Manifold-Aware Self-Training for Unsupervised Domain Adaptation on Regressing 6D Object Pose

Yichen Zhang¹, Jiehong Lin¹, Ke Chen¹,²,*, Zelin Xu¹, Yaowei Wang² and Kui Jia¹,²,*

¹South China University of Technology
²Peng Cheng Laboratory

{eezc, lin.jiehong, eexuzelin}@mail.scut.edu.cn, {chenk, kuijia}@scut.edu.cn, wangyw@pcl.ac.cn

Abstract

Domain gap between synthetic and real data in visual regression (e.g. 6D pose estimation) is bridged in this paper via global feature alignment and local refinement on the coarse classification of discretized anchor classes in target space, which imposes a piece-wise target manifold regularization into domain-invariant representation learning. Specifically, our method incorporates an explicit self-supervised manifold regularization, revealing consistent cumulative target dependency across domains, to a self-training scheme (e.g. the popular Self-Paced Self-Training) to encourage more discriminative transferable representations of regression tasks. Moreover, learning unified implicit neural functions to estimate relative direction and distance of targets to their nearest class bins aims to refine target classification predictions, which can gain robust performance against inconsistent feature scaling sensitive to UDA regressors. Experiment results on three public benchmarks of the challenging 6D pose estimation task can verify the effectiveness of our method, consistently achieving superior performance to the state-of-the-art for UDA on 6D pose estimation. Code is available at https://github.com/Gorilla-Lab-SCUT/MAST.

1 Introduction

The problems of visual regression such as estimation of 6D pose of object instances (i.e. their orientation and translation with respect to the camera optical center) and configuration of human body parts given RGB images are widely encountered in numerous fields such as robotics [Collet et al., 2011; He et al., 2020], augmented reality [Marchand et al., 2015] and autonomous driving [Geiger et al., 2012; Chen et al., 2017], which can be typically addressed by learning a single or multi-output regression mapping on deep representations of visual observations. Recent regression algorithms have gained remarkable success to handle with inconsistent lighting conditions and heavy occlusions in-between foreground and contextual objects in uncontrolled and cluttered environment, owing to recent development of representation learning of visual regression, such as introduction of self-supervised regularization [Lin et al., 2022] and powerful network architectures [He et al., 2021].

In those regression tasks, visual observations, i.e. RGB images, can be easily acquired in practice or directly collected from the Internet, but it is laborious or even unfeasible for manual noise-free annotation with continuous targets. As a result, the size of real training data with precise labels is typically limited and less scalable, e.g. eggbox and holepuncher training samples in the LineMOD [Hinterstoisser et al., 2012] for 6D pose estimation, which increases

*Corresponding authors.
the difficulty of learning good representations. The synthesis of images can be a powerful solution to cope with data sparsity, which can be gained via photorealistic rendering [Hodaň et al., 2019] with CAD models. However, domain discrepancy between synthetic and real data, e.g., appearance difference between CAD models and real objects, scene illumination, and systematic imaging noises, can lead to collapse of regression performance, which encourages the practical setting of unsupervised domain adaptation on visual regression (UDAVR), i.e., samples in the source and target domains cannot satisfy the i.i.d. condition.

Different from the widely-investigated problem of unsupervised domain adaptation on visual classification (UDAVC) [Gopalan et al., 2011; Zou et al., 2018; Zou et al., 2021], only a few works [Chen et al., 2021; Lee et al., 2022] have explored the vital factors of representation learning of visual regression that differ from classification in the context of UDA. [Chen et al., 2021] revealed and exploited the sensitivity of feature scaling on domain adaptation regression performance to regularize representation learning, which can achieve promising results to bridge domain gap. We argue that cumulative dependent nature and piece-wise manifolds in target space are two key factors of UDA regression yet missing in the existing algorithms. To this end, this paper proposes a Manifold-Aware Self-Training (MAST) scheme to decompose the problem of learning a domain-invariant regression mapping into a combination of a feature-scaling-robust globally coarse classification of discretized target anchors via self-training based feature alignment and a locally regression-based refinement less sensitive to inconsistent feature scale, as shown in Figure 1.

For exploiting the cumulative dependent nature of regression targets different from those in classification, the self-training method (e.g., the self-paced self-training [Zou et al., 2018]) originally designed for the UDAVC problem is now adapted to the coarse classification on discretization of continuous target space, with incorporating a novel piece-wise manifold regularization on domain-invariant representation learning, namely a self-supervised cumulative target correlation regularization. Intuitively, appearance ambiguities across domains in representation learning can be mitigated via leveraging consistent target correlation under certain distance metrics in target space (e.g., the Euclidean distance in $R(3)$ translation space). Furthermore, considering the risk of sensitivity to varying feature scaling in the UDAVR problem [Chen et al., 2021], learning unified local regression functions with those shared features of the classification of discretized target bins (typically having inconsistent feature scales) can achieve superior robustness against large scale variations of transferable representations. Extensive experiments on three popular benchmarks of the challenging UDA on 6D pose estimation can confirm the effectiveness of our MAST scheme, consistently outperforming the state-of-the-art.

The novelties of our paper are summarized as follows.

- Technically, a novel cumulative target correlation regularization is proposed to regularize the self-training algorithm on coarse classification with latent dependency across regression targets, while local refinement can be achieved via learning implicit functions to estimate residual distance to the nearest anchor within local target manifolds in a unified algorithm.
- Experiment results on multiple public benchmarks of UDA on 6D pose estimation can verify consistent superior performance of our scheme to the state-of-the-art UDA pose regressors.

2 Related Works

6D Pose Estimation. The problem of estimating 6D pose of object instances within an RGB image (optionally with a complementary depth image) is active yet challenging in robotics and computer vision. With the rise of deep learning, recent methods for predicting 6D poses can be divided into two main groups – keypoint-based [Peng et al., 2019; Zakharov et al., 2019] and regression based [Xiang et al., 2018; Labbé et al., 2020; Wang et al., 2021b]. The former relied on learning a 2D-to-3D correspondence mapping between object keypoints in 3D space and their 2D projection on images with the Perspective-n-Point (PnP) [Fischler and Bolles, 1981]. Such a correspondence can be achieved by either detecting a limited size of landmarks [Tekin et al., 2018; Peng et al., 2019] or pixel-wise voting from a heatmap [Park et al., 2019; Zakharov et al., 2019]. The latter concerned on deep representation learning for direct pose regression with the point-matching loss for optimizing output pose [Xiang et al., 2018; Labbé et al., 2020] or proposing a differentiable PnP paradigm in an end-to-end training style [Wang et al., 2021b; Chen et al., 2022a]. Alternatively, the problem can also be formulated into ordinal classification via discretization of $SE(3)$ space into class bins [Su et al., 2015; Kehl et al., 2017]. To alleviate representation ambiguities, the estimated 6D pose of objects can be further refined via either an iterative refinement with residual learning [Li et al., 2018b; Manhardt et al., 2018] or simply the Iterative Closest Point [Xiang et al., 2018], while some work introduced cross-view fusion based refinement [Labbé et al., 2020; Li et al., 2018a]. Existing refinement strategies are typically employed as a post-processing step following the main module of 6D pose estimation, some of which such as [Li et al., 2018b; Xu et al., 2022] can be designed in an end-to-end learning cascade to obtain significant performance gain, but they are not designed for bridging domain gap and therefore cannot ensure good performance under the UDAVR setting. Alternatively, [Li et al., 2018a] introduced a combined scheme of both coarse classification and local regression-based refinement simultaneously, which is similar to our MAST method. However, the main differences lie in the introduction of the cumulative target correlation regularization in our scheme to encourage domain-invariant pose representations revealing the dependent nature of regression targets.

Unsupervised Domain Adaptation on Visual Regression. Most of regression methods [Xu et al., 2019; Bao et al., 2022] employ annotated real data for model training, but
manual annotations on real data are usually laboriously expensive or even unfeasible. Lack of sufficient annotated real data encourages the practical setting of Simulation-to-Reality (Sim2Real) UDAVR, i.e., learning a domain-agnostic representation given annotated synthetic data as source domain and unlabeled real data as target domain. A simple yet effective way to narrow Sim2Real domain gap can rely on domain randomization [Kehl et al., 2017; Manhardt et al., 2018], while recent success of self-supervised learning for UDAVC [Zou et al., 2021; Yue et al., 2021] inspired a number of self-supervised regressors [Wang et al., 2021a; Yang et al., 2021] in the context of Regression. Self6D [Wang et al., 2020] and its extension Self6D++ [Wang et al., 2021a] leveraged a differentiable renderer to conduct self-supervised visual and geometrical alignment on visible and amodal object mask predictions. Bao et al. [Bao et al., 2022] introduced a self-supervised representation learning of relative rotation estimation to adapt one gaze regressor to the target domain. Zhang et al. [Zhang et al., 2021] utilized a Graph Convolutional Network to model domain-invariant geometry structure among key-points, which is applied to guide training of the object pose estimator on real images. These mentioned algorithms were designed for only one specific task and cannot be directly applied to other visual regression problems. [Chen et al., 2021] proposed the representation subspace distance (RSD) generic to multiple UDA VR problems, but cannot perform well on the challenging task having severe representation ambiguities, e.g., 6D pose estimation investigated in this paper (see Table 3). In contrast, the proposed MAST scheme is generic to UDA VR owing to exploiting explicit target correlation in the style of local manifolds to regularize deep representation learning agnostic to domains.

Self-Training. Self-training methods utilize a trained model on labeled data to make predictions of unannotated data as pseudo labels [Lee and others, 2013; Sohn et al., 2020] and UDA [Roy et al., 2019]. [Sohn et al., 2020] generated pseudo labels from weakly augmented images, which are adopted as supervision of strongly augmented variants in semi-supervised learning; similar scripts are shared with the noisy student training [Xie et al., 2020]. [Chen et al., 2011] proposed the co-training for domain adaptation that slowly adding to the training set both the target features and instances in which the current algorithm is the most confident. [Zou et al., 2018] proposed the self-paced self-training (SPST) for unsupervised domain adaptation classification that can perform a self-paced learning [Tang et al., 2012] with latent variable objective optimization. The representative SPST has inspired a number of follow-uppers such as [Zou et al., 2021] and [Chen et al., 2022b]. Nevertheless, all of existing self-training algorithms were designed for classification or segmentation, while self-training for the UDAVR remains a promising yet less explored direction.

3 Methodology

Given a source domain \( \{ \mathcal{Y}^s_i, \mathcal{X}^s_i \}_{i=1}^{N_s} \) with \( N_s \) labeled samples and a target domain \( \{ \mathcal{Y}^t_i, \mathcal{X}^t_i \}_{i=1}^{N_t} \) with \( N_t \) unlabeled samples, tasks of UDAVR aim at learning a domain-invariant regression mapping to a shared continuous label space \( Y \). In the context of our focused 6D object pose estimation, the source and target domains are often the synthetic and real-world data, respectively, while the shared label space between two domains is the whole learning space of \( SE(3) \).

To deal with the problems of UDAVR introduced in Sec. 1, e.g., cumulative dependent nature and piece-wise manifolds in target space, we propose in this paper a manifold-aware self-training scheme, which decomposes the learning of \( SE(3) \) space into a global classification on discretized pose anchors and a local pose refinement for feature scaling robustness, and incorporates a self-supervised manifold regularization to the self-training.

3.1 The Design of Network Architecture

Given a batch of \( B \) object-centric RGB images \( \{ \mathcal{X}^s_i \}_{i=1}^{B} \) as input, the proposed scheme is designed to predict 6D object poses \( \{ \mathcal{Y}^t_i \}_{i=1}^{B} \), with each pose \( \mathcal{T} = [\mathbf{R}, \mathbf{t}] \) represented by a 3D rotation \( \mathbf{R} \in SO(3) \) and a 3D translation \( \mathbf{t} \in \mathbb{R}^3 \). The whole network architecture is shown in Fig. 2, which consists of three main modules, including a Feature Extractor, a Coarse Classifier of discretized pose anchors, and a Fine Regressor of Residual Poses to the nearest anchor.

More specifically, we employ the same feature extractor as [Labbé et al., 2020] to learn the pose-sensitive feature vectors \( \{ \mathbf{f}^b \in \mathbb{R}^{C_i} \}_{i=1}^{B} \) from each frame, which are then fed into the decoupled coarse classifier and fine regressor individually, whose output are combined together as final pose predictions. The former learns coarse poses via classification on the discretized pose anchors, while the latter learns residual poses to refine the coarse ones of pose anchors locally; both modules share the same input features, achieving superior robustness against inconsistent feature scaling. We will take a single image as an example to detail the two pose estimation modules shortly, and thus omit the superscript \( b \) of the notations for simplicity in the following subsections.

Coarse Classification on Discretized Pose Anchors. Given the pose-sensitive feature \( \mathbf{f} \) of \( \mathcal{I} \), the goal of this module is to globally make coarse predictions of \( \mathbf{R} \) and \( \mathbf{t} \) via classification on their pre-defined anchors, respectively. For the rotation \( \mathbf{R} \), we generate \( N_R \) anchors that are uniformly distributed on the whole \( SO(3) \) space as [Li et al., 2018a], which are denoted as \( \{ \mathbf{R}^1, \cdots, \mathbf{R}^{N_R} \} \). For the translation \( \mathbf{t} \), we factorize it into three individual classification targets, including the two translation components \( v_x \) and \( v_y \) on the image coordinate system along X-axis and Y-axis, with the remaining component \( z \) along Z-axis; for each classification target \( t \in \{ v_x, v_y, z \} \), we discretize the range of \( [d_{\min}^t, d_{\max}^t] \) into \( N_t \) bins uniformly, and use the bin centers \( \{ t_{a}^1, \cdots, t_{a}^N \} \) as the anchors of \( t \). We implement the classifier as four Multilayer Perceptrons (MLPs) with \( N_R, N_{v_x}, N_{v_y}, N_z \) output neurons, which are collectively denoted as the probabilities \( \mathbf{S}_R \in \mathbb{R}^{N_R}, \mathbf{S}_{v_x} \in \mathbb{R}^{N_{v_x}}, \mathbf{S}_{v_y} \in \mathbb{R}^{N_{v_y}}, \) and \( \mathbf{S}_z \in \mathbb{R}^{N_z} \) of \( \mathbf{R}, \mathbf{v}_x, \mathbf{v}_y, \) and \( z \), respectively. Denoting their indexes of maximum probabilities as \( t_{a}^{\max}_R, t_{a}^{\max}_{v_x}, t_{a}^{\max}_{v_y}, \) and \( t_{a}^{\max}_z \), the classifier finally gives out their coarse pose predictions as \( \mathbf{R}_{cls} = \mathbf{R}^{t_{a}^{\max}_R}, \mathbf{v}_{x,cls} = v_{x,a}^{t_{a}^{\max}_{v_x}}, \mathbf{v}_{y,cls} = v_{y,a}^{t_{a}^{\max}_{v_y}}, \) and \( z_{cls} = z_{a}^{t_{a}^{\max}_z} \).
Fine Regressor of Residual Poses. This module shares the same input feature \( f \) as the coarse classifier to make the learning more robust to feature scale variations, and is implemented as four MLPs with \( N_R \times 6, N_{v_x}, N_{v_y}, N_z \) output neurons to regress the residuals of the pose anchors. We collectively denote the outputs as \( \{ R_{reg,6D}^t \}_{i=1}^{N_R} \), \( \{ v_{x,reg}^i \}_{i=1}^{N_{v_x}} \), \( \{ v_{y,reg}^i \}_{i=1}^{N_{v_y}} \), and \( \{ z_{reg}^i \}_{i=1}^{N_z} \); here we use the continuous 6D representations of rotation [Zhou et al., 2019] as the regression target, which can be transformed into rotation matrices \( \{ R_{reg}^t \}_{i=1}^{N_R} \). According to probabilities of the classifier, the fine regressor refines the coarse predictions via the residuals \( R_{reg}^t = \hat{R}_{reg}^t \), \( v_{x,reg}^i = v_{x,reg}^i \), \( v_{y,reg}^i = v_{y,reg}^i \), and \( z_{reg} = z_{reg}^i \).

Combining coarse anchor predictions and their residuals, our proposed network can generate the final object pose \( T = [R|\tilde{t}] \), with \( t = [x, y, z] \), as follows:

\[
\begin{align*}
R &= R_{reg} \cdot R_{cls} \\
\hat{x} &= (v_{x,cls} + v_{x,reg}) \cdot z / f_x \\
\hat{y} &= (v_{y,cls} + v_{y,reg}) \cdot z / f_y \\
\hat{z} &= z_{cls} + z_{reg}
\end{align*}
\]

(1)

where \( f_x \) and \( f_y \) are the focal lengths along X-axis and Y-axis, respectively.

3.2 Manifold-Aware Objective
To train our network, we formulate the following manifold-aware objective \( \mathcal{L} \) via combining a \textit{coarse-to-fine pose decomposition loss} \( \mathcal{L}_{pose} \) with a \textit{cumulative target correlation regularization} \( \mathcal{L}_{ctc} \):

\[
\mathcal{L} = \mathcal{L}_{pose} + \mathcal{L}_{ctc},
\]

(2)

where \( \mathcal{L}_{pose} \) favors for domain-invariant representations in 6D pose estimation across domains, while \( \mathcal{L}_{ctc} \) enforces target manifolds into representation learning.

Coarse-to-fine Pose Decomposition Loss. \( \mathcal{L}_{pose} \) consists of two loss terms \( \mathcal{L}_{cls} \) and \( \mathcal{L}_{reg} \) for the coarse classifier and the fine regressor, respectively, as follows:

\[
\mathcal{L}_{pose} = \frac{1}{B} \sum_{b=1}^{B} \mathcal{L}_{cls}^b + \mathcal{L}_{reg}^b.
\]

(3)

For simplicity, we introduce \( \mathcal{L}_{pose} \) on single input, and thus omit the batch index \( b \) accordingly.

For the coarse classifier, given the ground truth pose \( \hat{T} = [R|\hat{t}] \), with \( \hat{t} = [\hat{x}, \hat{y}, \hat{z}] \) (and \( \hat{v}_x, \hat{v}_y \)), we first adopt a sparse scoring strategy to assign the labels for \( S_R, S_{v_x}, S_{v_y} \) and \( S_z \), resulting in \( S_R, \hat{S}_{v_x}, \hat{S}_{v_y} \) and \( \hat{S}_z \), respectively, with each element \( \hat{s}_t \) \( (t \in \{R, v_x, v_y, z\}) \) assigned as follows:

\[
\hat{s}_t = \begin{cases} 
\theta_{t,1}, & i \in NN_{1}(\hat{t}) \\
\theta_{t,2}, & i \in NN_{k_i}(\hat{t}) \setminus \text{NN}_{1}(\hat{t}) \\
0, & \text{Otherwise}
\end{cases}
\]

(4)

where \( \theta_{t,1} \gg \theta_{t,2} \), and \( \theta_{t,1} + (k_i - 1) \theta_{t,2} = 1 \). \text{NN}_{k_i}(\hat{t}) \) denotes the set of indexes of the \( k_i \) nearest anchors of \( \hat{t} \). With the assigned labels, we use the cross-entropy loss \( \mathcal{H} \) on top of the classifier as follows:

\[
\mathcal{L}_{cls} = \sum_{t \in \{R,v_x,v_y,z\}} \mathcal{H}(s_t, \hat{s}_t).
\]

(5)

For the fine regressor, we make individual predictions on each anchor of \( t \in \{R, (v_x, v_y), z\} \) by combining the paired classification and regression results, and supervise the predic-
tions of their top K nearest anchors of \( \tilde{t} \) as follows:

\[
L_{reg} = \sum_{i \in NN_{K}(\tilde{R})} D(T_{Ri}, \tilde{T}) + \sum_{i \in NN_{K}(\tilde{x})} D(T_{zi}, \tilde{T}) + \sum_{i \in NN_{2}x_{v}(\tilde{v}, \tilde{v}_{y})} D(T_{v_{i}v_{j}}, \tilde{T}),
\]

where \( t' \) denotes the prediction of the anchor \( i \) of \( t \), and \( \tilde{T}_{t'} \) denotes the object pose computed by \( t' \) and other ground truths \( \{ \tilde{R}, (\tilde{v}, \tilde{v}_{y}), \tilde{x} \} \). \( D(\cdot, \cdot) \) is the \( L_{1} \) distance between the point sets transformed by two object poses from the same object point cloud \( O \), as follows:

\[
D(T, \tilde{T}) = \frac{1}{|O|} \sum_{x \in O} \| T x - \tilde{T} x \|_{1}.
\]

Following [Labbé et al., 2020], we combine the supervision of \( v_{x} \) and \( v_{y} \) for convenience in (6), and also employ the same strategy to handle object symmetries by finding the closest ground truth rotation to the predicted one.

Cumulative Target Correlation Regularization. For regression tasks, continuous targets preserve latent cumulative dependency [Chen et al., 2013]. When we discretize the continuously changing targets into discretized labels as classification, the assumption of independence across targets is adopted, which is invalid in regressing continuous targets. As a result, each class cannot seek support from samples of correlated class, which can significantly reduce performance especially for sparse and imbalanced data distributions. To better cope with this problem, we propose to regularize the features by an explicit relation in the regression target space.

Given the pose-sensitive feature vectors \( \{ f^{b} \in \mathbb{R}^{C} \}_{b=1}^{B} \) of a mini-batch inputs \( \{ T^{b} \}_{b=1}^{B} \), we first build the feature correlation graph \( G \in \mathbb{R}^{B \times B} \) across the data batch via feature cosine similarities, with the element \( g^{ij} \) indexed by \( (i, j) \) computed as follows:

\[
g^{ij} = \frac{f^{i} \cdot f^{j}}{||f^{i}||_{2} \cdot ||f^{j}||_{2}},
\]

where \( \cdot \cdot \cdot \) denotes inner product. Then we build the ground truth \( \tilde{G} \) based on a pre-computed correlation graph \( \tilde{G}_{0} \in \mathbb{R}^{N \times N} \) with \( N \) pose classes; assuming the classes of \( T_{i} \) and \( T_{j} \) are \( n_{i} \) and \( n_{j} \), respectively, we assign the value of \( \tilde{g}^{ij} \) in \( \tilde{G} \) as that of \( \tilde{g}_{0}^{n_{i}n_{j}} \). Finally, the proposed target correlation regularizer can be simply written as the squared \( L_{2} \) distance between \( G \) and \( \tilde{G} \):

\[
\mathcal{L}_{cfc} = ||G - \tilde{G}||^{2}.
\]

There are multiple choices for building the pose-related correlation graph \( \tilde{G}_{0} \); here we introduce a simple but effective one, which utilizes the similarity of depth components of translations along Z-axis to initialize \( \tilde{G}_{0} \), with \( N = N_{z} \).

Specifically, for the anchors \( \{ z^{1}, \ldots, z^{N} \} \) of \( z \), we map them linearly to the angles \( \{ \phi^{1}, \ldots, \phi^{N} \} \) as follows:

\[
\phi^{n} = \frac{z_{n}^{n} - z_{min}}{z_{max} - z_{min}} \frac{\pi}{2},
\]

and the element \( \tilde{g}_{0}^{n_{i}n_{j}} \) of \( \tilde{G}_{0} \) indexed by \( (n_{i}, n_{j}) \) can be defined as the cosine of difference between the angles:

\[
\tilde{g}_{0}^{n_{i}n_{j}} = \cos(|\phi^{n_{i}} - \phi^{n_{j}}|).
\]

When \( z_{n_{i}} \) and \( z_{n_{j}} \) are close, the difference of their corresponding angles is small, and thus the correlation value of \( \tilde{g}_{0}^{n_{i}n_{j}} \) will be large. The reason for choosing \( z \) is that the learning of this component is very challenging in 6D pose estimation without depth information. Experimental results in Sec. 4.2 also verify the effectiveness of our regularization.

3.3 Manifold-Aware Self-training

To reduce the Sim2Real domain gap, we design a manifold-aware self-training scheme for unsupervisedly adapting the pose estimator, which adaptively incorporates our proposed manifold-aware training objective in (2) with Self-Paced Self-Training [Zou et al., 2018] to select target samples in an easy-to-hard manner. More specifically, we first train a teacher model \( M_{T} \) on the labeled synthetic data (source domain) as a pseudo-label annotator for the unlabeled real-world data (target domain), and select the training samples from the real data with pseudo labels for a learning of a student model \( M_{S} \). Both teacher and student models share the same networks introduced in Sec. 3.1, and are trained by solving the problems of \( \min_{M_{T}} L \) and \( \min_{M_{S}} L \), respectively.

The core of sample selection on the target domain lies on the qualities of pseudo labels. For the tasks of visual classification, the categorical probabilities are usually used as the measurement of qualities, while for those of visual regression tasks, e.g., object pose estimation in this paper, direct usage of the typical mean square error (MSE) can be less effective due to lack of directional constraints for adaptation. In geometric viewpoint, the surface of a super ball can have the same MSE distance to its origin, but the optimal regions of object surface for domain adaptation exist, which can be omitted by the MSE metric. Owing to the decomposition of object pose estimation into coarse classification and fine regression in our MAST scheme, we can flexibly exploit the classification scores to indicate the qualities of pseudo labels, since the coarse classification points out the overall direction of pose estimation. In practice, we use the probabilities \( S_{z} \) as confidence scores because UDA on classification can perform more stably and robustly, and set a threshold \( \tau \) to select the samples with scores larger than \( \tau \) for training \( M_{S} \). Larger score indicates higher quality of the pseudo label. Following [Zou et al., 2018], the threshold \( \tau \) is gradually decreased during training, realizing the learning in an easy-to-hard manner and making \( M_{S} \) generalized to harder target samples.

4 Experiments

Datasets and Settings. The LineMOD dataset [Hinterstoisser et al., 2012] provides individual videos of 13 textureless objects, which are recorded in cluttered scenes with challenging lighting variations. For each object, we follow [Brachmann et al., 2014] to use randomly sampled 15% of the sequence as the real-world training data of the target domain, and the remaining images are set aside for testing. The OccludedLineMOD dataset [Brachmann et al., 2014] is a subset
<table>
<thead>
<tr>
<th>Method</th>
<th>Ape</th>
<th>Bvise</th>
<th>Cam</th>
<th>Can</th>
<th>Cat</th>
<th>Drill</th>
<th>Duck</th>
<th>Eggbox</th>
<th>Glue</th>
<th>Holep</th>
<th>Iron</th>
<th>Lamp</th>
<th>Phone</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAE [Sundermeyer et al., 2020]</td>
<td>4.0</td>
<td>20.9</td>
<td>30.5</td>
<td>35.9</td>
<td>17.9</td>
<td>24.0</td>
<td>4.9</td>
<td>81.0</td>
<td>45.5</td>
<td>17.6</td>
<td>32.0</td>
<td>60.5</td>
<td>33.8</td>
<td>31.4</td>
</tr>
<tr>
<td>DSC-PoseNet [Yang et al., 2021]</td>
<td>23.4</td>
<td>75.6</td>
<td>11.7</td>
<td>40.1</td>
<td>26.7</td>
<td>53.8</td>
<td>14.0</td>
<td>73.6</td>
<td>26.7</td>
<td>19.5</td>
<td>56.2</td>
<td>39.4</td>
<td>20.0</td>
<td>37.0</td>
</tr>
<tr>
<td>MHP [Manhardt et al., 2019]</td>
<td>11.9</td>
<td>66.2</td>
<td>22.4</td>
<td>59.8</td>
<td>26.9</td>
<td>44.6</td>
<td>8.3</td>
<td>55.7</td>
<td>54.6</td>
<td>15.5</td>
<td>60.8</td>
<td>-</td>
<td>34.4</td>
<td>38.8</td>
</tr>
<tr>
<td>DPOD [Zakhvorov et al., 2019]</td>
<td>35.1</td>
<td>59.4</td>
<td>15.5</td>
<td>48.8</td>
<td>28.1</td>
<td>59.3</td>
<td>25.6</td>
<td>51.2</td>
<td>34.6</td>
<td>17.7</td>
<td>84.7</td>
<td>45.0</td>
<td>20.9</td>
<td>40.5</td>
</tr>
<tr>
<td>SD-Pose [Li et al., 2021]</td>
<td>54.0</td>
<td>76.4</td>
<td>50.2</td>
<td>81.2</td>
<td>71.0</td>
<td>64.2</td>
<td>54.0</td>
<td>93.9</td>
<td>92.6</td>
<td>24.0</td>
<td>77.0</td>
<td>82.6</td>
<td>53.7</td>
<td>67.3</td>
</tr>
<tr>
<td>Self6D++ [Wang et al., 2021a]</td>
<td>50.9</td>
<td>99.4</td>
<td>89.2</td>
<td>97.2</td>
<td>79.9</td>
<td>98.7</td>
<td>24.6</td>
<td>81.1</td>
<td>81.2</td>
<td>41.9</td>
<td>98.8</td>
<td>98.9</td>
<td>64.3</td>
<td>77.4</td>
</tr>
<tr>
<td>MAR (ours)</td>
<td>68.6</td>
<td>97.4</td>
<td>79.4</td>
<td>98.3</td>
<td>87.1</td>
<td>94.2</td>
<td>61.3</td>
<td>82.0</td>
<td>87.1</td>
<td>56.7</td>
<td>94.3</td>
<td>92.3</td>
<td>68.8</td>
<td>82.1</td>
</tr>
</tbody>
</table>

Table 1: Comparative evaluation on the LineMOD dataset w.r.t. the Average Recall (%) of ADD(-S). Symmetric object classes are in italic. ‘MAR’ (manifold-aware regression) denotes our method without self-training.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ape</th>
<th>Bvise</th>
<th>Cam</th>
<th>Can</th>
<th>Cat</th>
<th>Drill</th>
<th>Duck</th>
<th>Eggbox</th>
<th>Glue</th>
<th>Holep</th>
<th>Iron</th>
<th>Lamp</th>
<th>Phone</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPOD [Zakhvorov et al., 2019]</td>
<td>53.3</td>
<td>95.2</td>
<td>90.0</td>
<td>94.1</td>
<td>60.4</td>
<td>97.6</td>
<td>60.0</td>
<td>99.6</td>
<td>93.8</td>
<td>64.9</td>
<td>99.8</td>
<td>88.1</td>
<td>71.4</td>
<td>82.6</td>
</tr>
<tr>
<td>DSC-PoseNet [Yang et al., 2021]</td>
<td>59.2</td>
<td>98.1</td>
<td>88.0</td>
<td>92.1</td>
<td>79.4</td>
<td>94.5</td>
<td>51.7</td>
<td>98.5</td>
<td>93.9</td>
<td>78.4</td>
<td>96.2</td>
<td>96.3</td>
<td>90.0</td>
<td>85.9</td>
</tr>
<tr>
<td>Self6D++ [Wang et al., 2021a]</td>
<td>85.0</td>
<td>99.8</td>
<td>96.5</td>
<td>99.3</td>
<td>93.0</td>
<td>100.0</td>
<td>65.3</td>
<td>99.9</td>
<td>98.1</td>
<td>73.4</td>
<td>86.9</td>
<td>99.6</td>
<td>86.3</td>
<td>91.0</td>
</tr>
<tr>
<td>MAR (ours)</td>
<td>81.4</td>
<td>99.9</td>
<td>90.7</td>
<td>99.6</td>
<td>94.6</td>
<td>98.1</td>
<td>85.5</td>
<td>97.6</td>
<td>98.5</td>
<td>89.2</td>
<td>97.1</td>
<td>99.7</td>
<td>96.0</td>
<td>94.5</td>
</tr>
</tbody>
</table>

Table 2: Comparative evaluation on the Occluded LineMOD and HomebrewedDB datasets w.r.t. the Average Recall (%) of the ADD(-S). Symmetric object classes are in italic. ‘MAR’ (manifold-aware regression) denotes our method without self-training.

Table of the LineMOD with 8 different objects, which is formed by the images with severe object occlusions and self-occlusions. We follow [Wang et al., 2021a] to split the training and test sets. The HomebrewedDB dataset [Kaskman et al., 2019] provides newly captured test images of three objects in the LineMOD, including bvise, driller, and phone. Following the Self-6D [Wang et al., 2020], the second sequence of HomebrewedDB is used to test our models which are trained on the LineMOD, to evaluate the robustness of our method on different variations, e.g., scene layouts and camera intrinsics. In the experiments, the above three real-world datasets are considered as the target domains, all of which share the same synthetic source domain. We employ the publicly available synthetic data provided by BOP challenge [Hodaň et al., 2020] as the source data, which contains 50k images generated by physically-based rendering (PBR) [Hodaň et al., 2019].

**Evaluation Metrics.** Following [Wang et al., 2020], we employ the Average Distance of model points (ADD) [Hinterstoisser et al., 2012] as the evaluation metric of the 6D poses for asymmetric objects, which measures the average deviation of the model point set $\mathcal{O}$ transformed by the estimated pose $\hat{T} = [\hat{R} | \hat{t}]$ and that transformed by the ground-truth pose $\bar{T} = [\bar{R} | \bar{t}]$:

$$D_{ADD}(\mathcal{T}, \hat{T}) = \frac{1}{|\mathcal{O}|} \sum_{x \in \mathcal{O}} ||(Rx + t) - (\hat{R}x + \hat{t})||_2.$$  (12)

For symmetric objects, we employ the metric of Average Distance of the closest points (ADD-S) [Hodaň et al., 2016],

$$D_{ADD-S}(\mathcal{T}, \hat{T}) = \frac{1}{|\mathcal{O}|} \sum_{x \in \mathcal{O}} \min_{x_1, x_2 \in \mathcal{O}} ||(Rx_1 + t) - (\hat{R}x_2 + \hat{t})||_2.$$  (13)

Combining (12) and (13), we report the Average Recall (%) of ADD(S) less than 10% of the object’s diameter on all the three datasets.

**Implementation Details.** For object detection, we use Mask R-CNN [He et al., 2017] trained purely on synthetic PBR images to generate the object bounding boxes for the target real data. For pose estimation, we set the numbers of anchors as $N_{Rx} = 60, N_{Rx} = 20, N_{Rx} = 40$, and set

Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
the ranges of \(v_z, v_y\) and \(z\) as \([d_{v_z}, d_{v_y}], [d_{v_z}, d_{v_y}] = [-200, 200], \) and \([d_{v_z}, d_{v_y}] = [0.0, 2.0], \) respectively. To train our network, we choose \(\theta_1^R = 0.7, \theta_2^R = 0.1\) and \(k_R = 4\) for rotation in (4), and also set \(\theta_1^v = \theta_1^v = \theta_1^v = 0.55, \theta_2^v = \theta_2^v = \theta_2^v = 0.075,\) and \(k_v = k_v = k_v = 7\) for translation. Following the popular setting [Wang et al., 2021a], we train individual networks for all the objects with the Adam optimizer [Kingma and Ba, 2014]. The teacher model \(M_T\) is firstly pre-trained on the synthetic images of all objects, and then fine-tuned on the single object, while the parameters of the student model \(M_S\) is initialized as those of \(M_T\); their initial learning rates are \(3 \times 10^{-4}\) and \(3 \times 10^{-5}\), respectively. The training batch size is set as \(B = 32\). We also include the same data augmentation as [Labbe et al., 2020] during training.

4.1 Comparative Evaluation

We compare our method with the existing ones on three benchmarks for 6D object pose estimation with RGB images.

On the LineMOD, we conduct experiments under three settings of training data, including 1) labeled synthetic data, 2) labeled synthetic and real data, and 3) labeled synthetic data and unlabeled real data. Results of the first two settings are the lower and upper bounds of that of the last setting. We report qualitative results of comparative methods in Table 1, where our method outperforms its competitors by large margins under all the settings, e.g., with the respective improvements of 4.7%, 5.5% and 1.6% over the state-of-the-art Self6D++ [Wang et al., 2021a]. On the Occluded LineMOD and the HomebrewedDB, results are shown in Table 2, where our method consistently performs better than the existing ones on both datasets, demonstrating superior robustness of our method against occlusion and the generalization to new scenes and cameras.

4.2 Ablation Studies and Analyses

Effects of Decomposing into Coarse Classification and Fine Regression. We decompose the problem of UDA on estimating object poses into a coarse classification on discretized anchors and a residual regression. As shown in Table 3, for the models trained purely on synthetic data, the design of pose decomposition realizes 4.0% and 5.7% improvements on the LineMOD and the Occluded LineMOD, respectively, compared to direct regression of object poses, since the global classification eases the difficulty in learning along with feature-scaling robustness, and the local regression achieves pose refinement.

Effects of Cumulative Target Correlation Regularization. As shown in Table 3, \(L_{ctc}\) consistently improves the results under different settings across different datasets, e.g., 5.6% improvement on the Occluded LineMOD for the model trained on synthetic data, which demonstrates the effectiveness of \(L_{ctc}\) on mining latent correlation across regression targets. We also visualize the feature distribution of an example via the t-SNE [Van der Maaten and Hinton, 2008] in Fig. 1, where, with \(L_{ctc}\), features assigned to different pose anchors preserve smooth and continuously changing nature of regression targets in the feature space.

<table>
<thead>
<tr>
<th>Pose Estimator</th>
<th>(L_{ctc})</th>
<th>Method of UDA</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg.</td>
<td>×</td>
<td>-</td>
<td>LM</td>
</tr>
<tr>
<td>Cls. + Reg.</td>
<td>×</td>
<td>-</td>
<td>LM</td>
</tr>
<tr>
<td>Cls. + Reg.</td>
<td>√</td>
<td>Self-Training</td>
<td>LMO</td>
</tr>
</tbody>
</table>

Table 3: Ablation studies on LineMOD (LM) and Occluded LineMOD (LMO) datasets w.r.t. the Average Recall (%) of ADD(-S). ‘Reg.’ denotes direct regression of object poses, while ‘Cls. + Reg.’ denotes the combined use of coarse classification and fine regressor for pose estimation.

Effects of Manifold-Aware Self-Training on Coarse Classification. The self-training schemes have been verified their effectiveness on reducing the Sim2Real domain gap by incorporating the unlabeled real data into training via pseudo label generation and training sample selection. Taking our network with \(L_{ctc}\) as example, the results are improved from 55.3% to 61.4% on the Occluded LineMOD via self-training. Compared to the RSD [Chen et al., 2021] designed for the problem of UDA on regression, our MAST scheme can significantly beat the competing RSD (see results in Table 3), where the only difference lies in replacing self-training on coarse classification with RSD on whole regression. Such an observation can again confirm the superiority of the proposed MAST scheme, consistently outperforming the state-of-the-art UDA on regression.

5 Conclusion

This paper proposes a novel and generic manifold-aware self-training scheme for UDA on regression, which is applied to the challenging 6D pose estimation of object instances. We address the UDADV problem via decomposing it into coarse classification and fine regression, together with a cumulative target correlation regularization. Experiment results on three popular benchmarks can verify the effectiveness of our MAST scheme, outperforming the state-of-the-art methods with significant margins. It is worth pointing out that our MAST scheme can readily be applied to any UDA regression tasks, as the UDA on coarse classification making our method robust against feature scaling while maintaining latent cumulative correlation underlying in regression target space.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant No.: 61902131), the Guangdong Youth Talent Program (Grant No.: 2019QN01X246), the Guangdong Basic and Applied Basic Research Foundation (Grant No.: 2022A1515011549), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No.: 2017ZT07X183), and the Guangdong Provincial Key Laboratory of Human Digital Twin (Grant No.: 2022B1212010004).
References


Augmented autoencoders: Implicit 3D orientation learning
Csaba Marton, Maximilian Durner, and Rudolph Triebel.
ICCV views. In

Li Fei-Fei, and Daphne Koller. Shifting weights: Whitening and consensus loss.
Peng et al., 2019

Roy et al., 2019

Sock et al., 2020

Sohn et al., 2020

Su et al., 2015
Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3D model views.
ICCV, 2015.

Sundermeyer et al., 2020

Tang et al., 2012
Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. Shifting weights: Adapting object detectors from image to video.

Tekin et al., 2018
CVPR, 2018.

Van der Maaten and Hinton, 2008

Wang et al., 2020
In ECCV, 2020.

Wang et al., 2021a

Wang et al., 2021b
In CVPR, 2021.

Xiang et al., 2018
RSS, 2018.

Xie et al., 2020
Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification.

Xu et al., 2019

Xu et al., 2022
Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization.
In CVPR, 2022.

Yang et al., 2021
In CVPR, 2021.

Yue et al., 2021
In CVPR, 2021.

Zakharov et al., 2019
In ICCV, 2019.

Zhang et al., 2021
In CVPR, 2021.

Zhou et al., 2019
Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks.
In CVPR, 2019.

Zou et al., 2018
In ECCV, 2018.

Zou et al., 2021
In ICCV, 2021.