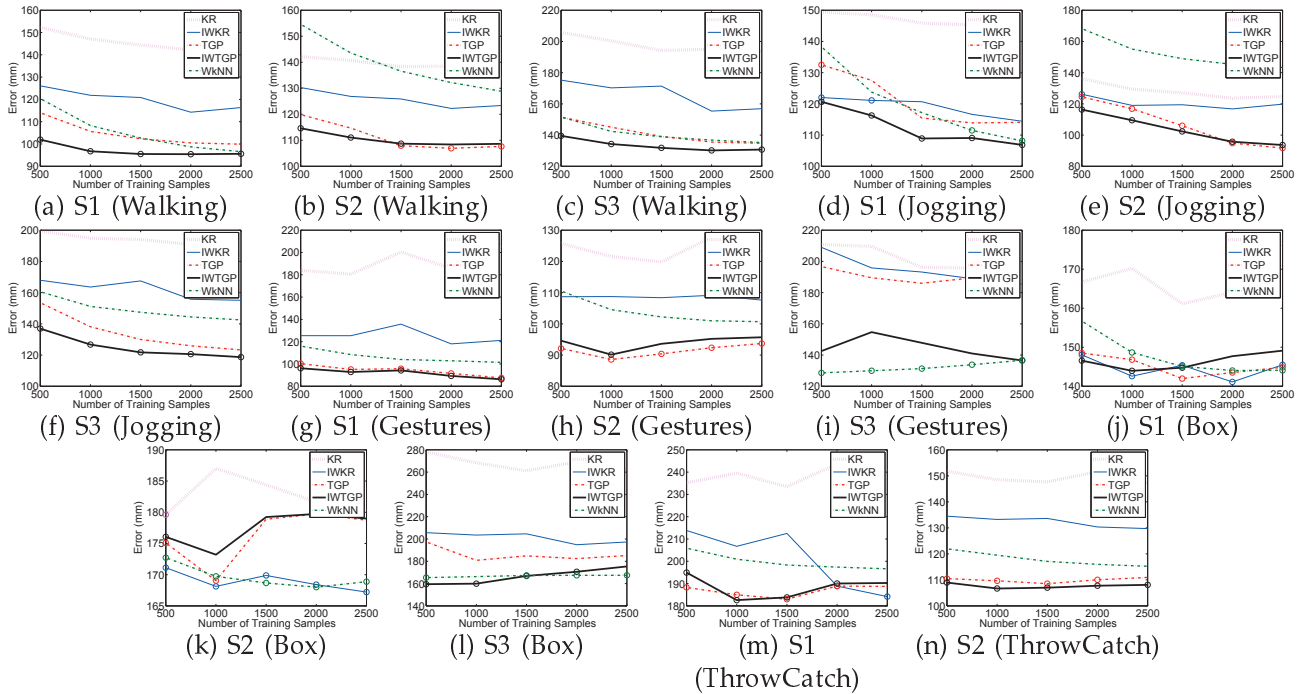

Fig. 7. **Results of subject transfer**. Mean error for HUMANEVA-I data with respect to the number of training samples for each subject and each motion. The horizontal axis denotes the number of training samples, and the vertical axis denotes the mean error. The best method achieving the smallest mean error and comparable methods according to the paired *t-test* at the significance level 5% are specified by '○'.

in *ICCV*, 2007.

[10] R. Rosales and S.Sclaroff, "Learning body pose via specialized maps," in *NIPS*, 2002.

[11] M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell, "Factorized orthogonal latent spaces," in *AISTATS*, 2010.

[12] L. Sigal, R. Memisevic, and D. Fleet, "Shared kernel informa-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. XX, NO. X, XXXX   14

tion embedding for discriminative inference," in *CVPR*, 2009.

[13] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *ICCV*, vol. 2, 2003, pp. 750—757.

[14] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Discriminative density propagation for 3D human motion estimation," in *CVPR*, 2005.

[15] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Learning joint top-down and bottom-up processes for 3D visual inference," in *CVPR*, 2006.

[16] R. Urtasun and T. Darrell, "Sparse probabilistic regression for activity-independent human pose inference," in *CVPR*, 2008.

[17] X. Zhao, H. Ning, Y. Liu, and T. S. Huang, "Discriminative estimation of 3D human pose using gaussian processes," in *CVPR*, 2008.

[18] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *ICCV*, 2011.

[19] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *CVPR*, 2011.

[20] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010.

[21] M. Stark, M. Goesele, and B. Schiele, "A shape-based object class model for knowledge transfer," in *ICCV*, 2009.

[22] A. Torralba and A. Efros, "Ubiased look at dataset bias," in *CVPR*, 2011.

[23] L. Sigal and M. J. Black, "Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion," in *TR CS-06-08, Brown Univ.*, 2006.

[24] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *ICCV*, 2011.

[25] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *CVPR*, 2012.

[26] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach," *NIPS*, 2010.

[27] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, 2000.

[28] M. Yamada, L. Sigal, and M. Raptis, "Covariate shift adaptation for discriminative 3D pose estimation," in *ECCV*, 2012.

[29] L. Sigal, A. Balan, and M. Black, "Combined discriminative and generative articulated pose and non-rigid shape estimation," 2007.

[30] T. de Campos and D. Murray, "Regression-based hand pose estimation from multiple cameras," in *CVPR*, vol. 1, 2006, pp. 782–789.

[31] R. Rosales, V. Athitsos, L. Sigal, and S. Scarloff, "3D hand pose reconstruction using specialized mappings," in *ICCV*, vol. 1, 2001, pp. 378–385.

[32] G. Fanelli, J. Gall, and L. V. Gool, "Real time head pose estimation with random regression forests," in *CVPR*, 2011.

[33] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *NIPS*, 2007.

[34] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *NIPS*, 2008.

[35] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *JMLR*, vol. 10, pp. 1391–1445, 2009.

[36] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama, "Relative density-ratio estimation for robust distribution comparison," in *NIPS*, 2011.

[37] M. Sugiyama, S. Nakajima, H. Kashima, P. Von Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," *NIPS*, pp. 1433–1440, 2008.

[38] X. Nguyen, M. Wainwright, and M. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *Information Theory, IEEE Transactions on*, vol. 56, no. 11, pp. 5847–5861, 2010.

[39] C. Cortes, Y. Mansour, and M. Mohri, "Learning bounds for importance weighting," in *NIPS*, 2010.

[40] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.

[41] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *JMLR*, vol. 8, pp. 985–1005, 2007.

Makoto Yamada received the M.S. degree in electrical engineering from Colorado State University, Fort Collins, in 2005 and the Ph.D. degree in statistical science from The Graduate Universities of Advanced Studies, Tokyo, in 2010. He has held positions as a systems engineer for Hitachi corporation from 2005 to 2007, as a researcher with Yamaha corporation from 2007 to 2010, and as a postdoctoral fellow with Tokyo Institute of Technology from 2010 to 2012. In 2012, he has held an internship appointment in computer vision and machine learning at Carnegie Mellon University and Disney Research Pittsburgh. Currently, he is a research associate at NTT Communication Science Laboratories. His research interests include machine learning and its application to signal processing and computer vision.

Leonid Sigal received the BSc degrees in computer science and mathematics from Boston University in 1999, the MA degree from Boston University in 1999, the MS degree from Brown University in 2003, and the PhD degree from Brown University in 2008. He is a research scientist at Disney Research, Pittsburgh, Pennsylvania and adjunct faculty at Carnegie Mellon University. From 1999 to 2001, he worked as a senior vision engineer at Cognex Corporation. He was a postdoctoral fellow in the Department of Computer Science at the University of Toronto from 2007 to 2009. His research interests include computer vision, visual perception, machine learning, and character animation. He has published numerous research articles and book chapters in top venues within these fields, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, CVPR, ICCV, ECCV, NIPS, and ACM Siggraph. The topics of published articles include human pose estimation and tracking, human motion perception, action recognition, latent variable models, transfer learning, physics-based models, fast approximate cloth simulation, and motion-capture. His work received the Best Paper Awards at the Articulate Motion and Deformable Objects Conference in 2006 (with Prof. Michael J. Black) and in 2012 (with Micha Andriluka). He maintains an active professional service within the community. As part of that service, he regularly serves on program committees and reviews papers for major computer vision, machine learning, and computer graphics conferences, and has organized three workshops and two tutorials (in conjunction with NIPS, CVPR, ECCV, and ICCV). He has also coedited the book Guide to Visual Analytics of Humans: Looking at People (Springer, 2011). He is a member of the IEEE.

Michalis Raptis received his Diploma in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 2006, and the M.Sc and Ph.D degrees from University of California, Los Angeles, USA, in 2008 and 2011 respectively. As of 2011 he is a postdoctoral research associate at Disney Research, Pittsburgh. His research interests are in the broader fields of computer vision, pattern recognition and time series analysis, in particular the application of discriminative approaches to the solution of event analysis tasks.