Generative Inference for object poses by Regressive-feature Reconstruction

Ryo ISOBE[1,†], Rei KAWAKAMI[2,†], Satoshi IKEHATA[3,†,††], and Ikuro SATO[4,†,†††]

† Institute of Science Tokyo
†† National Institute of Informatics
††† Denso IT Laboratory

E-mail: [1]risobe@d-itlab.c.titech.ac.jp, [2]reikawa@sc.e.titech.ac.jp, [3]sikehata@nii.ac.jp, [4]sato.ikuro@core.d-itlab.co.jp

Abstract Accurate object pose estimation from a single image is an important task for autonomous driving and manufacturing. While most of previous studies use machine-learning based pose regressors, some employ an analysis-by-synthesis approach to iteratively update the pose during inference. Without 3D models, the latter approach can employ novel-view synthesis networks, but the images generated by the synthesizer may lack features essential for accurate pose estimation. In this study, we propose GIRR: Generative Inference for object poses by Regressive-feature Reconstruction that leverages prior knowledge in a pre-trained pose regressor to guide the training of the novel-view synthesis network so that generated images contain features essential for pose estimation. At inference time, object pose is optimized with respect to the sequence of the image synthesizer and the pose regressor. We demonstrate experimentally that our method outperforms existing high-performing regressors on ModelNet10-SO3 and ShapeNet. *Code will be released upon acceptance*.

Key words 3d object pose estimation, analysis-by-synthesis, novel-view synthesis

1. Introduction

Image-based object pose estimation is a long-standing computer vision task, whose goal is to determine the spatial orientation (*i.e.*, tilt, yaw, and pitch) of the object of known categories (*e.g.*, car and chair) relative to some reference frame [1]. This task is critical for various applications such as robotics, augmented reality, and autonomous driving [2], [3].

Modern approaches of object pose estimation from a single image mostly rely on deep networks, which can be divided into two categories: direct approach [1], [4]–[17] and analysis-by-synthesis approach [18]–[20]. The former approach directly predicts object pose from an input image as a regression or classification task, whereas the latter iteratively generates images of the object from candidate view angles until the generated image matches with the input image in some metric.

Formerly, analysis-by-synthesis methods [21], [22] had been advancing direct methods in accuracy; however, today analysis-by-synthesis methods lag behind direct methods in various benchmarks [14], [15]. We hypothesized that this is because effective utilization of priors for pose estimation within the existing analysis-by-synthesis approach has not been fully explored in this approach.

Although some analysis-by-synthesis approaches require object 3D models that limit real-world applications, recent approaches [18] have replaced physics-based rendering with novel-view synthesis networks (a.k.a novel-view synthesizer), enabling image synthesis from a single image input without 3D models. Nevertheless, this method introduces a notable issue. Images generated directly by neural networks, without the assistance of 3D models, often struggle to maintain multi-view geometric consistency and may lack essential characteristics crucial for accurate pose estimation. The absence of such priors potentially hinders the effectiveness of the analysis-by-synthesis approach in pose estimation.

In this paper, we propose an analysis-by-synthesis method, GIRR: Generative Inference for object poses by Regressive-feature Reconstruction, that *leverages the prior knowledge embedded in pre-trained pose regressors*. In GIRR, a pre-trained pose regressor is used to guide the training of novel-view synthesis network in such a way that the generated image and the input image share common features in the regressor's feature space. During inference, we assess not only the photometric consistency (*e.g.*, reconstruction and perceptual losses) between synthesized and actual images but also the alignment of key features for pose estimation, extracted by a pre-trained pose regressor. This approach expects that even if the appearances of generated images do not perfectly align with the input image, the generated image sequence contains rich task-specific features that yield sufficient resolution to identify object pose.

We conducted evaluations of our proposed method using the ModelNet10-SO3 [1] and ShapeNet [23] datasets. The results demonstrate that our pose-aware, analysis-by-synthesis approach outperforms recent promising direct methods for 3D pose estimation [14], [15].

2. Related Work

Image-based object pose estimation has significantly evolved over the past few decades. In early studies, researchers focused on aligning 3D CAD models with image local features to estimate an object's pose [24], [25]. While effective, extracted key points directly affect the accuracy of pose estimation in this approach. With the development of deep learning, image-based object pose estimation has shifted towards leveraging neural networks which are mainly

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.
Copyright ©2024 by IEICE

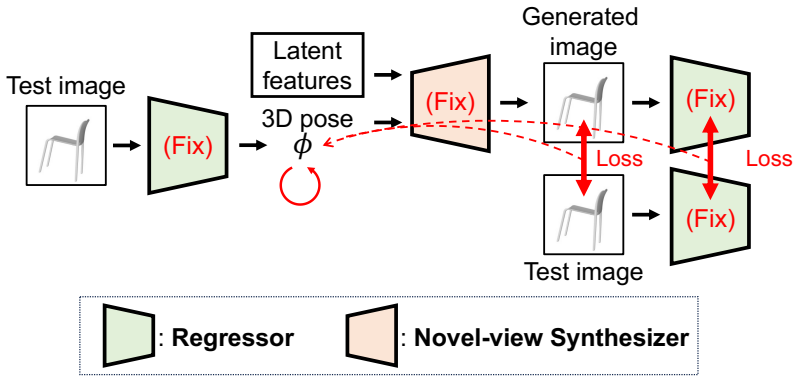


Fig. 1: Conceptual diagram of inference phase in GIRR. GIRR employs an analysis-by-synthesis approach with a loss term that measures discrepancy between regressor features of generated image and test image so that the novel-view synthesizer is encouraged to generate images containing the features essential to identify the pose.

categorized into either a direct approach [1], [4]–[17] or an analysis-by-synthesis approach [18]–[20].

An early direct approach utilizes CNNs to predict object pose from images as a classification task [4], further developed by [26], [27]. To overcome the difficulty in distinguishing small angular differences, Mahendran *et al.* [7] proposes a refiner network to enhance initial pose estimations. Recognizing that single-point estimates struggle with objects having symmetrical orientations, several recent studies have opted to predict distributions over orientations in SO(3) [11]–[17]. For example, Klee *et al.* [14] applies SO(3)-equivariance to predict distributions over 3D rotations from a single image, while Liu *et al.* [15] incorporates rotation normalizing flows for effectively representing arbitrary distributions on SO(3). Directly regressing pose from images is challenging; networks need to acquire feature representations valuable for pose estimation. We believe these representations are also beneficial in the analysis-by-synthesis approach, which will be discussed later.

The analysis-by-synthesis approach has evolved from the traditional template-based method, using 3D model to render images from multiple viewpoints to check consistency with the input image for object pose estimation. However, basic template matching struggles with complex scenarios, such as occlusions.

DeepIM [21] addresses these limitations by refining an object's pose from direct methods, leveraging a neural network to predict the relative pose between the rendered image (from a 3D CAD model) and the input image. Yet, the necessity of a 3D model for every inference is a significant constraint. In response, NOCS [28] introduces a method for reconstructing the 3D object model from RGB images within a canonical coordinate frame, aligning this with input depth measurements. LatentFusion [20] further eliminates the need for 3D models by developing a 3D latent space representation from multi-view images.

While the use of multiple views is effective, our objective is to estimate the 3D pose *from a single RGB image at inference*. The approach by Chen *et al.* [18], closely aligning with this goal, replaces explicit 3D CAD model rendering with a novel-view synthesis network. This network synthesizes images from various viewpoints without using 3D models, aiding pose estimation. However, this strategy's limitation stems from its inability to ensure geometric consistency in the synthesized images, due to the absence of strong multi-view reasoning during image synthesis, and it may lack essential characteristics crucial for accurate pose estimation. The absence of such priors could potentially hinder the effectiveness of the analysis-by-synthesis approach in pose estimation. In this work, we aim to improve pose prediction accuracy from high-performing regressive models with an analysis-by-synthesis framework leveraged by a pre-trained pose regressor to overcome the former limitation.

Algorithm 1 Inference phase of GIRR framework.

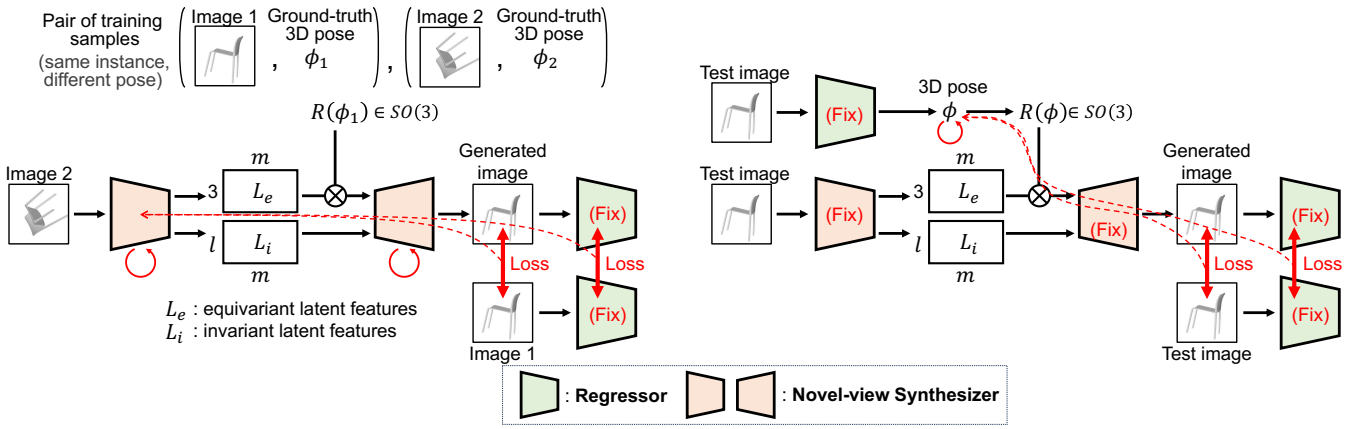
Require: test image I; 3D pose ϕ; novel-view synthesizer g; latent feature L; regressor feature loss $\mathcal{L}_r$; perceptual loss $\mathcal{L}_g$; pixel loss $\mathcal{L}_p$; hyperparameter λ; learning rate μ
Ensure: ϕ^*
  while $\mathcal{L}$ has not converged AND the number of iterations is less than the maximum **do**
    $I' \leftarrow g(\phi, L)$
    $\mathcal{L} \leftarrow \mathcal{L}_p(I', I) + \lambda\mathcal{L}_r(I', I) + (1-\lambda)\mathcal{L}_g(I', I)$
    $\phi \leftarrow \phi - \mu\frac{\partial\mathcal{L}}{\partial\phi}$
  end while
  $\phi^* \leftarrow \phi$

3. Method

We propose a learning-based analysis-by-synthesis approach for category-level 3D object pose estimation. Following the common 3 DoF setup as discussed in [1], [5], [8]–[17], our objective is to estimate 3D object pose $\phi \in \mathbb{R}^3$ (specifically the tilt, yaw, and pitch angles) relative to a category-specific reference frame from a single RGB image containing a single object(1). Building upon the methodology presented in [18], we train a single-image novel-view synthesis network, which is designed to synthesize images from various viewpoints of the object in an input image, without requiring access to instance-specific 3D CAD models. At inference time, the synthesized images are compared to the target image and the errors from this comparison are back-propagated through the network to optimize the object's pose.

While the convenience of CAD-free analysis-by-synthesis framework is presented in [18], its significant drawback is the omission of physical reasoning with 3D models, potentially leading to geometric inconsistencies between the input and synthesized images. These inconsistencies may result in not just imperfect shape but the deficit of critical features essential for accurate pose estimation. To overcome this issue, GIRR optimizes a novel-view synthesis network with an explicit loss term so that the generated images retain details crucial for pose estimation in a geometrically consistent manner. Specifically, we enhance the consistency of features extracted by a pre-trained object pose regressor (*e.g.*, [14], [15]) during training and testing, on top of the conventional photometric consistency between the generated and target images.

(1): Different angle representations can be used without loss of generality.



(a) Training phase (b) Inference phase

Fig. 2: The diagrams of the proposed GIRR framework in detail. In the training phase (a), an encoder and a novel-view synthesizer are jointly trained to minimize the regressive feature reconstruction loss and a conventional image reconstruction loss. In the inference phase (b), 3D pose ϕ is initialized by the regressor, and is updated multiple times until the sum of the loss terms is minimized. All deep modules are kept fixed in this phase.

Figure 2 illustrates the overall flow diagrams for the training and inference phases. Our framework incorporates three key modules: the pose regressor and the novel-view synthesizer. We first provide details of these components, followed by explanation of the training and inference processes.

3.1 Building Blocks

Pose regressor. The pose regressor r takes an image I as input and returns either a point estimate [1], [4], [7]–[10] or a probabilistic distribution [11], [13]–[17] in 3D pose space. Similar to other analysis-by-synthesis methods [21], [28], a pose predicted by the regressor is used as the initial solution in the inference. We also use the same pose regressor to compute the regressive-feature loss, which will be defined later, during training and inference phases. When this loss is ideally minimized, synthesized and target images share the same features in the regressor's feature space.

Owing to this usage, it is desirable for the regressor to extract features from images that are critical for pose estimation. Particularly, the importance of $SO(3)$ rotation-equivariant regressors for pose estimation has been highlighted in recent studies [14], [15], encouraging us to utilize these regressors in our approach. The specific methods of employing the regressor for training and inference phases will be detailed later.

Novel-view synthesizer. The novel-view synthesizer takes a single image of an object and the target view angle as input and returns the image of the object from that view angle.(2) This synthesizer consists of an image encoder and generator, where the encoder extracts instance-specific latent features and the generator generates an image of the same instance from the target view angle.

Specifically, the encoder e_{θ_e} takes an image I as input and returns high-dimensional latent features $L \in \mathbb{R}^{6d}$, specific to the object of interest; *i.e.*, $L = e_{\theta_e}(I)$. We divide these features into two: $L_e \in \mathbb{R}^{3d}$, and $L_i \in \mathbb{R}^{3d}$. Then, we reshape L_e into a $3 \times d$ matrix and apply 3D rotation matrix $R(\phi) \in SO(3)$ with the target angle ϕ, to transform $L \mapsto \{R(\phi)L_e, L_i\}$ before feeding them to the generator. After training with this construction, L_e embeds view-conditioned information necessary for novel-view synthesis, while L_i carries object-specific information that is invariant to the target pose. With the view-point independent part, abstract-level information, such as semantics, can be potentially expressed.

The generator g_{θ_g} takes $R(\phi)L_e$ and L_i as input, and synthesize the image I' from the viewpoint as

$$I' = g_{\theta_g}\left(\left[R(\phi)L_e, L_i\right]^T\right). \tag{1}$$

This equation can be rewritten in a short-hand form as follows,

$$I' = g_{\theta_g}(M(\phi)e_{\theta_e}(I)), \quad M(\phi) = \begin{bmatrix} R(\phi) & 0_{3\times l} \\ 0_{l\times 3} & E_l \end{bmatrix}, \tag{2}$$

where E_l represents the l-dimensional identity matrix, and $0_{a\times b}$ is $a \times b$ zero matrix.

3.2 Training Phase

In the training phase, the encoder e_{θ_e} and generator g_{θ_g} of the category-specific novel-view synthesizer is trained on a set of samples χ. Each sample consists of an image pair and a target pose, (I_j, I_k, ϕ_k), where ϕ_k represents the true pose corresponding to I_k, and I_j is the image of the same instance from a different viewpoint. Then, given I_j and ϕ_k, the novel-view synthesizer is trained to generate I_k by solving the optimization problem as follows,

$$\begin{aligned} \min_{\theta_e,\theta_g} \sum_{(j,k)\in\chi} & \|g_{\theta_g}(M(\phi_k)e_{\theta_e}(I_j)) - I_k\|_2^2 \\ & +\lambda\|f_r(g_{\theta_g}(M(\phi_k)e_{\theta_e}(I_j))) - f_r(I_k)\|_2^2 \\ & +(1-\lambda)\|f_p(g_{\theta_g}(M(\phi_k)e_{\theta_e}(I_j))) - f_p(I_k)\|_2^2. \end{aligned} \tag{3}$$

Here, f_r and f_p is the feature extractors of the regressor and the pre-trained VGG [29] model, respectively. The hyperparameter $\lambda \in [0, 1]$ is introduced to control preference of two regularization terms. The first term is the reconstruction loss between the output of the novel-view synthesizer and the target image in the pixel space. The second term is the regressive-feature reconstruction loss, which aims to make the image synthesizer to generate an image containing the same regressive features as the input image. The last term a.k.a. perceptual loss [30] is added to ensure that generated images have semantically meaningful appearance.

3.3 Inference Phase

In the inference phase, for a given image I of an instance from a specific object category, we predict its corresponding pose by solving the following optimization problem:

(2): It is important to note that the novel-view synthesis and synthesizing images of objects of interest in novel poses system are equivalent within a category-specific reference coordinate system.

$$\phi^\star = \underset{\phi}{\operatorname{argmin}} \|g_{\theta_g}(M(\phi)e_{\theta_e}(I)) - I\|_2^2 + \lambda\|f_r(g_{\theta_g}(M(\phi)e_{\theta_e}(I))) - f_r(I)\|_2^2 + (1-\lambda)\|f_p(g_{\theta_g}(M(\phi)e_{\theta_e}(I))) - f_p(I)\|_2^2. \quad (4)$$

Here, we use the same λ as in the training phase. Note that all the parameters of the modules are kept fixed during inference. The simplified algorithm for the inference phase is shown in Algorithm 1.

As has been mentioned, the regressor is used to initialize the pose ϕ before the test-time optimization to predict $\phi^\star$. In the case of distribution prediction, we pick the highest peak and perform the inference once.

4. Experiments

4.1 Experimental setup

Datasets. We used two datasets, ModelNet10-SO3 [1] and ShapeNet [23]. ModelNet10-SO3 [1] is a dataset of synthetic images created by rendering ModelNet10 [31]. It contains a total of 4,899 object instances across 10 categories. The training data are rendered with 100 different viewpoints for each object instance, while the test data are rendered with 20 viewpoints for each. There are no common object instances between the training and test sets. We used about 10% of the training data as validation data to tune hyperparameters. After hyperparameter tuning, the novel-view synthesizer was trained with all the training data. The models were evaluated using the test data.

ShapeNet [23] is a dataset of 3D models. We used the aeroplane, car, and chair categories, which contain 4,045, 7,497, and 6,778 models, respectively. The image rendering procedure and the proportions of training, validation, and test data are the same as in Mariotti *et al.* [8].

Evaluation metric. We use the median angular error between the predicted and the ground-truth 3D poses as an evaluation metric. This metric is commonly adopted in the field of object pose estimation [11], [13]–[17].

Baselines. Instead of training regressors from scratch, we intend to use existing pre-trained pose regressors to see if our proposed framework can further improve their performance.

On ModelNet10-SO3, we employed the two state-of-the-art regressors as baselines: I2S [14] and Liu *et al.* [15]. Liu *et al.* adopts normalizing flow that transforms a base distribution to target distribution in the pose space. It has two versions based on the distributions, and we chose the model with the Matrix Fisher distribution [11], since it shows a better performance. We also conducted an ablation study using no regressive-feature loss. In this ablation, we set $\lambda = 0$ in Eqs. (3) and (4). This allows us to evaluate the impact of the regressive-feature loss on performance.

On ShapeNet, to the best of our knowledge, the only other work that conducts experiments under the same settings is Mariotti *et al.* [8]. Therefore, we employ the regressor by Mariotti *et al.* within our proposed GIRR framework. The pre-training of the regressor was conducted by us, following the procedure described in [8]. Also, we conducted a comparison with and without the regressive-feature loss.

Architectures. The novel-view synthesizer consists of image encoder e_{θ_e} and generator g_{θ_g}. The encoder e_{θ_e} is composed of five blocks, where each block contains two convolutional layers, and an additional convolutional layer. The kernel size of every convolution is 3 except for the additional convolutional layer (*i.e.*, kernel size is 4) and the stride of each second convolutional layer in a block is 2. The generator g_{θ_g} consists of five blocks, where each block contains upsampling and two transposed convolutional layers, and an additional convolutional layer. The latent feature $L \in \mathbb{R}^{6d}$ mentioned in Sec. 3.1 has dimension of $d = 8,192$.

Table 1: Median angular error (°) on ModelNet10-SO3 [1]. Mean values over 3 different seeds are reported. †: reproduced by us.

(a) Median angular error (°) using I2S [14].

	avg.	avg. (w/o bath.)
I2S †	20.41	4.47
GIRR w/o reg-feature loss	20.25	4.27
GIRR (ours)	**20.16**	**4.14**

(b) Median angular error (°) using Liu *et al.* [15].

	avg.	avg. (w/o bath.)
Liu *et al.* †	12.06	3.37
GIRR w/o reg-feature loss	12.03	3.33
GIRR (ours)	**12.01**	**3.30**

Implementation details. The regressor is kept fixed throughout the training and inference phases, while the novel-view synthesizer is optimized in the training phase and kept fixed in the inference phase. Throughout the training and inference phases on the two datasets, Adam optimizer [32] was used. In the inference phase, the estimate of 3D pose predicted by the regressor is used as an initial value in an analysis-by-synthesis approach according to Eq. (4).

On ModelNet10-SO3, as I2S [14] uses ResNet50 [33] and Liu *et al.* [15] uses ResNet101 as backbones, we used the features from selected layer blocks, namely, conv2_x, conv3_x, conv4_x in [33], for the regressive-feature loss. To enhance synthesis quality, we trained a synthesizer per category, which is a common practice in regressive approaches [9], [12]. In the training phase, the batch size was set to 32. Each model was trained for 100 epochs. The learning rate was scheduled by linear warm-up for first 10 epochs from 5×10^{-6} to 5×10^{-5} and cosine annealing from 5×10^{-5} to 5×10^{-6}. The hyperparameter λ in Eq. (3) was set to 1.0 in GIRR.

In the inference phase, when using I2S [14] as a regressor, the learning rate was set to 1×10^{-4} (5×10^{-5}) for the model trained with $\lambda \neq 0$ ($\lambda = 0$). When using Liu *et al.* [15] as a regressor, the learning rate was set to 3×10^{-5}. The maximum number of iterations was 200. The hyperparameter λ in Eq. (4) at test time was set to 1.0 in GIRR.

On ShapeNet, the regressor by Mariotti *et al.* [8] consists of a block of six convolutional layers and max pooling layers, followed by two additional convolutional layers. We used the features from the first four convolutional layers on the lower side for the regressive-feature loss. In the training phase, the batch size was set to 64. Each model was trained for 100 epochs. The learning rate was scheduled by linear warm-up for 10 epochs from 1×10^{-5} to 1×10^{-4} and cosine annealing from 1×10^{-4} to 1×10^{-5}. The hyperparameter λ in Eq. (3) was set to 0.75 in GIRR.

In the inference phase, the learning rate was set to 1×10^{-3}. The maximum number of iterations was 200. The hyperparameter λ in Eq. (4) at test time was set to 0.75 in GIRR.

The computational cost is approximate, but training the novel-view synthesizer on four Nvidia Tesla P100s concluded within two days for each category of both datasets. Moreover, the inference time for performing pose estimation on a single image is approximately one minute.

4.2 Results

Quantitative results. We present the results of the median angular error on ModelNet10-SO3 [1] in Table 1. The results using I2S [14] and Liu *et al.* [15] as a regressor are shown in Tables 1a and 1b, respectively. The term "avg." represents the average value across all categories. Since the bathtub category exhibited extremely large

Fig. 3: Examples of generated images on ModelNet10-SO3 [1]. Top: test images. Middle: generated images with $\lambda = 0$. Bottom: generated images using I2S [14] with $\lambda = 1$ (ours).

Table 2: Median angular error (°) on ShapeNet [23]. Mean values over 3 seeds are reported. †: reproduced by us.

	avg.	aeroplane	car	chair
Mariotti *et al.* †	5.60	5.94	4.15	6.70
GIRR w/o reg-feature loss	4.27	4.35	3.03	**5.42**
GIRR (ours)	**4.23**	**4.25**	**3.00**	5.45

errors due to its highly symmetric shape, the overall average may not well represent the overall performance. Therefore, we also show the average value for the nine categories excluding the bathtub as "avg. (w/o bath.)". We also present the results of the median angular error on the ShapeNet [23] in Table 2.

These results show that the proposed GIRR framework demonstrates superior performance on average to all regressors, I2S [14] and Liu *et al.* [15] on ModelNet10-SO3 [1], and Mariotti *et al.* [8] on ShapeNet [23]. Though the performance gain from the SOTA method by Liu *et al.* is not very large, we observe clearer improvements from I2S and Mariotti *et al.* The ablation of the regressive-feature loss shown in Tables 1a), 1b) and 2 reveals that utilizing the regressive-feature loss during inference brings performance improvement.

Qualitative results. Examples of generated images in experiments using I2S [14] as a regressor are shown in Fig. 3. The top row shows the test images, the middle row shows the images generated without the regressive-feature loss (*i.e.*, $\lambda = 0$ in Eqs. (3) and (4)), and the bottom row shows the images generated using GIRR ($\lambda = 1$) with I2S [14] regressor. Comparing the middle and bottom rows qualitatively, it can be said that the bottom row (GIRR) better reconstructs fine details, such as furniture legs and object boundaries. Thus, by integrating the regressor's feature-space error into the novel-view synthesizer's loss function, the synthesizer can produce images with more details that help regressor to better estimate poses.

Examples of generated images in the experiments on ShapeNet are shown in Fig. 4. Again, the top, middle, and bottom rows show the test images, the images generated without the regressive-feature loss (*i.e.*, $\lambda = 0$), and the images generated using GIRR with Mariotti *et al.* [8] regressor. Similar to the case of ModelNet10-SO3, fine details are better reconstructed with our GIRR framework.

Analysis on the regressive-feature reconstruction loss. To further analyze our method, we validated our hypothesis that the novel-view synthesizer, when trained with our regressive-feature loss, is capable of synthesizing images containing features crucial for the object pose regressor. Specifically, we input a pair consisting of an image and its corresponding 3D pose into the synthesizer, trained both with and without the regressive-feature loss, to generate an image from the same viewpoint. This image is then fed into the regressor to predict the pose. Subsequently, we calculate the angular error between the predicted and input poses. We present the results of the median angular error evaluated on the ModelNet10-SO3 dataset [1] based on Liu et al. [15] in Table 3. The term "avg." represents the average value across all categories, and "avg. (w/o bath.)" represents the average value across all categories except bathtub.

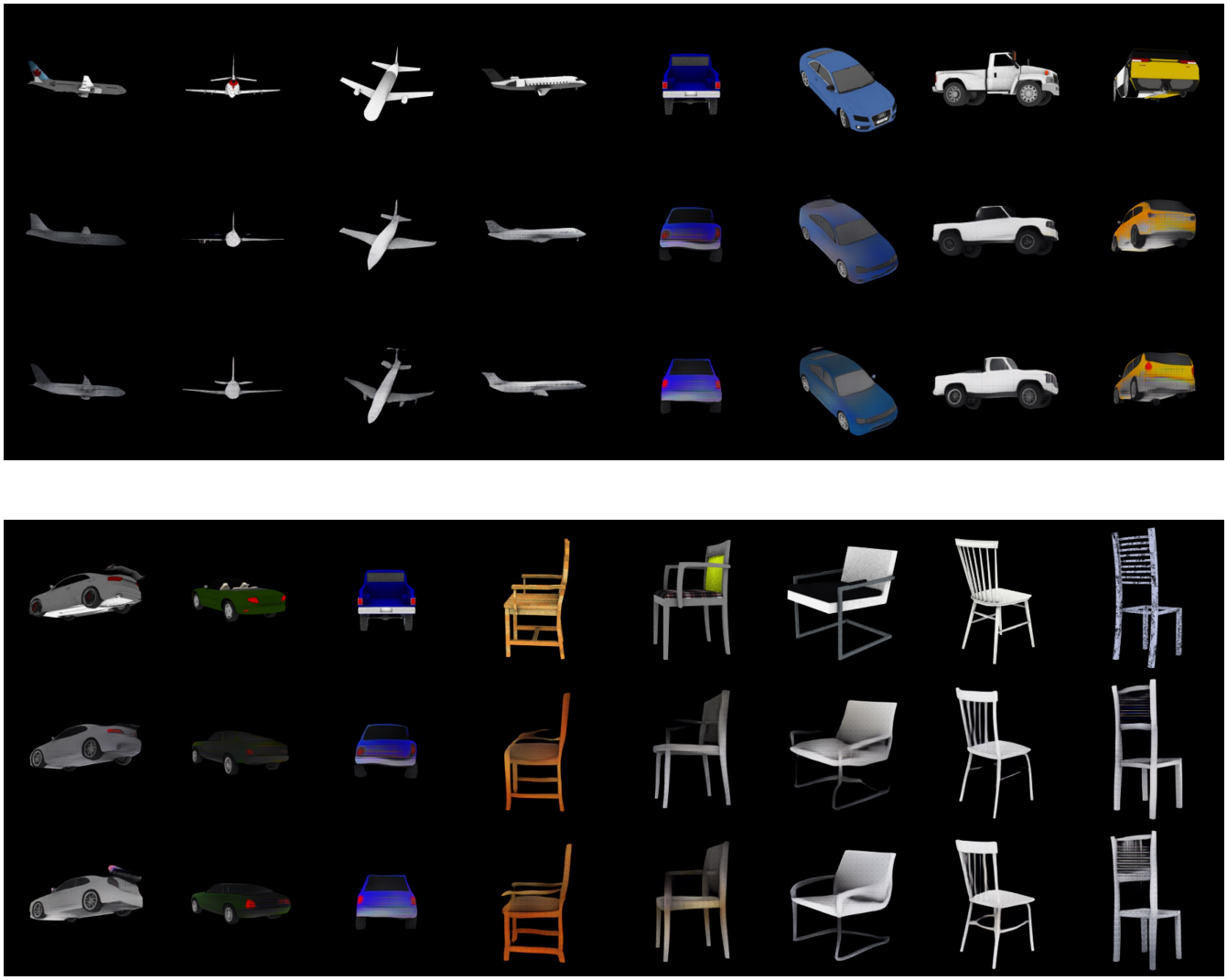


Fig. 4: Examples of generated images on ShapeNet [23]. Top: test images. Middle: generated images with $\lambda = 0$. Bottom: generated images using Mariotti *et al.* [8] with $\lambda = 0.75$ (ours). Each column corresponds to a different test case.

Table 3: Median angular error (°) for generated images on ModelNet10-SO3 [1]. Mean values over 3 different seeds are reported.

	avg.	avg. (w/o bath.)
GIRR w/o reg-feature loss	75.94	68.41
GIRR (ours)	**8.35**	**8.40**

The results of this experiment demonstrate that the use of our regressive-feature loss significantly improves the prediction accuracy when regressors are applied to the generated images. Conversely, this suggests that images generated by a novel-view synthesizer trained without this loss do not contain sufficient information for the regressor to accurately predict the pose, supporting our hypothesis.

5. Conclusion

This work introduces a new analysis-by-synthesis method for category-specific, image-based 3D object pose estimation that does not require 3D CAD models of the objects. To overcome the limited performance caused by geometrically inconsistent novel-view synthesizers, we propose a method that encourages the synthesizer to generate images with features crucial for accurate pose estimation and to infer the poses based on these features. Our method was validated on both the ModelNet10-SO3 and ShapeNet datasets.

The current challenges include the evaluation of our proposed method being limited to synthetic data and the predictions being confined to 3DOF, rather than the more practical 6DOF. Extending our theoretically proven concepts to more realistic problem settings in future work is an important task. Additionally, one inherent challenge associated with the analysis-by-synthesis approach is its significantly slower inference speed when compared to direct methods. This discrepancy in processing speed could potentially restrict its application in real-time scenarios, where immediate responses are crucial. Addressing this issue entails the complex and demanding task of enhancing the efficiency of novel-view synthesis itself. Despite these hurdles, the field is active in developing more efficient synthesis methods and we are optimistic about the issue.

References

[1] S. Liao, E. Gavves, and C.G. Snoek, "Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres," International Conference on Computer Vision and Pattern Recognition, 2019.

[2] Y. Lyu, R. Royen, and A. Munteanu, "Mono6d: Monocular vehicle 6d pose estimation with 3d priors," International Conference on Image Processing, 2022.

[3] J. Mazumder, M. Zand, and M. Greenspan, "Multistream validnet: Improving 6d object pose estimation by automatic multistream vali-

dation," International Conference on Image Processing, 2021.
[4] S. Tulsiani and J. Malik, "Viewpoints and keypoints," International Conference on Computer Vision and Pattern Recognition, 2015.
[5] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," International Conference on Computer Vision and Pattern Recognition, 2019.
[6] R. Brégier, "Deep regression on manifolds: A 3d rotation case study," International Conference on 3D Vision, 2021.
[7] S. Mahendran, H. Ali, and R. Vidal, "A mixed classification-regression framework for 3d pose estimation from 2d images," arXiv preprint arXiv:1805.03225, 2018.
[8] O. Mariotti and H. Bilen, "Semi-supervised viewpoint estimation with geometry-aware conditional generation," European Conference on Computer Vision, 2020.
[9] O. Mariotti, O.M. Aodha, and H. Bilen, "Viewnet: Unsupervised viewpoint estimation from conditional generation," International Conference on Computer Vision, 2021.
[10] O. Mariotti, O.M. Aodha, and H. Bilen, "Viewnerf: Unsupervised viewpoint estimation using category-level neural radiance fields," British Machine Vision Conference, 2022.
[11] D. Mohlin, J. Sullivan, and G. Bianchi, "Probabilistic orientation estimation with matrix fisher distributions," Neural Information Processing Systems, 2020.
[12] S. Prokudin, P. Gehler, and S. Nowozin, "Deep directional statistics: Pose estimation with uncertainty quantification," European Conference on Computer Vision, 2018.
[13] K. Murphy, C. Esteves, V. Jampani, S. Ramalingam, and A. Makadia, "Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold," International Conference on Machine Learning, 2021.
[14] D.M. Klee, O. Biza, R. Platt, and R. Walters, "Image to sphere: Learning equivariant features for efficient pose prediction," International Conference on Learning Representations, 2023.
[15] Y. Liu, H. Liu, Y. Yin, Y. Wang, B. Chen, and H. Wang, "Delving into discrete normalizing flows on so(3) manifold for probabilistic rotation modeling," International Conference on Computer Vision and Pattern Recognition, 2023.
[16] Y. Yin, Y. Wang, H. Wang, and B. Chen, "A laplace-inspired distribution on so(3) for probabilistic rotation estimation," International Conference on Learning Representations, 2023.
[17] O. Howell, D. Klee, O. Biza, L. Zhao, and R. Walters, "Equivariant single view pose prediction via induced and restricted representations," Neural Information Processing Systems, 2023.
[18] X. Chen, Z. Dong, J. Song, A. Geiger, and O. Hilliges, "Category level object pose estimation via neural analysis-by-synthesis," European Conference on Computer Vision, 2020.
[19] R. Araki, K. Mano, T. Hirano, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Iterative coarse-to-fine 6d-pose estimation using back-propagation," International Conference on Intelligent Robots and Systems, 2021.
[20] K. Park, A. Mousavian, Y. Xiang, and D. Fox, "Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation," International Conference on Computer Vision and Pattern Recognition, 2020.
[21] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," European Conference on Computer Vision, 2018.
[22] G. Marullo, L. Tanzi, P. Piazzolla, and E. Vezzetti, "6d object position estimation from 2d images: a literature review," Multimedia Tools and Applications, 2023.
[23] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," arXiv preprint arXiv:1512.03012, 2015.
[24] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3d geometry to deformable part models," International Conference on Computer Vision and Pattern Recognition, 2012.
[25] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," Winter Conference on Applications of Computer Vision, 2014.
[26] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet for joint object categorization and unsupervised pose estimation from multi-view images," IEEE TPAMI, 2021.
[27] S. Joung, S. Kim, H. Kim, M. Kim, I.J. Kim, J. Cho, and K. Sohn, "Cylindrical convolutional networks for joint object detection and viewpoint estimation," International Conference on Computer Vision and Pattern Recognition, 2020.
[28] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L.J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," International Conference on Computer Vision and Pattern Recognition, 2019.
[29] K. Simonyana and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations, 2015.
[30] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," European Conference on Computer Vision, 2016.
[31] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," International Conference on Computer Vision and Pattern Recognition, 2015.
[32] D.P. Kingma and J.L. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," International Conference on Computer Vision and Pattern Recognition, 2016.

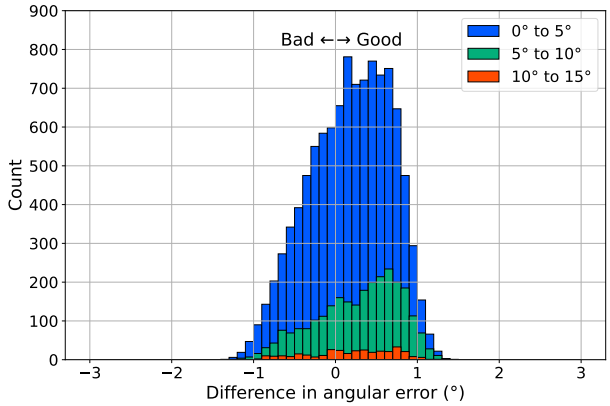
Appendix

1. Ablation analysis

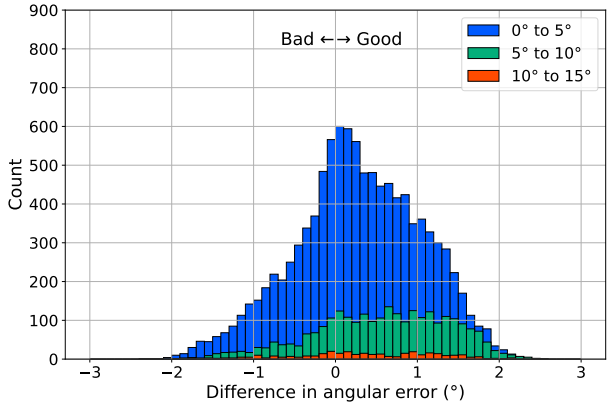
We investigated how much GIRR can improve the initial value of pose predicted by the regressor. The difference between the angular error predicted by the I2S [14] regressor and the angular error predicted by GIRR is shown in Table A·1. Each column corresponds to a range to which the angular error predicted by the regressor belongs. For example, a value with an angular error of 3° predicted by the regressor is included in 0° to 5°. When a number in the table is positive (negative), our GIRR improves (deteriorates) the final prediction accuracy compared to the initialization. Our proposed GIRR framework consistently reduces the angular error in all ranges compared to I2S and naive GIRR without regressive-feature loss. This demonstrates that GIRR can fine-tune the regressor's prediction to improve prediction accuracy to some extent.

A histogram of differences in angular error produced by the regressor I2S [14] and the angular error produced by GIRR is shown in Fig. A·1. Figure A·1 (a) represents the GIRR without regressive-feature loss ($\lambda = 0$ in Eqs. (3) and (4)), and Fig. A·1 (b) represents our proposed GIRR framework ($\lambda = 0.5$ in Eqs. (3) and (4)). Blue, green, and red bars indicate the ranges of angular error produced by the regressor: 0° to 5°, 5° to 10°, and 10° to 15°, respectively. Positive (negative) difference in angular error means that GIRR improves (deteriorates) the final prediction accuracy compared to the initialization. In both Fig. A·1 (a) and (b), it is clearly observed that the distributions lean toward positive side for 0-10°, meaning that overall the initial predictions are effectively corrected. Comparing Fig. A·1 (a) and (b), it can be seen that slightly more population reside in the positive side in (b), that is, the overall angular error has been further reduced.

Acknowledgements This work is an outcome of a research project, Development of Quality Foundation for Machine-Learning Applications, supported by DENSO IT LAB Recognition and Learning Algorithm Collaborative Research Chair (Tokyo Tech.).



(a) GIRR w/o reg-feature loss



(b) GIRR (ours)

Fig. A·1: Histogram of the difference in angular error (°) between the pose with I2S [14] and the pose with GIRR over all categories with one trial. (a) shows GIRR without regressive-feature loss (*i.e.*, $\lambda = 0$ in Eqs. (3) and (4)). (b) shows our GIRR with with regressive-feature loss (*i.e.*, $\lambda = 1.0$ in Eqs. (3) and (4)). Blue, green, and red bars indicate ranges of angular error predicted by the regressor: 0° to 5°, 5° to 10°, and 10° to 15°, respectively. Positive (negative) values in x-axis means that GIRR improves (deteriorates) the final prediction accuracy compared to the accuracy of regressor alone. The larger the difference in angular error is, the smaller the angular error becomes by GIRR.

Table A·1: The difference in angular error (°) between the poses with I2S [14] and the poses with GIRR (top: $\lambda = 0$, bottom: $\lambda = 1.0$) over all categories. Means are taken over three different seeds. Positive (negative) values indicate that GIRR improves (deteriorates) the performance in predicting 3D poses. Our GIRR consistently improves I2S and GIRR without regressive-feature loss performances when I2S absolute error is in 0-5°, 5-10° and 10-15° ranges.

	0° to 5°	5° to 10°	10° to 15°
GIRR w/o reg-feature loss	0.18	0.28	0.21
GIRR (ours)	0.28	0.56	0.40

This work was also supported by JSPS KAKENHI Grant Number JP22H03642.