IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 January 2023, accepted 6 February 2023, date of publication 8 February 2023, date of current version 14 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3243551

RESEARCH ARTICLE

CVML-Pose: Convolutional VAE Based Multi-Level Network for Object 3D Pose Estimation

JIANYU ZHAO, (Graduate Student Member, IEEE), EDWARD SANDERSON, AND BOGDAN J. MATUSZEWSKI, (Member, IEEE)

Computer Vision and Machine Learning (CVML) Group, University of Central Lancashire, PR1 2HE Preston, U.K.

Corresponding author: Jianyu Zhao (jzhao12@uclan.ac.uk)

The work of Bogdan J. Matuszewski was supported in part by the Engineering and Physical Sciences Research Council under Grant EP/K019368/1.

ABSTRACT Most vision-based 3D pose estimation approaches typically rely on knowledge of object's 3D model, depth measurements, and often require time-consuming iterative refinement to improve accuracy. However, these can be seen as limiting factors for broader real-life applications. The main motivation for this paper is to address these limitations. To solve this, a novel **C**onvolutional **V**ariational Auto-Encoder based **M**ulti-**L**evel Network for object 3D pose estimation (CVML-Pose) method is proposed. Unlike most other methods, the proposed CVML-Pose implicitly learns an object's 3D pose from only RGB images encoded in its latent space without knowing the object's 3D model, depth information, or performing a post-refinement. CVML-Pose consists of two main modules: (i) CVML-AE representing convolutional variational autoencoder, whose role is to extract features from RGB images, (ii) Multi-Layer Perceptron and K-Nearest Neighbor regressors mapping the latent variables to object 3D pose including, respectively, rotation and translation. The proposed CVML-Pose has been evaluated on the LineMod and LineMod-Occlusion benchmark datasets. It has been shown to outperform other methods based on latent representations and achieves comparable results to the state-of-the-art, but without use of a 3D model or depth measurements. Utilizing the t-Distributed Stochastic Neighbor Embedding algorithm, the CVML-Pose latent space is shown to successfully represent objects' category and topology. This opens up a prospect of integrated estimation of pose and other attributes (possibly also including surface finish or shape variations), which, with real-time processing due to the absence of iterative refinement, can facilitate various robotic applications. Code available: https://github.com/JZhao12/CVML-Pose.

INDEX TERMS 3D pose estimation, deep learning, variational autoencoder, synthetic data.

I. INTRODUCTION

The scene understanding and automatic manipulation of various objects present in that scene are of fundamental importance to robotics and automation in general. For example, an accurate and fast estimation of an object's location and orientation with six degrees of freedom is one of the cornerstones for many modern autonomous system applications, including explorative navigation, augmented reality, and robotic manipulation.

Most state-of-the-art methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] for 3D pose estimation typically require object's 3D model or 2D-3D correspondence information. Approaches that rely on 2D-3D correspondence estimate pixel-wise dense correspondence [4], [5], [9], or matching between a number of sparse keypoints [6], [10]. Subsequently, these approaches estimate pose through various Perspective-n-Point (PnP) algorithms. Other approaches construct a loss function utilizing 3D model keypoints [3], [7], [8] or iteratively match the image rendered from a 3D model at its estimated pose with the observed input image [1], [2]. Most object pose estimation networks only focus on regressing object 3D pose, in which case they are not able to generalize

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

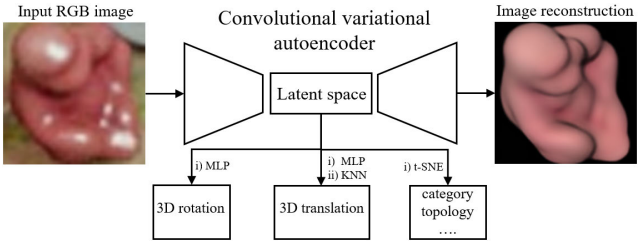

FIGURE 1. CVML-Pose pipeline. During training, the convolutional variational autoencoder captures object's latent representation in its latent space that is further interpreted to object 3D pose, category, and topology using MLPs, KNN, and t-SNE.

to other attributes of the object. Besides, it is impractical to construct a 3D model for every object of interest, or compute the prior 2D-3D correspondence in real-time.

There are a series of public challenges called "Benchmark for 6D[1] Object Pose Estimation (BOP)" [13], [14] that aim to continuously report the state-of-the-art in object 3D[2] pose estimation. The goal of the BOP challenge is to estimate 3D pose including 3D rotation and 3D translation of rigid objects from RGB/RGB-D images. There are a total of 12 publicly available datasets [15] adopted for evaluation of the challenge. Each dataset is provided in the BOP format and includes the 3D object's model with training and test images annotated with ground truth 3D poses. The training images are either captured by an RGB-D/Gray-D sensor or obtained by rendering the object model. The test images are captured in scenes with graded complexity, often with clutter and occlusion. In this paper, the LineMod [16] and the LineMod-Occlusion [17], [18] data are selected for evaluation, since LineMod is a widely-used pose estimation benchmark dataset for many state-of-the-art approaches [3], [6], [10], [19], [20], and LineMod-Occlusion introduces strong occlusions that increases the difficulty for pose estimation. Since the objects in LineMod-Occlusion are a subset of the objects in LineMod, the same network (weights) trained for each LineMod object can also be used for the corresponding object in LineMod-Occlusion. The LineMod and LineMod-Occlusion data are considered to be heterogeneous, as the objects are different in shape and appearance. Additionally, images of the object, both with and without occlusion and clutter, exhibit very different characteristics, varied illumination and shadow patterns.

The originality and novelty of this work is the proposal of a fast and accurate pose estimation algorithm which does not require an object's 3D model, i.e. a model-free object 3D pose estimation approach. To achieve this, we propose a novel **C**onvolutional **V**ariational Auto-Encoder based **M**ulti-**L**evel Network for the estimation of object 3D pose (CVML-Pose) from a single RGB image without depth information or any post-refinement. Unlike many state-of-the-art methods that rely on an object's 3D model or 2D-3D correspondence, the CVML-Pose pipeline (Fig. 1) firstly learns a latent representation directly from 2D images, then multiple supervised learning methods, such as Multi-Layer Perceptron (MLP) and K-Nearest Neighbors (KNN), are adopted to predict object 3D pose from the frozen latent space. Using the t-SNE algorithm [21], we show that the latent space captures not only 3D pose but also other attributes of the object e.g. class and topology. Our method has been evaluated on the LineMod and the LineMod-Occlusion data, and the experimental results demonstrate that it learns to represent the complete 3D pose of objects in its latent space. Compared to the state-of-the-art [2], [4], [5], [6], [9], [19], [20], [22], [23], the proposed CVML-Pose shows comparable results in the 3D pose estimation task, and the learnt representation provides a potential for a more complete characterization of objects.

There are two main motivations of this work: industrial applications and technical innovations. For applications, the estimation of object 3D pose is of great importance for augmented reality and robot manipulation. In particular, with the development of e-commerce, a large number of parcel packages have to be processed. In recent years, e-commerce companies have invested a large amount of money on packaging automation. As an example, Amazon Picking Challenge (APC) [24], [25] allows individuals to create their own small robot systems to pick items from the warehouse shelf, to overcome the problems of robotic automation and management. In industrial automation, estimating the 3D rotation and translation of target objects can enable industrial robots to reliably grab and manipulate these objects. From the perspective of technical innovation, the ability to perceive and recognize objects seems inherent for humans. Children learn about the world by observing, touching and grabbing the surrounding objects, even if they don't know what they are. When they grow up, they tend to rely on knowledge after they have accumulated a certain amount of experience. For example, a person who has never seen a specific design of a chair will still be able to recognize its function. Similarly, the proposed CVML-Pose is capable of estimating object's 3D pose from 2D images without knowing the object's 3D model, depth measurement or performing a post-refinement. The latent space also shows potential for the CVML-Pose to extend its functionality to include detection and estimation of other object's attributes.

II. LITERATURE REVIEW

There are a number of non deep learning-based techniques for object 3D pose estimation. Traditionally, such algorithms were based on extracting 2D local feature descriptors such as SIFT [26] and SURF [27] on textured objects. The 2D feature points are extracted and matched to the 3D points of object model, and then the 3D pose can be recovered by using PnP algorithms [28]. As an example, Lourakis and Zabulis [29] proposed a feature-based pose estimation method where the SIFT descriptors are detected in the observed image and then matched against those contained in the 3D model, then object 3D pose is estimated by using a RANSAC-based PnP

[1]6D refers to 6 degrees of freedom.
[2]3D refers to three-dimensional.

algorithm [30]. Another method based on feature descriptors is proposed in [31], which builds a global model description using the point pair feature. The final object pose can be estimated with an efficient voting loop if the reference point lies on the surface of the object. Approaches [29], [32], [33] that use explicit feature descriptors are capable of pose estimation with high precision. However, the availability of an accurate textured 3D model is essential. Furthermore, they are sensitive to objects' occlusions resulting in the reduction of 2D feature points and therefore compromising reliable pose estimation. Also, since feature-based methods assume rich textures, they cannot deal with texture-less objects where there are no feature points to be extracted and matched.

To work with texture-less objects, template-based methods [16], [34], [35] utilize color gradient information on the silhouette and interior surface normal features. For instance, Hinterstoisser et al. [16] first generate template images with ground truth object poses, then the most similar template can be retrieved from the silhouette. The final pose is further improved using the 3D surface normal orientations acquired from a depth sensor. Li et al. [36] proposed an efficient method capable of handling texture-less objects based on stably observed point pair features. The method requires both depth measurement and knowledge of the object's 3D model. Methods based on both texture and shape information were also proposed in [37]. Although template-based methods are useful for handling texture-less objects, they heavily rely on the knowledge of object's 3D model and/or depth information that are not always easily available in real-life applications.

With the introduction of deep convolutional neural networks (CNNs), many deep learning-based approaches [1], [2], [3], [4], [5], [6], [9] have achieved impressive results on 3D pose estimation. The rest of the literature review section describes some of the deep learning-based methods, including the indirect methods, the direct methods, and the latent representation methods. The BOP challenges and variational autoencoder are also discussed in this section. More comprehensive reviews on 3D pose estimation can be found in [11] and [12].

A. DEEP LEARNING-BASED INDIRECT METHODS

The deep learning-based indirect methods (indirect methods in short) focus on learning 2D-3D correspondence with different versions of PnP algorithms [4], [5], [6], [9], [10], [20], [38].

For example, Tekin et al. [10] utilize the YOLOv2 detector and then calculate object 3D pose with 2D-3D correspondence followed by a PnP algorithm. BB8 [38] first performs a two-level coarse-to-fine object segmentation based on the VGG network to localize objects in 2D images and then predict 3D pose in the form of 2D projections of the corners of the 3D bounding boxes based on the PnP. PVNet [6] establishes a voting-based 2D-3D keypoints detection using a modified ResNet-18 [39] followed by a RANSAC-based PnP algorithm which prunes the matched pairs and estimates the correct pose. EPOS [4] predicts correspondence between densely sampled pixels and the object's 3D fragments, before the pixels are linked with the predicted 3D locations and a variant of the PnP-RANSAC algorithm is used to estimate the final 3D pose. CDPN [9] disentangles 3D rotation and translation for estimation, they propose a coordinates-based method to estimate rotation by using PnP from the predicted 2D-3D correspondence, and the translation is then estimated directly from the local image patches. Pixel2pose [20] adopts an image-conditional Generative Adversarial Network (GAN) to first estimate the 3D coordinates, and then the pixel-wise predictions are used to form 2D-3D correspondence and finally compute object 3D pose via a RANSAC-based PnP algorithm. DPOD [5] regresses the object ID mask and 2D-3D correspondence map through an encoder-decoder network, and then the 3D pose is computed based on PnP and RANSAC.

Despite indirect methods generally performing well, since they can obtain prior 2D-3D correspondence information from the object's 3D model, they require additional computation for setting up either 2D-3D keypoints [6], [10], [38] or pixel-wise dense correspondence [4], [5], [9], [20]. In summary, they need an accurate 3D model to acquire such information.

B. DEEP LEARNING-BASED DIRECT METHODS

The deep learning-based direct methods (direct methods in short) directly regress 3D pose from CNNs [1], [2], [3], [7], [8], [22]. For instance, SSD6D [22] converts 3D pose estimation into a classification task which discretizes the 3D space with a set of classifiable viewpoints by rendering all possible views and in-plane rotations. This approach requires post-refinement as the initial prediction is only a rough approximation of pose discretization. PoseCNN [3] regresses object 3D pose directly from a customized detection pipeline by estimating projective distance, 2D projection of object center, and rotation represented by a quaternion in each region of interest (ROI). Wu et al. [8] employ a segmentation network to produce segmentation masks and then use a modified ResNet-18 [39] as a pose interpreter network which predicts the 3D pose for each instance. EfficientPose [7] modifies a detection network architecture (EfficientDet) and adds two subnetworks to predict the rotation and translation respectively for each anchor box. DeepIM [1] implements an iterative refinement where the proposed network predicts a relative pose transformation between the input image and the rendered image of its predicted pose. CosyPose [2] extends DeepIM with a novel six-dimensional rotation representation and a more advanced network architecture, and won the BOP challenge 2020 (BOP20 in short) [14].

Except for SSD6D, the above direct methods require accurate 3D model in network training. They either use the model point-based loss like Shape-Match loss [3], or establish iterative refinement from the object's 3D model. In practice, it is difficult to build an accurate 3D model for every possible object of interest.

C. FROM LATENT REPRESENTATION TO OBJECT CHARACTERISTICS

Different from the deep learning-based indirect/direct methods, AAE [19] leverages a denoising autoencoder to learn an implicit rotation representation from cropped objects, and then builds a codebook from the latent variables. At inference time, rotation is estimated using the K-Nearest Neighbors (KNN) with the highest cosine similarity from the codebook. The extended approach, Multi-Path [23], also proves that with multiple decoders, the latent representation can be shared through different objects. PoseRBPF [40] also maps the latent space to a codebook but estimates pose by computing observation likelihoods.

Both AAE and PoseRBPF can handle occluded objects due to strong data augmentations. However, they assign the most likely pose to the instance, which leads to notable errors, due to the discretization of 3D rotation. Besides, they only focus on 3D pose rather than other attributes of the object, which are also important for specific tasks. In contrast, the proposed CVML-Pose aims to construct an informative latent space which can be mapped not only to object 3D pose, but also to other characteristics, e.g. object categorization and shape topology. In terms of object characterization, most of the existing 3D pose estimation approaches are restricted to object 3D pose only, and have to be retrained or redesigned if other attributes of objects are needed. In this case, the latent space of our proposed CVML-Pose pipeline has the potential to be extended to multiple tasks without any retraining of the main network (less computational cost). The object characterization tasks, including object class and genus, will be explained in Sec. III-D. The latent representation based AAE and Multi-Path methods are compared against the proposed CVML-Pose method in the results section. Although the proposed CVML-Pose is not the first approach to use latent representation for 3D pose estimation, it is demonstrated that our method outperforms AAE, Multi-Path, and AAE-ICP (AAE using iterative closest point refinement and depth information) by some margin when testing on the more challenging LineMod-Occlusion data. PoseRBPF is not included in that comparison as the authors did not report results on LineMod/LineMod-Occlusion, nor participated in the BOP challenges.

D. THE BOP CHALLENGE AND PHYSICALLY BASED RENDERING (PBR)

The BOP challenge [13], [14] is a series of well-organized competitions in object 3D pose estimation. During the BOP challenge 2019 (BOP19 in short), classical point pair methods [31], [41] still outperformed deep learning-based methods [5], [9], [19], [20]. Based on the BOP19 results, Hodaň et al. [14] pointed out the two main problems for deep learning-based approaches: (i) insufficient number of real training images with annotation; (ii) large domain gap between real test images and commonly used synthetic training images (objects rendered on random backgrounds).

TABLE 1. Approaches that benefit from the LineMod PBR images in the BOP20. Each approach has two variants with different training data, the performance score AR_{score} is calculated from the average recall of three pose error functions in BOP20 [14] and tested on the BOP version of the LineMod-Occlusion data [17], [18]. Non-PBR images refer to synthetic images with no PBR techniques. As an example, CDPN achieves $AR_{score} = 0.569$ by using only PBR images, while it gets $AR_{score} = 0.374$ with non-PBR images.

Method name	PBR images	non-PBR images	real images	AR_{score}
CDPN	✓			0.569
		✓		0.374
CDPNv2	✓			0.624
	✓		✓	0.624
EPOS	✓			0.547
		✓		0.443
CosyPose	✓			0.633
	✓	✓	✓	0.633

BOP20 additionally provided 350K physically based rendering (PBR) images [14], [42] in the training data, in order to reduce the severe domain gap between synthetic training and real test images. From BOP20, deep learning-based methods have caught up with point pair methods. Some approaches and their results reported in the BOP20 are shown in Table 1: CDPN [9] and EPOS [4] have an obvious improvement when using only the PBR images, compared to when they use only non-PBR images (their own synthesized images with non-PBR techniques). CosyPose [2] and CDPNv2 [9] achieve competitive results with PBR images only.

We choose to use the LineMod PBR images instead of real training images because, as demonstrated in Table 1, the PBR training images provide good enough generalization for the network to perform well on real test images. Moreover, a simple rendering software (non-PBR techniques) called Pyrender[3] is used to synthesize output ground truth images with a clean background for the CVML-AE module training. These images are based on the ground truth 3D pose and camera intrinsic parameters provided by the LineMod PBR database.

E. VARIATIONAL AUTOENCODER

Variational autoencoders (VAE) have been proposed in a context of generative models, where the objective is to generate a new, typically highly dimensional, data point x with the generation process controlled by a low dimensional latent code z, randomly drawn from a distribution $p(z)$, which is preferably selected in such a way that the corresponding sampling process is simple to implement.

The unknown distribution $p(x)$ is approximated by $p_\theta(x) = \int p_\theta(x|z)p(z)dz$, where parameters θ are selected so x_i from the available training set $\mathcal{D}$ ($x_i \in \mathcal{D} = \{x_i\}_{i=1}^m$) are likely to be drawn from $p_\theta(x)$, i.e. using a maximum likelihood principle. However, computation of $p_\theta(x) = \int p_\theta(x, z)dz$ is typically intractable. Kingma and Welling [43], [44] described a computationally viable approach where the maximum likelihood optimization objective is replaced by the evidence lower

[3]Pyrender: https://github.com/mmatl/pyrender

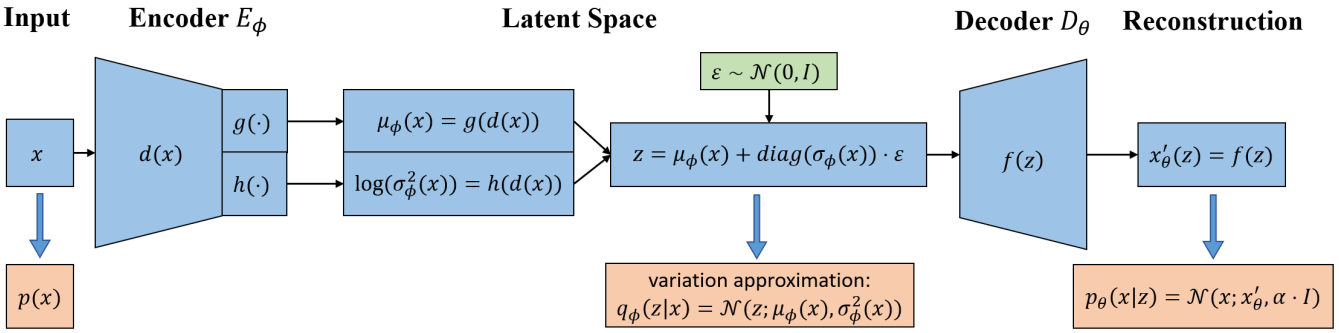

FIGURE 2. General architecture of variational autoencoder.

bound (*ELBO*) and the intractable posterior inference model $p(z|x)$ is replaced by a tractable variational approximation $q_\phi(z|x)$ with parameters ϕ controlling the approximation process.

The overall architecture of the adopted VAE is shown in Fig. 2. In this architecture the encoder network $E_\phi(x)$ estimates optimal parameters $\mu_\phi(x)$ and $\log(\sigma_\phi^2(x))$ of the assumed posterior model $q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi^2(x)) \simeq p(z|x)$ with the prior $p(z) = \mathcal{N}(z; 0, I)$. Since sampling from $\mathcal{N}(z; \mu_\phi(x), \sigma_\phi^2(x))$ is not differentiable, a reparameterization trick $z = \mu_\phi(x) + diag(\sigma_\phi(x)) \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is used to make sure that the sampling stage is differentiable. The decoder network $D_\theta(z)$ estimates optimal parameters θ of the observation model which is assumed to be Gaussian with a diagonal covariance matrix $\alpha \cdot I$, $p_\theta(x|z) = \mathcal{N}(x; x'_\theta(z), \alpha \cdot I)$.

Assuming data points in $\mathcal{D}$ are independent and identically distributed, the *ELBO* is given as:

$$ELBO(\phi, \theta, \mathcal{D}) = \sum_{x_i \in \mathcal{D}} \Big(\mathbb{E}_{q_\phi(z|x_i)} \log p_\theta(x_i|z) - D_{KL}(q_\phi(z|x_i)||p(z)) \Big) \leq \log p_\theta(\mathcal{D}) \quad (1)$$

and the parameters of the encoder and decoder in the VAE network in Fig. 2 are computed using:

$$\hat{\theta}, \hat{\phi} = \arg\max_{\theta,\phi} ELBO(\phi, \theta, \mathcal{D}) \quad (2)$$

With the assumed distribution $p_\theta(x|z)$, $q_\phi(z|x)$, and $p(z)$, the KL divergence $D_{KL}(q_\phi(z|x_i)||p(z))$ has a close form and the *ELBO* in Eq. 2 can be estimated using:

$$ELBO \simeq -c \cdot \sum_{i=1}^{m} \Big(||x_i - x'_i||^2 - \alpha \cdot \sum_{j=1}^{n} \Big(1 + \log(\sigma_{ij}^2) - {\mu_{ij}}^2 - \sigma_{ij}^2\Big)\Big) \quad (3)$$

where c is a positive constant, x_i is the input data, x'_i is the output reconstruction, $x'_i = x'_\theta(z(x_i))$, μ_{ij} refers to the j element of the vector μ_i, σ_{ij}^2 refers to the j element of the vector σ_i^2, $\mu_i = \mu_\phi(x_i)$, $\sigma_i^2 = \sigma_\phi^2(x_i)$, m represents the number of data points in the training set, and n refers to the dimensionality of the latent space. Note that the scalar α weighs the KL divergence, which controls the regularization (smoothness) of the latent space input data representation. The term $||x_i - x'_i||^2$ reflects reconstruction fidelity between the input training image (data point) x_i and the corresponding output reconstructed image x'_i. For the detailed derivations of VAE, please see [43], [44], [45].

Compared to many deep learning-based regression methods for pose estimation [3], [7], [8], VAE has several advantages. For instance, the autoencoder architecture can be trained to output the reconstruction image with a clean background, which means the latent space will only focus on representing the object itself instead of the background. Even if an object is occluded in the input image, the decoder is trained to output the complete object, which allows the model to learn to deal with occlusion. It is also demonstrated in the results section that our method achieves comparable results to the state-of-the-art on LineMod-Occlusion data, despite not using the object's 3D model. Besides, compared to a similar method AAE [19] that uses a denoising autoencoder, the KL divergence with the reparameterization trick in VAE helps to regularize the latent space to combat over-fitting. From the results in [43], the trained VAE has the ability to reconstruct new digits from the probability distribution of its latent space. We, therefore, postulate that the representation learned by a VAE can be adopted for the estimation of object attributes including pose, category, and topology.

III. METHODOLOGY

The proposed CVML-Pose network is trained to regress object 3D pose including 3D rotation $\mathbf{R} \in SO(3)$ and 3D translation $\mathbf{T} = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}^T \in \mathbb{R}^3$ in two steps. During training, the target objects are first cropped from the scene with the ground truth bounding box and resized as the input images, then the CVML-AE module is trained to reconstruct the images via a low dimensional latent representation. This requires the CVML-AE module to implicitly learn to represent images with a relatively small number of high-level features, which can then be used for downstream tasks. In the second step, the estimation of $\mathbf{R}$ and $\mathbf{T}$ are disentangled by adopting MLPs and the KNN algorithm. For the estima-

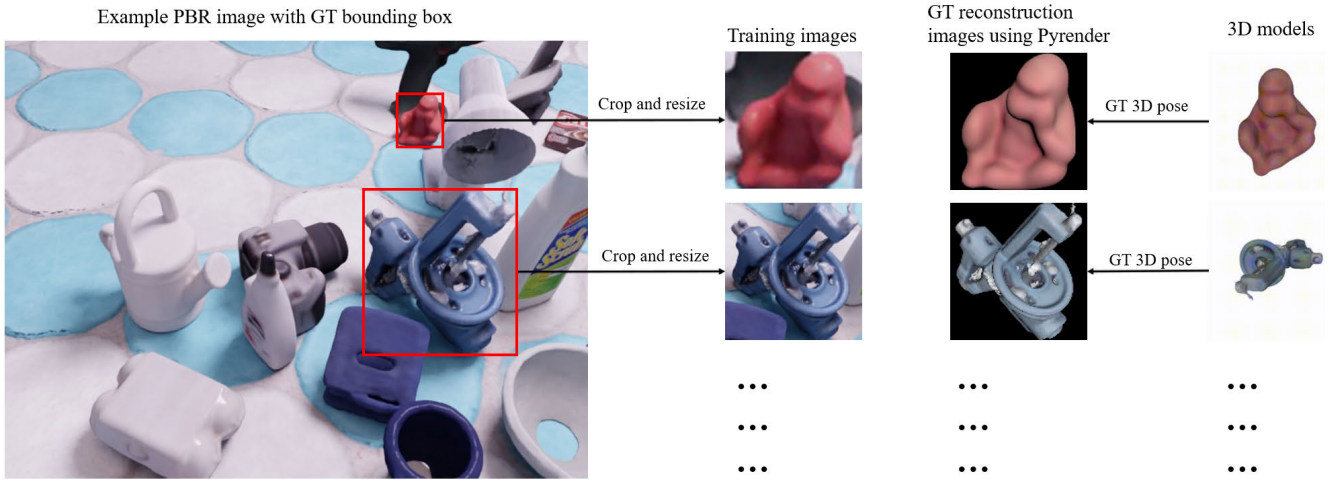

FIGURE 3. Example Data preprocessing based on the LineMod [16] objects. GT stands for ground truth, and the ground truth information (3D pose, bounding box, object visibility, etc.) is provided by the PBR database [42].

tion of **R**, a simple MLP is employed to regress the latent variables to a recently proposed six-dimensional continuous rotation representation [46]. To estimate **T**, the same latent variables with added information about the object bounding box are combined together to estimate the 2D projection of object center and projective distance using another MLP and a KNN regressor. To investigate whether the latent space represents other object characteristics, the CVML-base network is trained on multiple objects simultaneously and their corresponding test images are forwarded to the trained network to generate test latent variables. By analyzing the latent variables computed for the test images with the t-SNE algorithm [21], it can be concluded that it is possible to use the proposed architecture to infer other characteristics, such as object class and/or topology. The remainder of this section describes: data preparation, training procedure of the CVML-AE module, methods for inferring object 3D pose, latent space visualization, implementation details, and ablation tests.

A. DATA PREPARATION

As explained in Sec. II-D, since the LineMod PBR images are adopted for training based on the reported BOP20 results, they enable a comparable network performance to one trained on real images. Each target object is cropped into a square shape from the PBR images using the provided ground truth bounding box and then resized to $128 \times 128 \times 3$ (with bicubic interpolation) to fit the network input size. The cropping size is defined as the longer side l of the bounding box. The object's resize factor $s = 128/l$ is kept for further calculation of the object 3D translation. Within the LineMod PBR dataset, there are approximately $50k$ images per object with the ground truth bounding box. Since Hodan et al. [13] only evaluate objects with visibility of at least 10% area in the scene, we use the same criterion and finally select around $40k$ images per object as the input x. For each object, 90% of images are randomly assigned to the training set and the remaining 10% to the validation set.

Besides, Pyrender is used to additionally synthesize ground truth (GT) reconstruction images $\hat{x}$ with a clean background based on the ground truth 3D pose and camera intrinsic parameters provided by the LineMod PBR database. Fig. 3 illustrates how the LineMod PBR images and ground truth reconstruction images are prepared. Please notice that the ground truth reconstruction images $\hat{x}$ are different from the input images x of the object of interest, as shown in Fig. 4, $\hat{x}_i$ shows a complete object without any background or occlusions, which could be possibly present in the original input image x_i.

For evaluation, the original LineMod and the BOP version of the LineMod-Occlusion test data are used. They are processed with the same crop-and-resize strategy based on the bounding box from the detection model. For more details about the evaluation procedure, please refer to Sec.IV.

B. LATENT REPRESENTATION FROM 2D IMAGES

To learn object latent attributes, the first part of the CVML-Pose is implemented by using a symmetrical encoder-decoder network as depicted in Fig. 4. To avoid zero gradients, the ELU activation function is adopted rather than the commonly-used ReLU activation. To compare different network architectures, the decoder network remains unchanged but the encoder is replaced with more advanced networks such as ResNet-18 and ResNet-34 [39] (with ELU activation). To distinguish different variants, the vanilla symmetrical encoder-decoder network is named CVML-base, and the ResNet-based autoencoders are named CVML-18 and CVML-34 respectively. All these autoencoder networks are collectively called CVML-AE module. In terms of training the module, the encoder is trained to form the latent space from cropped images with various online augmentations, which are reported in Table 2. The idea, detailed explanation,

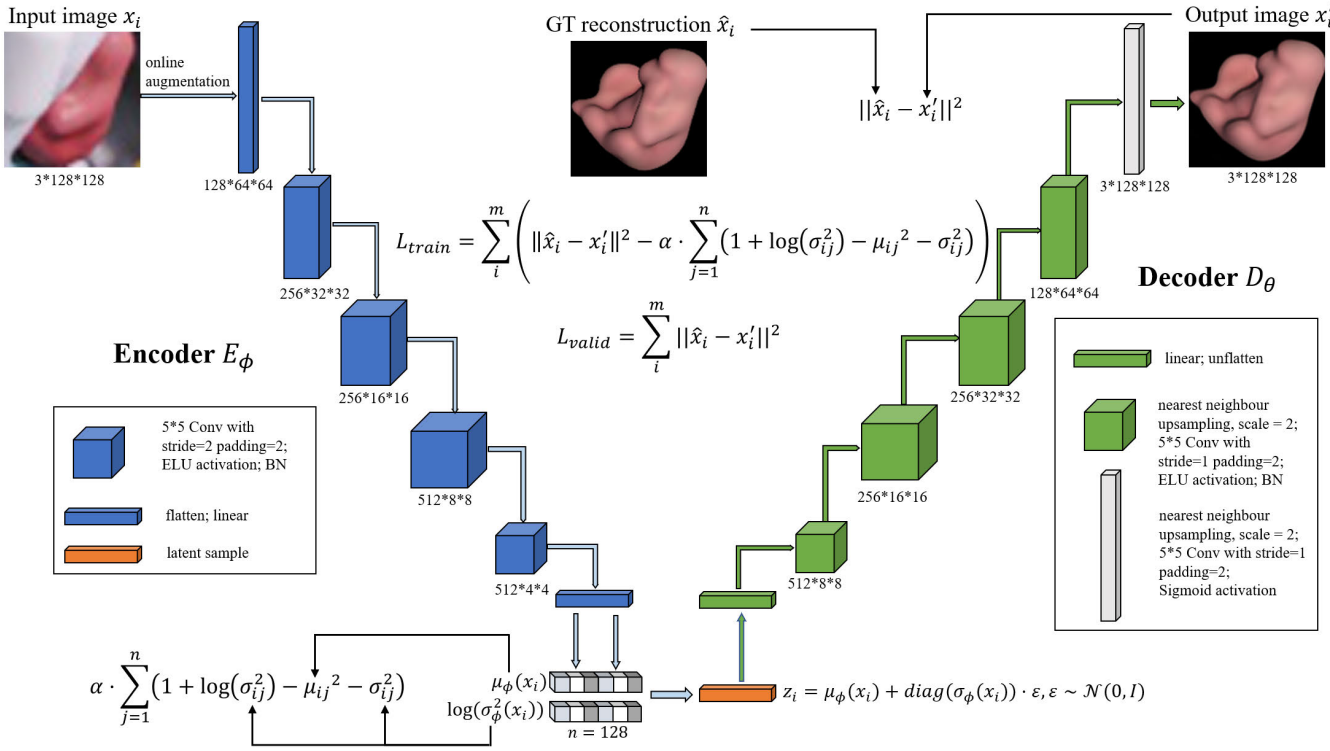

FIGURE 4. The vanilla CVML-base architecture. ϕ represents all the parameters of the encoder network (represented in blue) and θ represents all the parameters of the decoder network (shown here in green). The encoder part can be further replaced with ResNet-18 and ResNet-34 backbones [39]. The input image is first forwarded to the encoder which produces the latent variables. The decoder then generates the output reconstruction image from the latent sampling. The training loss L_{train} for the CVML-AE module is modified from Eq. 3, and the validation loss L_{valid} is the pixel-wise L2 loss.

TABLE 2. Proposed online augmentation details. The proposed augmentations are implemented with a 30% possibility during training the CVML-AE module.

Augmentation	Parameters	Value
ColorJitter	brightness	$[0.4, 2.3]$
	contrast	$[0.4, 2.3]$
	saturation	$[0.8, 1.2]$
GaussianBlur	kernel size	$(5, 5)$
	sigma	$[0.1, 1.2]$
RandomAffine	translate	$(-0.1, 0.1)$
	scale	$(0.9, 1.1)$
RandomErasing	scale	$(0.1, 0.2)$
	ratio	$(0.3, 3.0)$
	value	random

and comparison of different variants of the autoencoder networks can be found in Sec. III-F.

As shown in Fig. 4, given the input data x_i, latent variables $(\mu_\phi(x_i), \sigma_\phi^2(x_i))$ are produced by the encoder $E_\phi(x_i)$. The KL divergence with a weight α is used to regularize the latent space. The dimensionality of the latent space is represented as n. In [19], the authors of the AAE method, which also used latent space representation, implemented ablation tests to find an optimal size of the latent space. In that case, the pose estimation accuracy started to saturate when $n = 64$, and AAE achieved the best results when $n = 128$. Moreover, although the results are not reported in this paper, we tested different latent space sizes and got the best results when $n = 128$ as well. Thus, $n = 128$ is set for all the autoencoder networks in the CVML-AE module. After sampling in the latent space, the decoder network $D_\theta(z_i)$ outputs the reconstruction image x_i' from the latent sample $z_i = \mu_\phi(x_i) + diag(\sigma_\phi(x_i)) \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, and a pixel-wise L2 loss is computed between the output image x_i' and the ground truth reconstruction image $\hat{x}_i$, i.e. term $||x_i - x_i'||^2$ in *ELBO* (see Eq. 3) is replaced with $||\hat{x}_i - x_i'||^2$, and the constant c is removed. In the provided code implementation,[4] the training loss $L_{train} = -ELBO$ is iteratively minimized using AdamW optimizer algorithm with randomly selected mini-batches of 128 elements (i.e. m in Eq. 3 is replaced with 128). To prevent over-fitting in training, the same pixel-wise L2 loss is used as the validation loss L_{valid} (no KL divergence is included in the L_{valid}, i.e. $\alpha = 0$), indicating whether the latent space has accumulated enough information from the training images.

C. ESTIMATE OBJECT 3D POSE FROM THE LATENT SPACE

After training the CVML-AE module, the latent space accumulates rich information of the special orthogonal group in 3D (abbreviated as SO(3)). Fig. 5 depicts the training procedure to estimate the complete 3D pose based on the MLPs and KNN after the training of the encoder is completed. To estimate the rotation matrix $\mathbf{R} \in \text{SO}(3)$, the latent variables $\mu_{train} \in \mathbb{R}^{128}$ of the training data are generated by the trained

[4]Code available: https://github.com/JZhao12/CVML-Pose

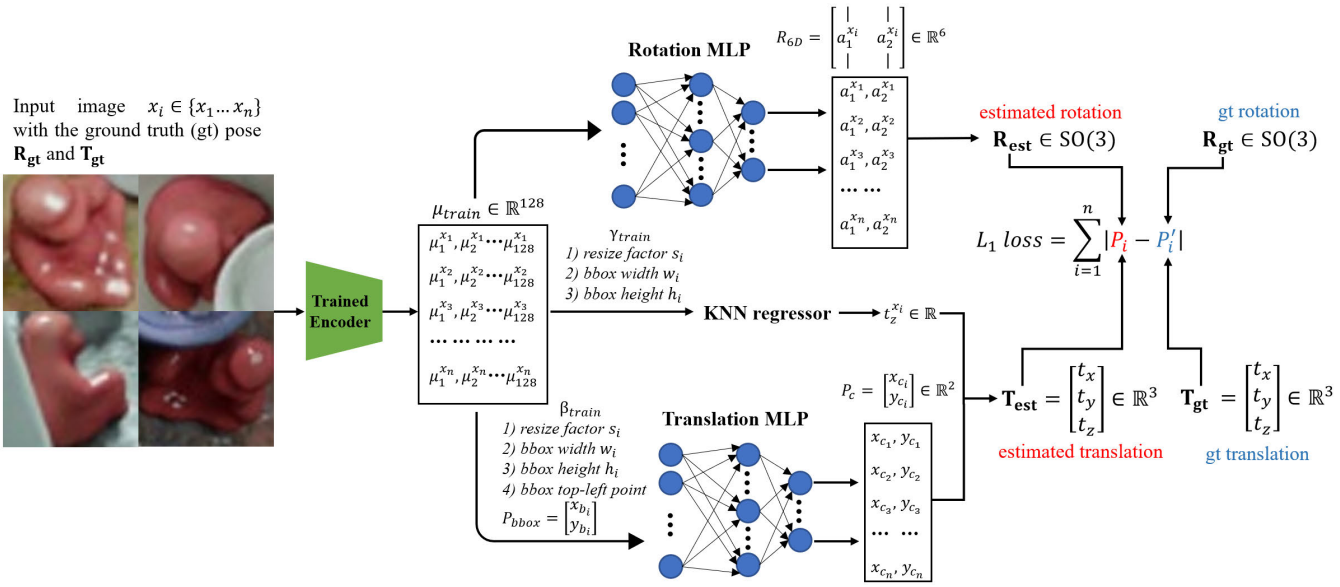

FIGURE 5. Latent space training process. Example input images from the LineMod ape object are used to produce the training latent variables $\mu_{train} \in \mathbb{R}^{128}$ through the trained encoder. The MLPs and KNN are utilized to regress the complete 3D pose, and the training loss L_1 for MLPs is calculated between the estimated pose P_i and the ground truth pose P_i', where the ground truth pose is provided by the LineMod PBR dataset.

encoder, then a simple MLP (named the Rotation MLP) is trained to regress the latent variables to a recently proposed six-dimensional rotation representation $R_{6D} \in \mathbb{R}^6$ introduced by Zhou et al. [46]. The authors demonstrated that this new rotation representation outperforms other previously used representations, and it was successfully used by CosyPose [2] (winner of the BOP20). The transformation between the rotation matrix $\mathbf{R} \in SO(3)$ and the six-dimensional representation $R_{6D} \in \mathbb{R}^6$ is written in Eq. 4:

$$\mathbf{R} = \left(\begin{bmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{bmatrix} \right) \longrightarrow R_{6D} = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix},$$
$$R_{6D} = \left(\begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix} \right) \longrightarrow \mathbf{R} = \begin{bmatrix} | & | & | \\ b_1 & b_2 & b_3 \\ | & | & | \end{bmatrix} \tag{4}$$

where $b_i = \begin{bmatrix} N(a_1) & i = 1 \\ N(a_2 - (b_1 \cdot a_2)b_1) & i = 2 \\ b_1 \times b_2 & i = 3 \end{bmatrix}$, N represents normalization, $\times$ denotes cross product between two vectors, a_1, a_2, a_3 are the three columns of the rotation matrix $\mathbf{R}$. The detailed derivations can be found in [46].

To estimate 3D translation $\mathbf{T} = (t_x\ t_y\ t_z)^T \in \mathbb{R}^3$, the 2D projection of object center $P_c = (x_c, y_c)^T \in \mathbb{R}^2$ and projective distance $t_z \in \mathbb{R}$ are regressed separately, and are then used to calculate t_x and t_y based on the projective camera model (Eq. 5 and 6). For the estimation of P_c, the latent variables $\mu_{train} \in \mathbb{R}^{128}$ are concatenated with the corresponding prior information β_{train} which includes the object's resize factor s, width w, height h, and top left corner $P_{bbox} = (x_b, y_b)^T$ of the ground truth bounding box. P_c is then localized by implementing another simple MLP (named the Translation MLP) with these concatenated variables. Afterwards, a KNN regressor is trained to regress t_z with the latent variables μ_{train} and γ_{train} which consist of s, w, and h, where t_z is predicted by local interpolation of the targets associated with $K \in [1, 20]$ nearest neighbors in the training set. For each value of K, we calculate the validation error based on the complete 3D translation, instead of the error of t_z, and the final K can be determined where the validation error is the smallest. After knowing t_z and $P_c = (x_c, y_c)^T$, the first two elements of the 3D translation $\mathbf{T} = (t_x\ t_y\ t_z)^T \in \mathbb{R}^3$ can be further calculated based on Eq. 6.

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} f_x \frac{t_x}{t_z} + c_x \\ f_y \frac{t_y}{t_z} + c_y \end{bmatrix} \tag{5}$$
$$\begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} (x_c - c_x)\frac{t_z}{f_x} \\ (y_c - c_y)\frac{t_z}{f_y} \end{bmatrix} \tag{6}$$

where f_x and f_y denote the focal lengths, $(c_x, c_y)^T$ is the principal point.

D. OBTAIN OTHER CHARACTERISTICS FROM THE LATENT SPACE

This section reports on our experiments, which investigate if the latent space can be used to represent and subsequently estimate other characteristics of objects. In the experiments reported here, both shape topology (as described by a genus) and object class were investigated. An intuitive way to introduce the concept of a genus is that it represents the number of "holes" an object has [47]. For instance, a sphere has genus 0, and a torus has genus 1. Considering, for example, a robotic arm tasked with grasping a mug, it would be beneficial for the system to differentiate between mugs with and without

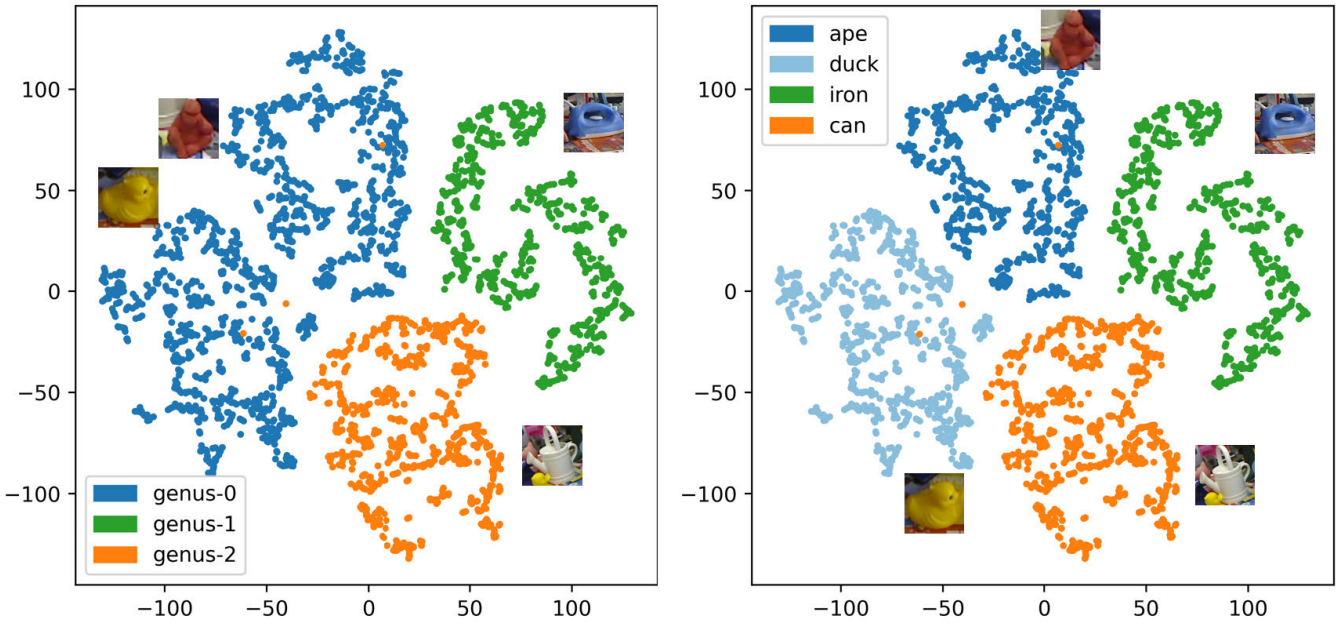

FIGURE 6. t-SNE visualization of the latent space for the purpose of both topology recognition and object classification. The latent space is trained with the LineMod PBR images and the latent variables are generated from the LineMod test images. The selected objects are LineMod ape, duck, iron, and can.

a handle. Thus, to investigate the hypothesis from Sec. II-E that VAE can learn other latent attributes, the CVML-base network is trained to encode four LineMod objects with three different genera: ape and duck (genus 0), iron (genus 1), and can (genus 2). After training the CVML-base network, the test images of the selected objects are used to generate corresponding latent variables and the t-SNE [21] feature reduction method is utilized to visualize the variables. Although t-SNE could not be directly used for classification tasks, since new data points would change the spatial distance of the existing points, it is still expedient to notice that the visualization of the latent space in Fig. 6 shows obvious clusters representing different object characteristics. For example, the ape's and duck's latent variables are clustered together, reflecting on the fact that these objects share the same topology. In this specific example, it can be concluded that the learnt latent space represents the objects' class and topology. We can therefore speculate that the latent space in the CVML-AE module is capable of encoding various objects' characteristics, which can be used in robotic object manipulation applications.

E. NETWORK PARAMETERS AND IMPLEMENTATION DETAILS

Table 3 reports the network parameters of the CVML-AE module and MLPs. The training details of the CVML-AE module, the MLPs, and the KNN regressor are also shown in Table 4. The CVML-AE module and MLPs are implemented using PyTorch. The batch size of the CVML-AE module is set to 128, while MLPs take all input variables as a batch. For the CVML-AE module and MLPs, the AdamW optimizer [48] is used with betas = (0.9, 0.999), eps = $1e-8$, weight decay = $1e-2$. The learning rate is scheduled to reduce when the validation error does not improve for certain epochs (which is the Patience in Table 4), with a drop factor $d = 0.5$ on a plateau. To implement the KNN algorithm, a KNN regressor from scikit-learn [49] is adopted with the 'distance' weights, in which case the closer neighbors of a query point will have a greater influence than neighbors which are further away.

Using the training data described in Sec. III-A, two CVML-base networks can be trained in parallel in approximately 8 hours on a single NVIDIA Geforce RTX 3090. CVML-18 takes about 12 hours, and CVML-34 takes around 20 hours on the same device. All MLPs are trained in parallel on the same device, which takes roughly 12 hours.

TABLE 3. Network parameters of the CVML-AE module and MLPs models.

Networks	Number of parameters
CVML-base network	2.7774211×10^7
CVML-18 network	7.9082243×10^7
CVML-34 network	1.12682499×10^8
Rotation MLP	1.0966×10^4
Translation MLP	1.1338×10^4

TABLE 4. Training details of the CVML-AE module, MLPs, and KNN. The term "Patience" here means the learning rate drop patience. The training is terminated when the model reaches the lowest learning rate and the validation loss stops dropping for N number of epochs, $N = 30$ for the CVML-AE module and $N = 1000$ for the MLPs.

Training details	CVML-AE	MLPs	KNN
Training loss	L_{train}	L1	-
Validation loss	L_{valid}	L1	L1
Learning rate	$[10^{-4}, 10^{-6}]$	$[10^{-3}, 10^{-7}]$	-
Patience	30	1000	-

TABLE 5. Ablation test based on different autoencoder network structures.

Network structure	$AP(ADD(S))$
autoencoder from [19]	41.14
CVML-base network	51.13
CVML-18 network	60.12
CVML-34 network	**60.47**

TABLE 6. Ablation studies with various weighting α of the KL regularization term based on the CVML-base network.

Loss function	$AP(ADD(S))$
L_{train} with $\alpha = 1$	45.18
L_{train} with $\alpha = 0.5$	48.29
L_{train} with $\alpha = 0.1$	**51.13**
L_{train} with $\alpha = 0$	45.13

F. ABLATION TESTS

We implement a number of ablation tests for different purposes including network architecture, weight of the KL regularization term, online data augmentation, and scalability of the network. The dimensionality n of the latent space is set to 128 for all autoencoders as explained in Sec. II-C. To evaluate the results of the ablation tests, a pose evaluation metric called $ADD(S)$ [3], [16], [50] is used, and the performance score $AP(ADD(S))$ is calculated from the average precision of the $ADD(S)$ metric. All the results reported in Table 5, 6, 7, 8, and 9 are calculated based on the ground truth bounding box of the test images to reduce dependence of the results on performance of any specific detector. The objects used in the ablation test are ape, driller, eggbox, and phone from the LineMod dataset. For details of the evaluation metrics, please refer to Sec. IV-B.

In the proposed CVML-AE module, a well-structured latent space plays a key role in the interpretation of object 3D pose. When designing the network architecture, we initially use a similar approach to the one used in AAE [19] and build the symmetric CVML-base network. To test the benefit of alternative encoders, the variants CVML-18 and CVML-34 using ResNet-18 and ResNet-34 backbone (with ELU activation) are implemented to see how well each of the CVML-AE module encodes object pose information in the latent space. Additionally, to see how well the proposed CVML-AE module compares against autoencoders used in the existing pose estimation networks, the AAE's autoencoder was used in a comparative test. Table 5 shows each network architecture and their corresponding average performance scores.

Regarding the loss function of the CVML-AE module, the regularization term α directly affects the latent space, since we want both an informative representation of the original data and a good generalization ability. To find a good balance, the CVML-base network is trained with different weighting α of the KL regularization term, where the value of α is set to 0, 0.1, 0.5, and 1. Table 6 demonstrates the results for the CVML-base network with different values of the regularization term α and their average performance scores.

TABLE 7. Ablation studies based on the CVML-base network with different data augmentation methods. The CVML-base network is trained with the LineMod PBR images of the ape, driller, eggbox and phone objects and tested on LineMod.

Online augmentation type	$AP(ADD(S))$
all augmentations	51.13
no ColorJitter	49.47
no GaussianBlur	51.11
no RandomErasing	**53.43**
no data augmentation	50.09

The various types of online augmentation listed in Table 2 were used to train the CVML-AE module. The random change of brightness, contrast, and saturation is used to improve robustness to variation in environmental settings. The Gaussian blur is introduced to minimize the distortion problem of the test camera. The random scale and translation are adopted to improve robustness to variation in the estimated bounding box. We also considered the random erasing technique [51] to add more occlusion to objects' observations, but however omitted this operation from the final augmentation pipeline. To identify whether these online augmentation methods are beneficial in training the CVML-AE module, the CVML-base network is trained with different combinations of the augmentation methods. Table 7 shows the pose estimation performance with different combinations of the considered online augmentation.

From the results reported in Table 5, it can be seen that with increasing depth of the encoder, the latent space acquires more 3D information about the objects. For the CVML-base network, although it has only 1 more layer than the network from [19] in both encoder and decoder, it boosts pose estimation accuracy significantly. However, there is very little difference between the performance of the CVML-18 and the CVML-34 networks, which probably indicates that the decoder from the CVML-base network is not capable of taking full advantage of the rich latent information provided by the ResNet-34 encoder, in which case a deeper decoder network may help. It is also possible that available relevant pose information (based on the available training data) is well enough encapsulated (summed up) by the ResNet-18 encoder. Therefore, more complicated encoders, like ResNet-34, are unlikely to improve the results as there is nothing much to add (with respect to pose) to what the ResNet-18 captured. From the results reported in Table 6, the best value for the regularization weight α found in our experiments was 0.1. In particular, the CVML-base network is no longer a VAE when $\alpha = 0$ (the reparameterization trick is not used), and it performs worse when generalizing to the test images. When $\alpha = 1$, the latent space is greatly affected by the KL divergence, which seems to smooth the distribution too greatly, leading to a loss of information about the object's pose. From the results reported in Table 7, the omission of ColorJitter and GaussianBlur slightly decreases the accuracy on the LineMod test data, however, the network performs better without RandomErasing augmentation. One possible reason is, the

TABLE 8. Additional ablation tests for data augmentation methods based on different test data. The CVML-base network is trained with the LineMod PBR images of the ape, driller, and eggbox objects, and tested on LineMod and the BOP version of LineMod-Occlusion. The LineMod phone object is not included because it does not exist in LineMod-Occlusion (as it was for the results given in Table 7).

	$AP(ADD(S))$	
Online augmentation type	LineMod	LineMod-Occlusion
all augmentations	58.31	23.82
no ColorJitter	56.10	21.32
no GaussianBlur	58.54	23.09
no RandomErasing	**60.45**	**25.25**
no data augmentation	56.25	20.04

TABLE 9. Scalability test with different size of images based on the CVML-base network. Data augmentation is not used in this experiment.

Input image size	$AP(ADD(S))$
$32 \times 32 \times 3$	33.99
$64 \times 64 \times 3$	45.26
$128 \times 128 \times 3$	**50.09**
$256 \times 256 \times 3$	47.13

LineMod PBR images already introduced much occlusion in the training data. Applying the RandomErasing augmentation during training can lead to an object being entirely occluded and the network being trained to predict its pose without any information to infer this from, resulting in destructive updates. To investigate this, an additional experiment on the LineMod-Occlusion test data is implemented and the results are shown in Table 8. Without RandomErasing augmentation, the CVML-base network achieves better results on both LineMod and the BOP version of LineMod-Occlusion test data, which indicates that RandomErasing may not be useful in our proposed approach.

Except for network architectures, data augmentation, and loss functions, the scalability of the CVML-Pose is also considered. For example, to test whether the input image size could significantly affect the network performance, the CVML-base network is trained with images of different sizes $32 \times 32 \times 3$, $64 \times 64 \times 3$, $128 \times 128 \times 3$, and $256 \times 256 \times 3$. Since the input size is changed, the CVML-base network is modified accordingly to fit the size of the input image. Table 9 gives the results for the CVML-base network with different image sizes and their average performance scores. As the size of the input image increases, the pose estimation accuracy starts to saturate at size $128 \times 128 \times 3$. Although the modified CVML-base network is capable of extracting certain 3D information from both smaller ($64 \times 64 \times 3$) and larger ($256 \times 256 \times 3$) images, training with $128 \times 128 \times 3$ images can provide the best results.

Based on all the results in Table 5, 6, 7, 8, and 9, we select the parameters for the main experiments. Despite the CVML-18 network not performing as well as the CVML-34 network, the gap between their pose estimation accuracy is quite narrow, and the CVML-18 has fewer network parameters, which saves training time. With $\alpha = 0.1$, the training loss could guarantee both stabilization and generalization of the network. In this case, we only adopt ColorJitter, GaussianBlur, and RandomAffine (only translation and scale) to train the CVML-18 network for each LineMod object with image size $128 \times 128 \times 3$ and the weighting term α of the training loss is set to 0.1. The final results are presented and discussed in Sec. IV.

IV. EVALUATION

A. OVERVIEW

As explained in Sec. III-F, the proposed CVML-18 network is evaluated on LineMod [16] and the BOP version of the LineMod-Occlusion [17], [18] test images. For evaluation with the LineMod data, we use the same training/test data split strategy as in [6], [10], [38], and [52], but exclude the real training data. In terms of the LineMod-Occlusion test data, since the BOP challenge evaluates only a subset of the original data, the same BOP version of the LineMod-Occlusion data is used to make a fair comparison to the results reported by the BOP challenge.

To enable CVML-Pose to operate on custom images, an object detector needs to be used first, as it is not built into the CVML-Pose network. In the experiments reported in this section, a Mask-RCNN [53] detector is adopted. The detector was trained on the LineMod PBR images as used by CosyPose [2]. During inference, the detection bounding box from Mask-RCNN is used to crop the object of interest into a square shape. Table 12 reports the detected number of objects in each test dataset. After object detection, the cropped image is resized to $128 \times 128 \times 3$ and fed into the trained encoder network which produces a set of test latent variables. These variables are finally forwarded to the trained MLPs and KNN to predict the complete 3D pose of the test objects as depicted in Fig. 7. To see how object detection can affect pose estimation, the ground truth bounding box of the test data is also used to make a comparison with pose estimation results based on the Mask-RCNN detector.

Besides, in the paper supplementary material, a video is provided showing a real-time pose estimation. The video demonstrates that the proposed CVML-Pose works well with different levels of object occlusions and cluttered scenes, with low-resolution images captured with an inexpensive webcam, under no specific illumination or any constraints to the object's motion. In terms of the heterogeneity mentioned in Sec. I, hardware heterogeneity is also considered. For example, the real data used for evaluation are captured by a Microsoft Kinect camera, which are high-resolution images, while the real data in the video are captured by a Logitech C270i webcam, and they are low-resolution images. The method can cope well with heterogeneous data (different images of objects, high-resolution and low-resolution images) and hardware (expensive camera and inexpensive webcam).

In the reported experiments, three popular metrics have been adopted to evaluate the pose estimation results: $ADD(S)$, *MSSD*, and *MSPD* (defined in Sec. IV-B). Although object's

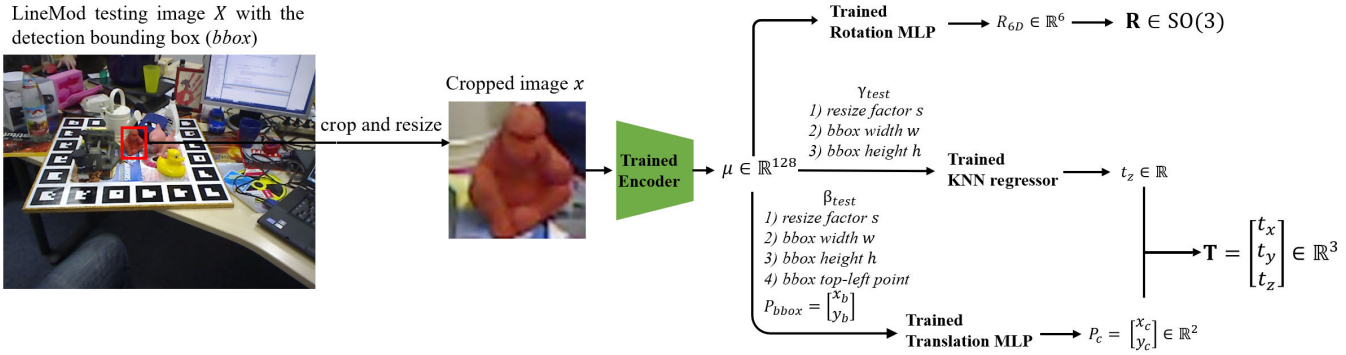

FIGURE 7. Inference procedure. During inference, an object of interest is first detected by the detector (Mask R-CNN) and then cropped by using the detection bounding box. The cropped test image x is resized to fit the trained encoder which generates the test latent variables $\mu \in \mathbb{R}^{128}$. Finally the trained MLPs and KNN regressor estimate object's 3D rotation $\mathbf{R} \in SO(3)$ and 3D translation $\mathbf{T} = (t_x \ t_y \ t_z)^T \in \mathbb{R}^3$ from the latent variables.

3D model is not used in our proposed approach, the three evaluation metrics all require object's 3D model points, and the model points used in evaluation are a subset of points from the original object's 3D model. We report the performance on the LineMod test data based on the *ADD*(*S*) metric as it is widely used in many state-of-the-art [5], [6], [9], [19], [20], [22]. The performance on the LineMod-Occlusion is evaluated based on the *MSSD* and *MSPD* metrics since the BOP challenge uses the two metrics for evaluation and [2], [4], [5], [6], [9], [19], [20], [22], [23] have reported their results on the leaderboard. We benchmark the CVML-Pose in two ways, the first is to compare different network configurations, i.e. ablation tests reported in Sec. III-F. The second way is to follow the evaluation methodology proposed in the BOP challenge and compare our results to the state-of-the-art approaches. The rest of this section explains the three evaluation metrics and presents our evaluation results on the LineMod and the LineMod-Occlusion datasets, together with detailed discussions.

B. EVALUATION METRICS

The *ADD*(*S*) [3], [16], [50] metric is used for the evaluation of LineMod which have two variants: the *ADD* and the *ADD-S*. The *ADD* metric calculates the mean value of the point pair distances between the transformed 3D model points that have no indistinguishable views (non-symmetric), while the *ADD-S* metric is computed by using the closest point distance if the object is symmetrical. For the LineMod objects, glue and eggbox are considered as symmetric objects which use the *ADD-S* metric for evaluation, while the remaining objects are non-symmetric objects which use the *ADD* metric. The estimated pose is considered correct when the *ADD*(*S*) is smaller than a specific criterion of the 3D model diameter. We follow the state-of-the-art approaches [6], [7], [10], [38] and set the criterion as 10%, and report the average precision *AP*(*ADD*(*S*)) in Table 10 which is calculated from the mean accuracy based on the *ADD*(*S*) metric for all objects. The *AP*(*ADD*(*S*)) is not reported in Table 11 because the BOP challenge does not use the *ADD*(*S*) metric for evaluation, however we still use it to compare the results based on the detection bounding box and the ground truth bounding box in Table 13.

$$ADD = \frac{1}{m} \sum_{x \in M} ||(Rx + T) - (\hat{R}x + \hat{T})|| \tag{7}$$

$$ADD\text{-}S = \frac{1}{m} \sum_{x_1 \in M} \min_{x_2 \in M} ||(Rx_1 + T) - (\hat{R}x_2 + \hat{T})|| \tag{8}$$

where R and T is the ground truth pose, $\hat{R}$ and $\hat{T}$ is the estimated pose, m is the number of points and M depicts the 3D points of the model.

In addition, two metrics *MSSD*[5] and *MSPD*[6] proposed by [14] are used to evaluate the BOP version of the LineMod-Occlusion [17], [18]. *MSSD* is related to robot manipulation as the robotic grasping is heavily dependent on the maximum surface deviation. *MSPD* is suitable for augmented reality applications because it only evaluates the perceivable discrepancy in the projective space. For a fair comparison, we follow the same evaluation criterion of the BOP challenge and report our results on the BOP version of the LineMod-Occlusion in Table 11 by calculating the average recall rates AR_{MSSD} and AR_{MSPD} for different error thresholds ranging from 5% to 50%, i.e. calculate the metrics based on all error thresholds then compute the average.

$$MSSD = \min_{S \in S_M} \max_{x \in V_m} ||\hat{P}x - \bar{P}Sx||_2 \tag{9}$$

$$MSPD = \min_{S \in S_M} \max_{x \in V_m} ||proj(\hat{P}x) - proj(\bar{P}Sx)||_2 \tag{10}$$

where S_M is a set of global symmetry transformations of the target object, V_m denotes a set of model vertices, *proj* is the 2D projection in pixels, $\hat{P}$ is the estimated pose, and $\bar{P}$ is the ground truth pose. The global object symmetries can be obtained from the Hausdorff distance, the detailed derivations for which can be found in [14].

C. RESULTS AND DISCUSSION

Based on the LineMod results reported in Table 10, the proposed CVML-18 network outperforms AAE [19] and

[5] *MSSD* stands for Maximum Symmetry-Aware Surface Distance.
[6] *MSPD* stands for Maximum Symmetry-Aware Projection Distance.

TABLE 10. Average precision of the *ADD(S)* metric evaluated on the LineMod objects. Our results are based on the CVML-18 network which combines a modified ResNet-18 (encoder) and the vanilla decoder shown in Fig. 4. The glue and eggbox are considered as symmetric objects, bowl and cup are not included due to the reported problems with their 3D models in the LineMod dataset, leading to inaccurate computations of the *ADD(S)* metric. Since Multi-Path [23], EPOS [4], and CosyPose [2] do not provide results on the LineMod test data, we only report their LineMod-Occlusion results in Table 11. Results of other approaches are taken from [5], [6], [9], [19], [20].

	Latent representation based methods			Direct method	Indirect methods			
Objects	CVML-18	AAE [19]	AAE-ICP [19]	SSD6D [22]	DPOD [5]	Pix2Pose [20]	CDPN [9]	PVNet [6]
ape	26.64	4.18	24.35	2.6	37.22	58.1	64.38	43.62
benchvise	52.47	22.85	89.13	15.1	66.76	91.0	97.77	99.90
cam	25.68	32.91	82.10	6.1	24.22	60.9	91.67	86.86
can	50.05	37.03	70.82	27.3	52.57	84.4	95.87	95.47
cat	36.20	18.68	72.18	9.3	32.36	65.0	83.83	79.34
driller	47.81	24.81	44.87	12.0	66.60	76.3	96.23	96.43
duck	24.44	5.86	54.63	1.3	26.12	43.8	66.76	52.58
eggbox	77.41	81.00	96.62	2.8	73.35	96.8	99.72	99.15
glue	55.08	46.17	94.18	3.4	74.49	79.4	99.61	95.66
holepuncher	21.05	18.20	51.25	3.1	24.50	74.8	85.82	81.92
iron	54.85	35.05	77.86	14.6	85.02	83.4	97.85	98.88
lamp	54.13	61.15	86.31	11.4	57.26	82.0	97.89	99.33
phone	28.50	36.27	86.24	9.7	29.08	45.0	90.75	92.41
$AP(ADD(S))$	42.64	32.63	71.58	9.1	50	72.38	89.86	86.27

TABLE 11. Average recall rate of the *MSSD* and the *MSPD* metrics based on the BOP version of the LineMod-Occlusion data. Our results are based on the CVML-18 network. Results of other approaches are from the BOP challenge leaderboard, results for individual object are not reported here since the challenge only provides the results for all objects. Please note CVML-18, CosyPose, Pix2Pose, CDPN, CDPNv2, EPOS and PVNet all use the LineMod PBR images, CDPNv2 refers to the extended version of CDPN, CosyPose uses object's 3D model during training for iterative refinement. For more comprehensive comparison and results, please refer to the BOP20 [14] or the BOP challenge leaderboard: https://bop.felk.cvut.cz/leaderboards/.

	Latent representation based methods				Direct methods		Indirect methods					
Metric	CVML-18	AAE [19]	AAE-ICP [19]	Multi-Path [23]	SSD6D [22]	CosyPose [2]	DPOD [5]	Pix2Pose [20]	CDPN [9]	CDPNv2 [9]	EPOS [4]	PVNet [6]
AR_{MSSD}	0.338	0.095	0.218	0.153	0.083	0.606	0.126	0.307	0.537	0.612	0.501	0.543
AR_{MSPD}	0.706	0.254	0.285	0.346	0.285	0.812	0.278	0.550	0.779	0.815	0.750	0.754

SSD6D [22], the only other approaches which don't use the object's 3D model. It even achieves a slightly better result on the specific LineMod objects (ape and driller) than AAE-ICP, which uses depth information and ICP refinement (3D model points), AAE-ICP may be limited on the basis that the depth measurement is unavailable under some lightning conditions, and the iterative ICP refinement cannot guarantee real-time processing. Compared to the results for typical indirect methods [5], [6], [9], [20] given in Table 10, the CVML-18 network is not as accurate as the indirect methods on most of the LineMod objects. However, it still performs slightly better than DPOD [5] on the LineMod cam, cat, and eggbox. From the LineMod-Occlusion results reported in Table 11, the CVML-18 network outperforms all the latent representation based methods, and it is significantly better than AAE-ICP that uses depth information and ICP refinement. It even outperforms some approaches using pixel-wise dense correspondence (DPOD, Pix2Pose) and achieves comparable results to EPOS [4]. The CVML-18 network ranks better on the more challenging LineMod-Occlusion test data, while it shows a moderate rank on the LineMod test data. One possible reason for this is that approaches like DPOD and Pix2Pose predict the pixel-wise dense correspondence, which means they can perform well when the object of interest is relatively complete in the scene since they can compute enough 2D-3D dense predictions. However, when objects are occluded, the performance starts to decline due to an insufficient number of points available for assessing correspondence. In contrast, the proposed CVML-18 network benefits from the autoencoder architecture that could handle occluded objects robustly, i.e. the network is trained to reconstruct a whole object from the occluded views, and the interpretation methods of the latent space do not discretize the 3D rotation, which makes the system more robust than similar methods [19], [23] using latent representation.

Based on the LineMod results, it can be concluded that the latent representation based method AAE and the direct

TABLE 12. Number of test instances in the LineMod and the BOP version of the LineMod-Occlusion datasets. The original objects are cropped from the ground truth bounding box provided by the BOP challenge, while the detected objects come from the Mask-RCNN detector. The BOP version of the LineMod-Occlusion data have only 8 objects.

	LineMod test		BOP LineMod-Occlusion	
Objects	original	detected	original	detected
ape	1050	1021	187	141
benchvise	1031	1031	-	-
cam	1020	954	-	-
can	1016	1015	199	179
cat	1002	1000	196	139
driller	1009	1006	200	194
duck	1065	1064	188	180
eggbox	1065	974	193	116
glue	1036	1033	154	133
holepuncher	1051	1050	200	199
iron	979	979	-	-
lamp	1042	1042	-	-
phone	1041	1021	-	-

TABLE 13. Average precision of the *ADD(S)* metric evaluated on the LineMod and the BOP version of the LineMod-Occlusion with ground truth bounding box (gt bbox for short) and detection bounding box (detection bbox for short). Results are based on the CVML-18 network. The glue and eggbox are considered as symmetric objects, bowl and cup are not included due to the reported problems with their 3D models in the LineMod dataset, leading to inaccurate computations of the *ADD(S)* metric. The number of test instances can be found in Table 12.

	LineMod test		BOP LineMod-Occlusion	
Objects	gt bbox	detection bbox	gt bbox	detection bbox
ape	34.38	26.64	16.58	8.51
benchvise	70.71	52.47	-	-
cam	45.00	25.68	-	-
can	62.40	50.05	33.67	18.44
cat	41.92	36.20	11.73	10.07
driller	64.52	47.81	40.50	20.10
duck	33.52	24.44	22.87	13.33
eggbox	90.89	77.41	31.09	16.38
glue	57.43	55.08	35.71	40.60
holepuncher	30.26	21.05	25.5	20.60
iron	57.20	54.85	-	-
lamp	66.12	54.13	-	-
phone	43.23	28.50	-	-
$AP(ADD(S))$	53.66	42.64	27.21	18.50

method SSD6D perform significantly worse than the indirect methods. For instance, even with the help of additional depth information and ICP refinement, AAE-ICP still lacks accuracy on some LineMod objects (driller and iron) compared to DPOD. One possible reason is that the indirect methods implicitly use the object's 3D model to build the 2D-3D correspondence. Also, as mentioned in Sec.II-B and II-C, due to the discretization of 3D rotation, AAE and SSD6D both categorize the 3D pose into a finite set of typical possibilities and then apply post-refinement. However, from the LineMod-Occlusion results, AAE and SSD6D achieve comparable results to DPOD based on the *MSPD* evaluation metric. As mentioned above, when objects are occluded, the performance of DPOD starts to decline due to insufficient number of points available for assessment of correspondence.

It is worth noting that the proposed CVML-18 network still has a lower accuracy than PVNet, CosyPose, and CDPN on the LineMod or the BOP version of LineMod-Occlusion. PVNet uses RANSAC-based 2D-3D keypoints voting with a PnP algorithm. The voting hypotheses warrant robust 2D keypoint localization and naturally deal with occlusions. CosyPose directly employs the object's 3D model with iterative refinement in training that increases the pose estimation accuracy. CDPN adopts Scale-Invariant Translation Estimation (SITE) based on the local image patches which improves the pose estimation results, and the extended version CDPNv2 uses a better network architecture together with domain randomization [19] to improve the system robustness to occlusion. Although the results of the above three approaches are better than ours, they require accurate object's 3D model, while the novelty of the proposed CVML-Pose method is to address the 3D pose estimation problem without object's 3D model, depth measurement, and post-refinement.

Comparing the proposed CVML-Pose pipeline to the similar AAE method, our results show that our use of the latent space information is better. Instead of generating a codebook and finding the nearest neighbor in SO(3), we build an MLP to regress a continuous six-dimensional representation for rotation. In this case, an MLP can outperform KNN because it is impractical to have viewpoints sampled densely enough to reduce the pose estimation beyond some threshold level, and therefore the discretization of SO(3) may cause significant errors when assigning the nearest training pose to the object's pose during testing. Another improvement is that the 2D projection of object center is predicted using another MLP, while AAE directly uses the detection bounding box center to be the 2D projection of object center which might be inaccurate when the object is heavily occluded.

What's more, from the results reported in Table 13, the CVML-18 network gets significant improvements with the ground truth bounding box in both LineMod and BOP version of LineMod-Occlusion. These results highlight the importance of an accurate object detector, and further work should look to improve this aspect of the pipeline.

V. CONCLUSION

The paper addresses one of the key challenges in autonomous robotic manipulation of objects, i.e. finding an object's 3D pose in real-time without the object's 3D model. It shows that a state-of-the-art performance can be achieved by using only data from a monoscopic camera without a need for the object's 3D model, depth measurement, or further iterative refinement. The main contribution of the reported research is the proposed use of the VAE and regularized latent space representation, learned from a set of 2D training images. Subsequently, the learned latent space is interpreted with various supervised learning methods for pose estimation, object classification, and shape topology characterization. Different configurations of the method, named CVML-Pose, were systematically evaluated using ablation tests leading to an optimized selection of the method design parameters. In terms of the pose estimation results, the proposed CVML-Pose achieves comparable performance to the state-of-the-art on challenging texture-less occluded data without use of 3D models or depth measurements, which other state-of-the-art methods use extensively, which can be attributed to the construction of robust representation through the use of regularized learning. Due to the implicit learning process, the autoencoder architecture could handle object occlusions, cluttered scenes, and does not need prior knowledge of the object's 3D model or post-refinement, e.g. using ICP. A video demonstration, provided in the paper supplementary material, demonstrates that the method also copes well with low-resolution images captured with an inexpensive webcam. Additionally, the latent space visualization shows potential for the CVML-Pose to be extended to a multi-purpose object characterization, including 3D pose estimation, object classification, and object topology estimation. As the CVML-Pose does not use any iterative post-refinement, the processing

time is predictable, opening the possibility for real-time implementations, which is important for several practical applications including robotics. It is expected that the latent representation could be further leveraged to infer other attributes of the object like material, surface finish, and deformations, which are also important for specific tasks such as robotic grasping.

For future research, there are three main directions envisaged for an expanded version of the CVML-Pose. First, a more comprehensive object characterization including shape, shape deformations, material, pose stability, and surface finish will be investigated. We will also evaluate the sensitivity of the method to various practical limitations of an image acquisition system, including nonlinear lens distortion and motion blur. Second, the results reported here are based on a synthetic training dataset. The results reported in the literature for the state-of-the-art methods suggested that high-quality PBR synthetic training data is a good surrogate for real training images. This conjecture will be extensively tested for the proposed CVML-Pose method. Finally, we have shown that the inaccuracies in object detection can significantly impact performance, and that further work on object detection would be beneficial. To this end, we will investigate the possibility of an end-to-end pose estimation algorithm, which would include object detection, latent space and pose estimation as a single learning process.

ACKNOWLEDGMENT

Data access statement: The study reported in this article has been supported by two existing openly available datasets, namely LM (Linemod) and LM-O (Linemod-Occluded). Both these datasets are available from https://bop.felk.cvut.cz/datasets/.

REFERENCES

[1] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 683–698.

[2] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 574–591.

[3] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot., Sci. Syst. (RSS)*, 2018, pp. 1–10.

[4] T. Hodan, D. Barath, and J. Matas, "EPOS: Estimating 6D pose of objects with symmetries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11703–11712.

[5] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1941–1950.

[6] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4561–4570.

[7] Y. Bukschat and M. Vetter, "EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach," 2020, *arXiv:2011.04307*.

[8] J. Wu, B. Zhou, R. Russell, V. Kee, S. Wagner, M. Hebert, A. Torralba, and D. M. S. Johnson, "Real-time object pose estimation with pose interpreter networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6798–6805.

[9] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7678–7687.

[10] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 292–301.

[11] S. Hoque, M. Y. Arafat, S. Xu, A. Maiti, and Y. Wei, "A comprehensive review on 3D object detection and 6D pose estimation with deep learning," *IEEE Access*, vol. 9, pp. 143746–143770, 2021.

[12] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1677–1734, Mar. 2021.

[13] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, and X. Zabulis, "BOP: Benchmark for 6D object pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.

[14] T. Hodan, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "Bop challenge 2020 on 6d object localization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 577–594.

[15] (2020). *BOP: Benchmark for 6D Object Pose Estimation [Datasets]*. [Online]. Available: https://bop.felk.cvut.cz/datasets/

[16] 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 *Proc. Asian Conf. Comput. Vis.* Springer, 2012, pp. 548–562.

[17] E. Brachmann, "6D object pose estimation using 3D object coordinates [data]," IWR Vis. Learn. Lab, Heidelberg Univ., Heidelberg, Germany, 2020, doi: 10.11588/data/V4MUMX.

[18] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 536–551.

[19] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Augmented autoencoders: Implicit 3D orientation learning for 6D object detection," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 714–729, Mar. 2020.

[20] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7668–7677.

[21] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, Nov. 2008.

[22] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.

[23] M. Sundermeyer, M. Durner, E. Y. Puang, Z.-C. Marton, N. Vaskevicius, K. O. Arras, and R. Triebel, "Multi-path learning for object pose estimation across domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13916–13925.

[24] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first Amazon picking challenge," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 172–188, Jan. 2018.

[25] C. Hernandez et al., "Team delft's robot winner of the Amazon picking challenge 2016," in *Robot World Cup*. Cham, Switzerland: Springer, 2016, pp. 613–624.

[26] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1150–1157.

[27] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2006, pp. 404–417.

[28] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EP*n*P: An accurate O(n) solution to the P*n*P problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009.

[29] M. Lourakis and X. Zabulis, "Model-based pose estimation for rigid objects," in *Proc. Int. Conf. Comput. Vis. Syst.* Cham, Switzerland: Springer, 2013, pp. 83–92.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 998–1005.

[32] V. Lepetit, J. Pilet, and P. Fua, "Point matching as a classification problem for fast and robust object pose estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2004, pp. 1–7.
[33] M. Martinez, A. Collet, and S. S. Srinivasa, "MOPED: A scalable and low latency object recognition and pose estimation system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 2043–2049.
[34] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 858–865.
[35] T. Hodaň, X. Zabulis, M. Lourakis, Š. Obdržálek, and J. Matas, "Detection and fine 3D pose estimation of texture-less objects in RGB-D images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 4421–4428.
[36] D. Li, H. Wang, N. Liu, X. Wang, and J. Xu, "3D object recognition and pose estimation from point cloud using stably observed point pair feature," *IEEE Access*, vol. 8, pp. 44335–44345, 2020.
[37] C.-Y. Tsai and S.-H. Tsai, "Simultaneous 3D object recognition and pose estimation based on RGB-D images," *IEEE Access*, vol. 6, pp. 28859–28869, 2018.
[38] 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," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3828–3836.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[40] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao–Blackwellized particle filter for 6-D object pose tracking," *IEEE Trans. Robot.*, vol. 37, pp. 1328–1342, 2021.
[41] J. Vidal, C.-Y. Lin, X. Lladó, and R. Martí, "A method for 6D pose estimation of free-form rigid objects using point pair features on range data," *Sensors*, vol. 18, no. 8, p. 2678, Aug. 2018.
[42] 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," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 66–70.
[43] P. D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.
[44] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.
[45] P. Ghosh, M. S. Sajjadi, A. Vergari, and M. Black, "From variational to deterministic autoencoders," in *Proc. 8th Int. Conf. Learn. Represent.*, 2020, pp. 1–25.
[46] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5745–5753.
[47] P. Popescu-Pampu, *What is the Genus?* vol. 2162. Cham, Switzerland: Springer, 2016.
[48] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.
[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
[50] T. Hodaň, J. Matas, and Š. Obdržálek, "On evaluation of 6D object pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 606–619.
[51] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 13001–13008.
[52] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3364–3372.
[53] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

JIANYU ZHAO (Graduate Student Member, IEEE) was born in Nanjing, Jiangsu, China. He received the B.Eng. degree (Hons.) in electronic engineering from the University of Central Lancashire (UCLan), Preston, U.K., in 2019, where he is currently pursuing the Ph.D. degree in machine intelligence with the Computer Vision and Machine Learning (CVML) Group. He is also responsible for teaching a postgraduate laboratory course as an Associate Lecturer with the School of Engineering, UCLan. His research interests include object 3D pose estimation and robotic manipulation.

EDWARD SANDERSON received the M.Eng. degree in computer aided engineering and the Ph.D. degree in deep learning-based energy consumption simulation from the University of Central Lancashire, in 2017 and 2022, respectively. He is currently working at the Computer Vision and Machine Learning (CVML) Group, UCLan, as a Postdoctoral Researcher responsible for the development of deep learning-based solutions for assisting clinicians during colonoscopy on an STFC-funded project. His research interests include deep learning and its application to problems, such as the prevention of cancer and the development of sustainable energy supplies and involves data from multiple and mixed modalities. He received third place in the Best Paper Award at the U.K. Conference on Medical Image Understanding and Analysis 2022.

BOGDAN J. MATUSZEWSKI (Member, IEEE) received the Ph.D. degree in electronic engineering from the Wrocław University of Science and Technology, Poland, in 1996. He is currently a Professor in computer vision, the Deputy Director of the Research Centre in Engineering, and the Head of the Computer Vision and Machine Learning (CVML) Group, University of Central Lancashire, Preston, U.K. He has participated in 24 research projects, leading 11 of them, with funding secured from the U.K. Research Councils, EU, and industry. Most recently, he is also the Principal Investigator of the Science and Technology Facilities Council CDN+ funded "Machine Learning System for Decision Support and Computational Automation of Early Cancer Detection and Categorization in Colonoscopy (AIdDeCo)" Project. He has published over 150 research papers, won several awards for his research, and supervised 18 Ph.D.'s to successful completion. His research interests include computer vision, data science, and AI (machine learning) within areas of imaging, healthcare technologies, digital engineering, deformation modeling, segmentation, registration, object characterization, and 3D scene reconstruction and understanding.