基于关键点图的目标位姿估计学习框架

Shaobo Zhang, Wanqing Zhao*,Ziyu Guan,Xianlin Peng,Jinye Peng
西北大学
西安, 中国
{zhangshaobo@stumail., zhaowq@, ziyuguan@, pxl@, pjy@}nwu.edu.cn

摘要

目前许多 6D 姿态估计的方法利用了目标物体的 3D 模型来生成用于训练的融合图像，因为标签是容易获得的。然而，由于真实图像和融合图像之间的数据分布发生了域转移，仅用融合图像训练的网络不能在 6D 姿态估计的真实图片中捕捉到稳定的特征。我们解决这个问题是通过使网络对不同域不敏感，而不是通过使融合图像和真实图像更相似这一更困难的方法来解决。受域适应方法的启发，域自适应关键点检测网络 (DAKDN) 包含了用于最小化融合图像和真实图像之间的深度特征的域自适应层。这里的一个独特的挑战是缺乏真实图像的真标签值 (即关键点)。幸运的是，关键点之间的几何关系在真实和对称域之间是不变的。因此，我们提出利用关键点之间的域不变几何结构作为“桥”约束来优化 DAKDN，实现跨域 6D 位姿估计。具体来说，DAKDN 用了一个图卷积网络 (GCN) 块从合成图像中学习几何结构，并利用 GCN 块来指导对真实图像的训练。目标的 6D 位姿是基于预测的关键点的 PnP 算法来计算得到的。实验表明，我们的方法优于没有认为设置标签的最先进方法，并可与有人为设置标签的方法相媲美。

1. 引言

检测三维物体并估计它们的 6D 位姿是许多计算机视觉应用非常重要的任务，例如：增强现实，机器人技术，机器视觉。最近，深度学习方法 [13, 39, 4, 29, 34, 10, 26, 24, 25, 18, 3]已经在 RGB 图像的位姿估计中展现了不错的结果。然而，它们需要大量的人为标签，包括：2D 关键点，掩模，物体的 6D 位姿和其他标签。这是非常耗时的。

因此，有相关工作 [33, 12, 41, 36]尝试对三维模型进行渲染后的融合图像进行训练，产生一个具有位姿标签的大数据源。然而，这样做在三维模型和真实物体之间存在巨大差异 (例如：外表，光照条件)。融合图像和真实图像之间的差异称为域转变。仅用原始域数据 (融合图像) 对网络进行训练不能目标域数据 (真实图像) 的稳定特征，这降低了 6D 位姿估计的性能。为了提高真实图像的性能，AAE [33]和 DPOD [41]用了额外的改进包括在不同光照和背景下渲染图片来模拟真实的环境。不幸的是，数据增强不能重现真实世界的统计规律。Self6D [36]利用了真实图像上的预测掩模得到了作为约束条件的域独立性质，用该性质来改善基于物理渲染网络的估计位姿。尽管物理渲染能得到高质量的融合图像来模拟真实环境，掩模预测中仍然存在域间隔，这可能会限制位姿优化的性能。

为解决域转变问题，一些迁移学习方法 [35, 19, 17]在训练过程中加入未标记的目标数据。除了在原始域数据上训练模型，这些方法还通过直接优化自适应层产生的表示来最小化原始和目标域数据的特征分布的距离，以完成域转移。然而，由于 6D 位姿估计任务预测信息的高复杂度，直接使用未标记的目标数据进行 6D 位姿估计交叉域学习的性能还远

不能令人满意。

在本文中，我们旨在解决这个交叉域目标 6D 位姿估计的问题。受上述域转移工作的启发，我们也旨在学习域不变特征。不过，我们工作的创新点是在目标域上解决 6D 位姿估计任务缺少标签的问题。为此，我们提出了一个 6D 跨域姿态估计的关键点图驱动的学习框架，该框架结合了域转移和目标任务优化。具体来说，一个域自适应关键点检测网络 (DAKDN) 被用于在交叉域估计物体的 2D 关键点。物体的 6D 位姿可由预测的 2D 关键点通过 PnP 算法 [16]计算得到。为了使 DAKDN 能学到域不变的，合适的关键点检测的鲁棒特征，我们使用了融合图像和未标记的真实图像来训练，在网络中嵌入了自适应层以及基于最大均值偏差 (MMD) [35]的域对齐损失。为了提高对真实图像关键点检测的正确性，我们明确地将关键点之间的域不变结构从合成图像转移到真实图像，作为优化的“桥梁”约束跨域 6D 姿态估计的 DAKDN。关键点之间的几何结构是域不变结构，与真实图像和融合图像无关。因此，我们将几何关系表示为图，并训练一个图卷积网络 (GCN) [14]块来模拟合成图像中目标关键点的结构。然后将该结构转换到真实图像中作为约束，指导 DAKDN 正确检测真实图像中的关键点。通过对域不变性和结构预测的联合优化，域不变特性可以提高 GCN 结构的预测正确率，并且 GCN 可以引导网络在真实图像上提取合适的关键点特征进行姿态估计。实验表明，我们的方法优于没有认为设置标签的最先进方法，并可与有人为设置标签的方法相媲美。

2. 相关工作

近年来，6D 位姿估计采用了基于卷积神经网络的方法并展示出了不错的效果。这些方法用 CNN 来建立物体姿态与不同方位图像之间的对应关系。有的方法 [13, 39, 3, 4]直接预测物体位姿来建立联系。有的基于关键点的方法 [27, 25, 34, 10, 26, 31, 42]使用稀疏二维关键点作为姿态估计的中间表示来建立对应关系。一些基于密集的方法 [18, 24, 41]确定了密集的 2D-3D 对应，而不是稀疏的。虽然这些方法在准确性和运行时间上都有很好的效果，但它们需要大量的人工标记训练数据，如：2D 关键点，掩模，物体 6D 位姿以及其他标签，这通常是很耗时的。因此，有的方法 [33, 32, 12, 20, 41, 36]提出对物体 3D 模型渲染得到的融合图像进行训练，产生一个具有标签的大数据集。由于融合图像和真实图像之间的统计差异 (被称为域转移)。仅用原始域数据 (即融合图像) 训练模型通常会导致过拟合，导致识别目标域数据 (真实图像) 的性能降低。Self6D [36]利用融合图像和真实 RGB-D 图像作为训练数据。用融合图像训练网络得到 6D 位姿估计和掩模预测。RGB-D 图像具有域独立的性质，可作为自监督学习的约束来提高对真实图像 6D 位姿估计的性能。Self6D [36]高度依赖掩模预测来定位 RGB-D 图像中的物体，并且掩模预测还是对融合图像的训练。尽管基于物理的渲染能生成高质量的融合图像来模拟真实环境，但域间隔仍然存在且会限制掩模预测的性能。缩小融合图像和真实图像之间的域间隔对提高 6D 位姿估计的正确率非常重要。许多方法通过学习一种转变来解决这个问题，该转变是通过生成对抗网络 (GANs) [1, 15, 40]或特征映射 [28]对齐合成域和实域。例如，[15]使用基于解纠缠表示的跨周期一致性损失将图像嵌入到域不变内容空间和域特定属性空间。 [28]将基于颜色的姿态估计器的特征映射到基于深度的姿态估计器。Hinterstoisser [8]冻结了在真实图像大数据集上训练网络的第一层，并且提出了一个数据生成方法，使渲染的对象看起来真实。这些方法在不优化真实图像的姿态估计任务的情况下，将不同领域的数据分布从一个特定的度量对齐。

与这些方法不同，我们设计了一个关键点图驱动的目标位姿估计学习框架。该框架不仅对不同域图像上的关键点特征进行了比对，而且采用了域不变的关键点结构来优化真实图像中的关键点检测，提高了 6D 位姿估计的性能。

3. 方法

我们的目标是解决在交叉域 6D 位姿估计性能退化的问题。我们引入域自适应关键点检测网络

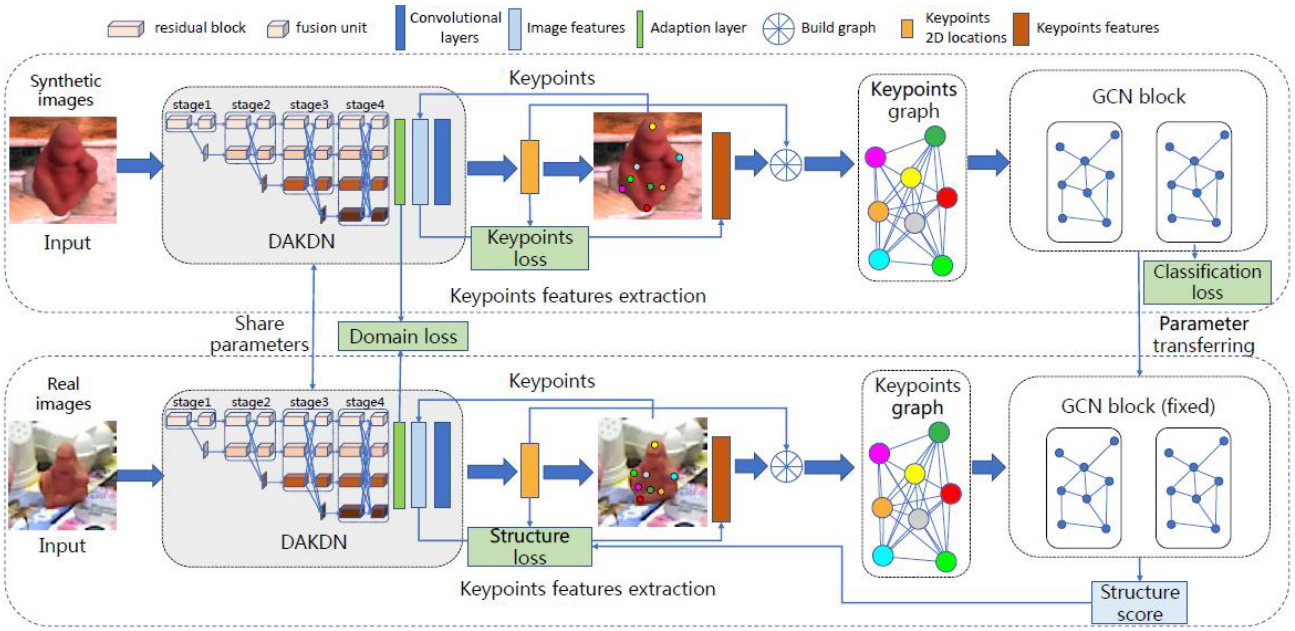

图 1. 关键点图驱动的学习框架的概览。标记的合成图像和未标记的真实图像用于训练域自适应关键点检测网络 (DAKDN) 和图卷积网络 (GCN) 块。我们对合成图像和真实图像使用不同的损失函数。对于合成图像，我们用关键点损失训练 DAKDN 来预测关键点，用分类损失训练 GCN 块来学习关键点结构。对于真实图像，我们使用由 GCN 预测结构分数计算得到的结构损失来训练 DAKDN。通过这种方式，可以利用关键点结构来优化真实图像的关键点检测。域对齐损失用来最小化合成图像和真实图像之间的域特征分布差异，使得 DAKDN 能学习域不变特征。

(DAKDN) 来预测目标关键点并通过 PnP 算法来计算 6D 姿态。因为物体的关键点有域不变结构，例如几何信息，所以我们将结构从原始域转变到目标域作为约束来在目标域上训练网络。具体来讲，在训练阶段，标记的合成图像和未标记的真实图像作为训练数据输入到 DAKDN。标记的合成图像通过关键点损失来训练 DAKDA 以用于关键点检测。我们也用所有图像通过域对齐损失来训练 DAKDN 以最小化域特征分布差异。为提高对真实图像关键点预测的正确率，我们使用一个 GCN 块来学习一个域不变关键点结构分类器。GCN 块能为在真实图像上检测到的关键点预测一个关键点结构分数来进一步优化 DAKDN。图 1 展示了我们方法的概览。

3.1. 域自适应关键点检测网络 (DAKDN)

图 1 展示了 DAKDN 的主要结构。DAKDN 将 256×256 RGB 图像作为输入，输出一系列热力图，一个关键点一张，热力图的亮度表示对应关键点被定位在该位置的置信度。我们用 HRNet[37]来提取关键点特征，这也能用其他常用的关键点检测网络来代替。HRnet 包含并行的高到低分辨率分支，通过重复特征融合生成可靠的高分辨率特征。网络主体由 4 个阶段组成，每阶段的分支随着阶数增加而增加。第 4 阶段包含 4 个分支，每个分支融合其他分支产生的多分辨率特征。我们使用输出的第 1 个分支来做后续的处理，它的分辨率在最后一阶段是最高的。在 HRNet 输出之后，我们增添 1 个 1×1 卷积层作为自适应层来学习一个表示，该表示可最小化合成图像和真实图像特征分布的距离。最后，两个连续的 1×1 卷积层用于产生热力图并通过 softmax 函数将热力图转换成概率分布图 $P(u, v)$，(u, v) 是热力图中的 2D 坐标。第 i 个关键点的坐标 x_i, y_i 可由如下公式计算：

$$x_i = \sum_{u,v} \lfloor u \cdot P_i(u, v) \rfloor, y_i = \sum_{u,v} \lfloor v \cdot P_i(u, v) \rfloor \quad (1)$$

其中，$P(u, v)$ 是关键点 i 的概率分布图，$\lfloor \cdot \rfloor$ 是地板运算符。

为了对 DAKDN 进行跨域训练，我们对

DAKDN 输入合成图像和真实图像并根据输入采用不同的损失函数。对于合成图像，因为它们被物体的关键点所标记，DAKDN 由监督方式进行训练。我们用如下定义的关键点损失函数 $L_{keypoints}$ 来比较预测关键点和真实关键点：

$$L_{keypoints} = \frac{1}{N}\sum_{i=1}^{N} Smooth_{L1}(x_i^p - x_i^{gt}) \quad (2)$$

其中 N 是关键点个数，x_i^p 是第 i 个预测点的坐标，x_i^{gt} 是第 i 个真实关键点的坐标，$Smooth_{L1}$ 定义如下：

$$Smooth_{L1} = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (3)$$

域对齐损失 L_{domain} 用来最小化合成图像和真实图像特征分布的距离。此外，对于真实图像，GCN 块用作输出关键点结构分数的分类器，并且我们使用关键点结构得分来计算结构损失 l 结构，以鼓励从真实图像中预测的关键点来满足某些目标关键点的结构。总损失定义如下：

$$L = L_{keypoints} + \mu L_{domain} + vL_{structure} \quad (4)$$

我们不仅想要最小化源域中的关键点损失，而且还想要域不变的特性。这样的特性将使我们能够学习一个可靠的关键点检测器，它可以轻松地检测跨域的关键点。结构损耗 $L_{structure}$ 可以使 DAKDN 减少预测关键点在真实图像上的几何误差。在训练阶段，只使用标注的合成图像计算 $L_{keypoints}$，只使用未标注的真实图像计算 $L_{structure}$，同时使用两个域的图像计算 L_{domain}. 在本节的其余部分中，我们将提供关于域对齐损失和结构损失的进一步详细信息。

3.2. 对齐域转变

在本节中，我们将详细描述域对齐损失的客观性。因为在跨域检测时，图像特征在合成特征和真实特征之间的域漂移会导致性能下降，学习领域不变特征可以提高目标领域的性能。为了学习合成图像和真实图像的域不变特征，我们在 DAKDN 中使用了自适应层和域对齐损失，以最小化合成图像特征和真实图像特征分布之间的距离。我们使用标准的分布距离度量，最大平均 (MMD)[35]. 这个距离是根据一个特定的表示来计算的,$\phi(\cdot)$。此时，我们定义 $\phi(\cdot)$ 为一个高斯核，它作用在合成图像和真实图像的特征上。因为高斯核函数可以将数据反映到无限维来测量分布距离。然后 MMD 可通过下式计算：

$$MMD(X_S, X_R) = \| \frac{1}{|X_S|}\sum_{x_S \in X_S} \phi(f(x_s)) - \frac{1}{|X_R|}\sum_{x_r \in X_R} \phi f(x_r) \| \quad (5)$$

其中，X_S, X_R 分别是合成图像和真实图像的块 (batch)。$f(x_s)$ 和 $f(x_r)$ 是自适应层的输出特征。最后，我们用 MMD 度量域对齐损失，即：$L_{domain} = MMD^2(X_S, X_R)$。域对齐损失减少了源域和目标域特征分布之间的差异，以对齐域偏移，并鼓励 DAKDN 学习域不变特征。

3.3. 转化结构

除了对齐域转变来减少差异，我们也希望学习到的真实图像特征适合于关键点检测任务。因此，我们进一步挖掘关键点之间的域不变结构，并以此作为约束来优化真实图像上的 DAKDN。采用 GCN 块来学习域不变关键点结构，即关键点之间的几何关系。首先，通过自适应层和域对齐损失对合成图像和真实图像之间的域移位进行对齐，但不能保证 DAKDN 算法适用于真实图像的关键点检测。其次，关键点之间的几何关系只与不同领域中具有相同结构的关键点位置有关。因此，几何关系是域不变的，可以利用几何关系作为监督信息来约束跨域的关键点预测。最后，域不变关键点结构可以用图来表示，GCNs 可以通过图自然地对关键点之间的骨架约束进行建模。该结构可以从合成图像中学习，并通过 GCN 块作为约束转移到真实图像中，进一步优化网络，以跨域检测关键点。基于这些，我们定义一个邻接矩阵 $A \in \mathbb{R}^{N \times N}$ 来表示几何关系。N 是关键点个数，A_{ij} 是关键点 i 和关键点 j 之间的几何信息。同时定义 $X \in \mathbb{R}^{N \times M}$ 表示关键点特征。M 是关键点特征的维度。该方法的一个关键是构造合适的 X 和 A，使在合成图像上学习的关键点结构可以优化 DAKDN 表示真实图像。我们将每个图像中的关键点看作一个图，其中每个节点代表一个关键点，边是关键点之间的几何关系。

在本例中，我们定义了如图 2 所示的几何关系。节点 1 和节点 2 之间的关系是夹角 α 除以中心点 (黑点) 到直线 (绿色线) 的距离。角度除以距离可以看作是对物体的缩放、旋转和材质不变的几何表示。在我们的实验中，我们简单地使用物体包围框的中心作为物体的中心点。做如下定义：A_{ij} 由关键点 x_i 和 x_j 的坐标计算得到。节点的特征是自适应层最终输出的特征图上对应节点位置的值。

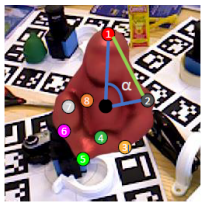

图 2. 节点 1 和节点 2 之间的角度和距离的演示

我们使用两层的 GCN 用于关键点结构分类。首先计算 $\widehat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$，其中 $\widehat{A} = A + I_N$，$\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$。$I_N$ 是单位矩阵。这是为了标准化 A，以避免数值不稳定性和爆炸/消失梯度。我们的正向模型采用了简单的形式:

$$Z = f(X, A) = \widehat{A}ReLU(\widehat{A}XW^{(0)})W^{(1)} \quad (6)$$

其中 $W^{(0)} \in \mathbb{R}^{M\times H}$ 是由 H 个单元的输入到隐藏层的隐藏层权重矩阵，$W^{(1)} \in \mathbb{R}^{H\times F}$ 是隐层到输出层的权重矩阵。F 是节点特征的输出维数。在本例中，设置 $H = 128 \quad F = 64$。$Z \in \mathbb{R}^{N\times F}$ 是 GCN 提取的图的高阶特征。GCN 网络之后，有一个全连接层和一个 softmax 函数用于输出 $\mathrm{p} \in \mathbb{R}^{1\times C}$，其中 C 是目标类别数。p^c 表示输入图是 c 的概率。通过分类损失训练 GCN 块，利用标记的合成图像学习关键点的结构。分类损失是定义如下的交叉熵损失：

$$L_{class} = -\frac{1}{[X_S]}\sum_{x_s\in X_S}\sum_{c=1}^{C} y_{x_s}^c log(p_{x_s}^c) \quad (7)$$

$y_{x_s}^c$ 是指示变量 (0 或 1)。$y_{x_s}^c = 1$ 表示 x_s 属于 c 类。$p_{x_s}^c$ 是 x_s 属于 c 类的预测概率。然后利用在合成图像上学习到的关键点域不变结构作为监督信息，对真实图像上的关键点预测进行约束。具体来说，对于真实图像，GCN 块用作分类器输出关键点结构得分，该得分衡量预测的关键点是否满足输入目标类的结构。基于此，我们设计了如下结构损耗:

$$L_{structure} = -\frac{1}{|X_R|}\sum_{x_r\in X_R}\sum_{c=1}^{C} y_{x_r}^c log(p_{x_r}^c) \quad (8)$$

$y_{x_r}^c$ 是指示变量 (0 或 1)。$y_{x_r}^c = 1$ 表示 x_r 属于 c 类。$p_{x_s}^c$ 是 x_r 属于 c 类的预测概率。L_{class} 和 $L_{structure}$ 的形式是一样的。但是与 L_{class} 不同，$L_{structure}$ 是用于合成图像训练 DAKDN 来预测满足特定对象类结构的关键点。对于真实图像，GCN 块的参数是固定的，$L_{structure}$ 的梯度通过 A_{ij} 向后传播，只更新 DAKDN。这种损失将结构作为监督信息来细化真实图像上的 DAKDN。

3.4. 位姿估计

本节描述使用 DAKDN 的输出计算姿势的过程。DAKDN 预测的 2D 关键点是预先定义的 3D 关键点的投影。我们利用 PnP 方法从二维和三维点之间的对应关系估计 6D 位姿。在我们的案例，PnP 仅使用 8 个关键点对应就能估计出物体在相机坐标系中的 3D 旋转矩阵 R 和 3D 平移矩阵 t 。

4. 实验

在本节中，我们首先介绍实现细节和数据准备。然后分析了不同消融方式下的区域移位对齐和 GCN 块的有效性。最后，在 LINEMOD[7]、OCCLUSION[2]、HomebrewedDB[11]和剪裁过的 LINEMOD[38]数据集上对算法进行了评估，并将结果与目前最先进的 6D 姿态估计方法进行了比较。在姿态估计之前，每个边界框的中心、宽度和高度由现成的目标检测网络进行预测，即 Faster R-CNN[30]。然后物体的区域被裁剪并调整为输入的大小：256 × 256 px。在补充材料中给出了目标检测网络的细节和评价。

4.1. 实施细节

我们使用 Pytorch 深度学习框架来实现我们的方法。训练和评估使用 8 个 Nvidia GTX 2080Ti gpu 和 i7-6700K CPU 执行。为训练网络，对于 DAKDN，我们使用学习率为 2.5×10^{-4} 的 ADAM，权重衰减为 5×10^{-4}batch 大小为 32，进行了 250k 次迭代。对于 GCN 块，学习率为 0.001，权重衰减为 5×10^{-4}。对于多任务损失函数权重，我们经验性的设置 μ 为 0.5，ν 为 0.3。

人工数据合成给定物体的三维模型，首先按照PVnet][26]中的方法定义物体表面的关键点，使用最远点采样 (FPS) 算法选取 K 个关键点。我们使用搅拌机 [26]从不同的摄像机视点渲染这些 3D 模型，以充分覆盖对象，并将关键点投影到视点下的图像。我们等距离的在物体上方不同距离的半球体周围采样 40000 个相机视点。物体在随机选择的背景图像中的随机位置被渲染。为了使关键点检测网络对不同背景进行泛化，防止训练过程中对背景过拟合，我们选择来自 PCASOL VOC 数据集 [5]的图像作为背景。为了进一步增强合成图像，我们使用随机扰动的光色，并在渲染过程中添加图像噪声。此外，我们使用高斯滤波器模糊对象，以更好地集成渲染与背景。我们还使用对象的 CAD 模型和相应的姿态计算一个紧密拟合的边界框。除了基于混合器的合成图像外，我们还采用了 [9]中提出的逼真和物理可靠的渲染过程来渲染图像进行训练。实验中的训练数据是两种合成图像的混合。

图 3.

Table 1. Ablation results on LINEMOD dataset.

Training data		Backbone		Modules			Accuracy
Syn	Real	HRNet [37]	SH [22]	Adaption layer	w/ G-info	w/F-info	Mean
✓		✓					46.5
✓	✓	✓		✓			51.8
✓	✓	✓			✓	✓	50.4
✓	✓	✓		✓		✓	53.4
✓	✓	✓		✓	✓		55.3
✓	✓	✓		✓	✓	✓	68.2
✓			✓				31.3
✓	✓		✓	✓	✓	✓	46.1

4.2. 数据集和评价度量

我们在 LINEMOD[7]、OCCLUSION[2]、HomebrewedDB[11]和剪裁过的 LINEMOD[1]四个公共基准数据集上进行实验。它们被广泛应用于评估 6D 位姿估计方法。LINEMOD 由 15 个颜色、形状和大小有区别的无纹理家用物品组成。每个对象都与一个测试图像集相关联，该测试图像集显示一个带有明显杂乱的注释对象实例。其中只有 13 个对象具有完整的 CAD 模型，因此我们选择了相应的图像序列。OCCLUSION 起源于 LINEMOD，其中在每个测试图像中使用不同级别的遮挡标注多个对象实例。如 self-6D[37]一样。我们使用最近的 HomebrewedDB[11]数据集，并使用 LINEMOD 中覆盖三个对象的序列，在未见过的环境中评估我们的方法。该方法的训练数据包括合成图像和真实图像。在 LINEMOD 和 OCCLUSION 数据集中，我们使用与 YOLO 6D[35]相同的训练/测试分割并为每个物体渲染 40000 张图像作为训练数据。郑重声明，用于训练的真实图像没有标记。为与其他用于 6D 位姿估计的域自适应方法作比较，我们引用裁剪的 LINMODE 数据集 [1]，它由从 LINEMOD 剪裁过的真实和合成图像组成。我们在 [1]中使用相同的训练/测试图像，其中训练图像包含有标记的合成图像和未标记的真实图像，而测试图像只包含真实图像。

为了评估估计姿态的准确性，我们使用了其他相关论文 [36, 41, 26]中使用的 LINEMOD 的两个标准度量，它们是 ADD 和 ADD-S(对合成目标)。当 $ADD(-S)$ 小于物体直径的 10% 时，估计的姿态被认为是正确的。

4.3. 消融学习

为分析自适应层和 GCN 块的有效性，表 1 中，指出了我们的方法在 LINEMOD 数据集上使用的 ADD 度量时不同训练数据和模块的平均精度。如表 1 所示，合成图像用 Syn 表示，真实图像用 Real 表示。由于关键点检测网络是可选的，我们使用了两个较广泛的网络，即 HRNet[37]和 Stacked Hourglass(SH)[22]。我们还使用不同的方法来说明 GCN 块的作业，并将它们与其他替代方法进行比较。(a) 我们只用 GCN 来学习关键点之间的几何信息，即用一个单位矩阵代替特征矩阵，用 w/G-info 表示。(b) 我们只使用 GCN 来学习关键点的深度特征信息，将 GCN 的邻接矩阵替换为充满 1 的矩阵，用 w/F-info 表示。(c) 我们使用 GCN 来学习关键点的几何信息和深度特征信息。

首先，我们使用 HRNet 作为 DAKDN 的主干，只在合成图像上训练 DAKDN，不使用自适应层和 L_{doamin}。结果表明，姿态估计的精度为 46.5%。其次，我们用两种类型的数据训练整个网络，准确率

图 4. 在 ADD(-S) 度量方面，我们的方法和最先进的方法在 LINEMOD 数据集上的准确性

labels	w/o manual pose labels					w/ manual pose labels			
Training data	Syn			Syn+Real		Real			
Method	AAE [33] †	MHP [20] †	DPOD [41]	Self6D [36] †	Ours	YOLO6D [34]	DPOD	PVNet [26]	CDPN [18]
Ape	4.2	11.9	55.2	38.9	**78.4**	21.6	53.3	43.6	64.4
Bvise	20.9	66.2	72.3	75.2	**79.7**	81.8	95.3	99.9	97.8
Cam	32.9	22.4	34.8	36.9	**48.3**	36.6	90.0	86.9	91.7
Can	37.0	59.8	**83.6**	65.6	71.1	68.8	94.1	95.5	95.9
Cat	18.7	26.9	**65.1**	57.9	58.4	41.8	60.4	79.3	83.8
Drill	24.8	44.6	73.3	67.0	**75.5**	63.5	97.7	96.4	96.2
Duck	5.9	8.3	**50.0**	19.6	35.6	27.2	66.0	52.6	66.8
Eggbox	81.0	55.7	89.1	**99.0**	97.2	69.6	99.7	99.2	99.7
Glue	46.2	54.6	84.4	94.1	**96.8**	80.0	93.8	95.7	99.6
Holep	18.2	15.5	**35.4**	16.2	28.5	42.6	65.9	81.9	85.8
Iron	35.1	60.8	**98.8**	77.9	83.1	75.0	99.8	98.9	97.9
Lamp	64.2	-	74.3	68.2	**76.8**	71.1	88.1	92.4	97.9
Phone	36.3	34.4	46.9	50.1	**57.5**	47.7	71.2	86.3	90.8
Mean	32.6	38.8	66.4	58.9	**68.2**	56.0	83.0	86.3	89.9

由 46.5% 提高到 51.8%。当我们移除自适应层并添加整个 GCN 块时，准确率由 46.5% 提高到 50.4%。然后，我们评估了 GCN 块对 DAKDN 的作用，结果表明，G-info 和 F-into 使 DAKDN 的精度分别提高了 3.5% 和 1.6%。当采用我们的训练框架时，可以从 46.5% 到 68.2% 提高 21.7% 的精度。结果表明，自适应层和 GCN 块都能提高精度。与单独使用这两个模块相比，将它们结合使用的改进是显著的。这是因为结合两个模块具有互补的优势。域自适应层减小了两个域关键点深度特征的差异，提高了 GCN 对真实图像结构预测的准确性。GCN 块可以进一步微调真实图像上关键点的深度特征，提高姿态估计的性能。我们还用 Stacked Hourglass(SH)[22]代替 HRNet[37]并在 LINEMOD 上进行测试。与 Stacked Hourglass 在合成图像训练中相比，我们的框架将正确率从 31.3% 提高到 46.1%。结果表明，该框架可用于其他关键点检测网络，提高跨域估计的准确性。

图 5. 裁剪后的 LineMOD 数据集上的平均角度误差

Method	PixelDA [1]	Self6D [36]	Ours
Mean Angle Error(°)	23.5	19.8	**17.6**

Table 4. The Average Recall(%) on the HomebrewedDB dataset.

Method	SSD6D +Ref [12]	DPOD [41]	Self6D [36]	Ours
Training data	Syn	Syn	Syn+Real	Syn+Real
Mean	32.7	43.37	59.7	**63.8**

4.4. 目标检测与关键点图分类的分析

我们使用 Faster-RCNN[30]和 Resnet-101[6]骨干对图像中的目标进行裁剪，进行 6D 姿态估计。Faster-RCNN 的正确 2D 边界框平均百分比 (IoU>0.5) 在 LINEMOD 上达到了 84.3%。我们还独立评估了 GCN 块对 LINEMOD 的影响。我们将预测的结构分数绘制在不同关键点误差的关键点上，分析 GCN 的影响。关键点误差定义为实际测试图像上预测关键点与真实关键点之间的平均欧氏距离。根据预测的关键点误差对预测的关键点进行排序，并将预测的关键点平均分为 12 个部分。然后计算各部分关键点的结构得分均值和关键点误差。结果表明，随着关键点误差的增加，GCN 的预测结构评分会下降。这确保结构得分可作为优化关键点检测网络的约束条件，即关键点误差越大，训练 DAKDN 的损失就越大。对比使用 GCN 块对真实图像 DAKDN 进行优化前后的结果，从图6可以看出，优化后预测的关键点关键点误差减小。

4.5. 单个物体位姿估计

图 4 指出了在 LineMOD 数据集上用 ADD 正确估计的位姿的百分比。我们将我们的方法与使用由 3D CAD 模型生成的合成图像 (AAE[33]，MHP[20]，DPOD[41]和 Self6D[36]) 和人为注释的真实图像 (YOLD6D[34]，DPOD[41],PVNet[26]，CDPN[18]) 的最先进 6D 姿态估计方法进行了比较。表 2 左

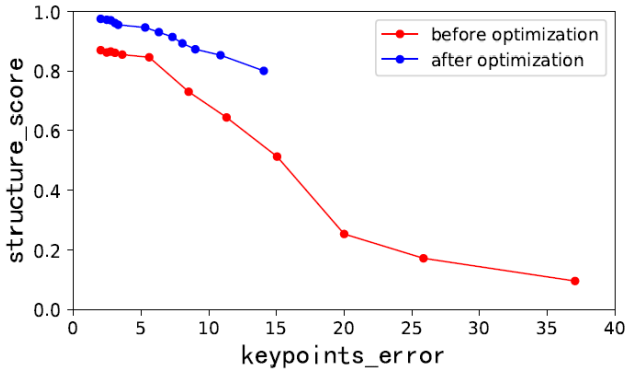

图 6. 在 LINEMOD 上用不同关键点误差预测的结构得分

侧报告了未经人工 3D 标注的数据训练方法的准确性。在大多数物体上，我们的方法优于所有其他方法。我们的关键点图驱动学习框架确保了我们的方法比 DPOD 的性能高出 11.3%，尽管 DPOD 可用一个后改善的网络提高它的性能。AAE[33]，MHP[20]和 Self6D[36]使用平均召回率 (%)。为了进行公平的比较，我们计算了用 Faster-RCNN[30]检测图像中物体的平均召回率，平均召回率为 60.4%。我们的平均召回率是 AAE[33]的近 2 倍，优于 MHP[20]21.6%。与 Self6D[36]相比，我们的方法略优 1.5%。值得注意的是，Self-6D 需要使用真实的未注释图像进行深度测量，而我们的方法可以用 RGB 图像。这些结果表明了该方法的优越性。这主要是为了调整合成图像和真实图像之间的域位移，并学习域不变结构作为约束来优化真实图像中的网络。表 2 的右侧是人工姿态标注在真实图像上训练的方法的准确率。我们的方法可以获得比 YOLO6D[34]更好的结果，并且在一些目标序列中仍然可以获得接近甚至更好的结果。

由于我们的方法旨在弥合合成数据和真实数据之间的域转移，我们将我们的方法与其他使用了剪裁 LineMOD[1]的域适应技术进行了比较。我们在测试集上报告了平均角度误差。如表 3 所示，我们的方法在真实图像上成功地优于其他方法，将平均角度误差降低到 17.6。。

我们还使用 HomebrewedDB 数据集来评估我们的方法在未见过的环境中，并与 Self6D，DPOD 和使用 [21]改善的 SSD6D[12]。表 4 显示，我们的方法显著优于 SSD6D 和 DPOD(31.1% 和 21.5%)，略优于 Self6D(4.1%)。图 4 提供了真实姿态与预测姿态的视觉对比。除此之外，我们在补充材料中使用不同的度量和 T-LESS 数据集展示了更多的实验结果。

4.6. 多目标位姿估计

本方法在检测目标增多和遮挡下性能评估在 OCCLUSION 数据集上进行。我们使用在合成图像上训练的模型在遮挡数据集上进行测试，并将我们的方法与不需要人为姿态标签的三种方法 (DPOD[41],CDPN[18],Self6D[36]) 和使用人为姿态标签的三种方法 (YOLO6D[29],HMap[23],PVNet[26]) 进行比较。我们的方法优于 DPOD27.4%，CDPN12.9%，Slef6D1.6%。与使用人为姿态标签的方法相比，我们方法的平均精密度为 38.7%，其性能显著优于 YOLO6D(32%)，优于 HMap(8.3%)，可与 PVnet 媲美。当我们忽略 GCN 块时，平均召回率下降到 12.5%。这表明我们的方法可以处理遮挡，因为目标的结构包含关键点之间的几何关系，并由 GCN 块学习。当物体的部分被遮挡时，其他关键点仍然可以根据学习到的关系来预测未被遮挡部分上的关键点。

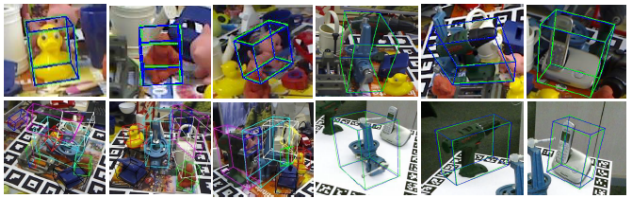

图 7. 结果：在 LineMOD, OCCLUSION 和 HomebrewedDB 上用所提出的方法预测的姿态。绿色边界框代表真实姿态，蓝色边界框代表预测姿态。

5. 总结

在本文中，我们提出了一个关键点图驱动的学习框架，用于跨领域的目标姿态估计。我们设计了 DAKDN 来预测物体上的关键点，并使用 PnP 算法计算 6D 位姿。为了使 DAKDN 对域移位有鲁棒性，我们使用 GCN 块从合成图像中学习域不变关键点结构，并将其转换到真实图像中。实验结果表

明，该方法优于目前最先进的没有人为姿态标签的方法，可与需要真实人为姿态标签的方法相媲美。

致谢. 这项研究得到了中国国家自然基金会的支持。

参考文献

[1] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. 1066, 1070, 1072

[2] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In ECCV. Springer, September 2014. 1069, 1070

[3] C. Capellen, M. Schwarz, and S. Behnke. Convposecnn: Dense convolutional 6d object pose estimation. CoRR, abs/1912.07333, 2019. 1065, 1066

[4] T. Do, M. Cai, T. Pham, and I. D. Reid. Deep-6dpose: Recovering 6d object pose from a single RGB image. CoRR, abs/1802.10367, 2018. 1065, 1066

[5] M. Everingham, S. Eslami, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111:98–136, 2014. 1070

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. 1071

[7] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, editors, Computer Vision – ACCV 2012, pages 548–562, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 1069, 1070

[8] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige. On pre-trained image features and synthetic images for deep learning. CoRR, abs/1710.10710, 2017. 1066

[9] T. Hodan, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. N. Sinha, and B. Guenter. Photorealistic image synthesis for object instance detection. CoRR, abs/1902.03334, 2019. 1070

[10] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann. Segmentation-driven 6d object pose estimation. CoRR, abs/1812.02541, 2018. 1065, 1066

[11] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic. Homebreweddb: RGB-D dataset for 6d pose estimation of 3d objects. CoRR, abs/1904.03167, 2019. 1069, 1070

[12] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. CoRR, abs/1711.10006, 2017. 1065, 1066, 1072

[13] A. Kendall, M. Grimes, and R. Cipolla. Convolutional networks for real-time 6-dof camera relocalization. CoRR, abs/1505.07427, 2015. 1065, 1066

[14] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. CoRR, abs/1609.02907, 2016. 1066

[15] H. Lee, H. Tseng, Q. Mao, J. Huang, Y. Lu, M. Singh, and M. Yang. DRIT++: diverse image-to-image translation via disentangled representations. CoRR, abs/1905.01270, 2019. 1066

[16] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. Int. J. Comput. Vision, 81(2):155–166, Feb. 2009. 1066

[17] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu. Adaptive batch normalization for practical domain adaptation. Pattern Recognition, 80:109–117, 2018. 1065

[18] Z. Li, G. Wang, and X. Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 7677–7686, 2019. 1065, 1066, 1071, 1072

[19] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In F. Bach and D. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 97–105, Lille, France, 07–09 Jul 2015. PMLR. 1065

[20] F. Manhardt, D. M. Arroyo, C. Rupprecht, B. Busam, N. Navab, and F. Tombari. Explaining the ambigu-

ity of object detection and 6d pose from visual data. CoRR, abs/1812.00287, 2018. 1066, 1071, 1072

[21] F. Manhardt, W. Kehl, N. Navab, and F. Tombari. Deep model-based 6d pose refinement in RGB. CoRR, abs/1810.03065, 2018. 1072

[22] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. CoRR, abs/1603.06937, 2016. 1070, 1071

[23] M. Oberweger, M. Rad, and V. Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. CoRR, abs/1804.03959, 2018. 1072

[24] K. Park, T. Patten, and M. Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. CoRR, abs/1908.07433, 2019. 1065, 1066

[25] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. CoRR, abs/1703.04670, 2017. 1065, 1066

[26] S. Peng, Y. Liu, Q. Huang, H. Bao, and X. Zhou. Pvnet: Pixel-wise voting network for 6dof pose estimation. CoRR, abs/1812.11788, 2018. 1065, 1066, 1070, 1071, 1072

[27] M. Rad and V. Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. CoRR, abs/1703.10896, 2017. 1066

[28] M. Rad, M. Oberweger, and V. Lepetit. Domain transfer for 3d pose estimation from color images without manual annotations. CoRR, abs/1810.03707, 2018. 1066

[29] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. CoRR, abs/1612.08242, 2016. 1065, 1072

[30] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, abs/1506.01497, 2015. 1069, 1071, 1072

[31] C. Song, J. Song, and Q. Huang. Hybridpose: 6d object pose estimation under hybrid representations. CoRR, abs/2001.01869, 2020. 1066

[32] M. Sundermeyer, M. Durner, E. Y. Puang, Z. Marton, and R. Triebel. Multi-path learning for object pose estimation across domains. CoRR, abs/1908.00151, 2019. 1066

[33] M. Sundermeyer, Z. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from RGB images. CoRR, abs/1902.01275, 2019. 1065, 1066, 1071, 1072

[34] B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. CoRR, abs/1711.08848, 2017. 1065, 1066, 1071, 1072

[35] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. CoRR, abs/1412.3474, 2014. 1065, 1066, 1068, 1070

[36] G. Wang, F. Manhardt, J. Shao, X. Ji, N. Navab, and F. Tombari. Self6d: Self-supervised monocular 6d object pose estimation. CoRR, abs/2004.06468, 2020. 1065, 1066, 1070, 1071, 1072

[37] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. CoRR, abs/1908.07919, 2019. 1067, 1070, 1071

[38] P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. CoRR, abs/1502.05908, 2015. 1069

[39] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. CoRR, abs/1711.00199, 2017. 1065, 1066

[40] S. Zakharov, W. Kehl, and S. Ilic. Deceptionnet: Network-driven domain randomization. CoRR, abs/1904.02750, 2019. 1066

[41] S. Zakharov, I. Shugurov, and S. Ilic. DPOD: dense 6d pose object detector in RGB images. CoRR, abs/1902.11020, 2019. 1065, 1066, 1070, 1071, 1072

[42] Z. Zhao, G. Peng, H. Wang, H. Fang, C. Li, and C. Lu. Estimating 6d pose from localizing designated surface keypoints. CoRR, abs/1812.01387, 2018. 1066