A Visual Navigation Perspective for Category-Level Object Pose Estimation

Jiaxin Guo[1,2], Fangxun Zhong[2], Rong Xiong[1], Yunhui Liu[2], Yue Wang[1,*], and Yiyi Liao[1,**]

[1] Zhejiang University, Hangzhou, China
[2] The Chinese University of Hong Kong, Hong Kong, China
{jxguo, fxzhong, yhliu}@mae.cuhk.edu.hk,
{rxiong, ywang24, yiyi.liao}@zju.edu.cn

Abstract. This paper studies category-level object pose estimation based on a single monocular image. Recent advances in pose-aware generative models have paved the way for addressing this challenging task using analysis-by-synthesis. The idea is to sequentially update a set of latent variables, e.g., pose, shape, and appearance, of the generative model until the generated image best agrees with the observation. However, convergence and efficiency are two challenges of this inference procedure. In this paper, we take a deeper look at the inference of analysis-by-synthesis from the perspective of visual navigation, and investigate what is a good navigation policy for this specific task. We evaluate three different strategies, including gradient descent, reinforcement learning and imitation learning, via thorough comparisons in terms of convergence, robustness and efficiency. Moreover, we show that a simple hybrid approach leads to an effective and efficient solution. We further compare these strategies to state-of-the-art methods, and demonstrate superior performance on synthetic and real-world datasets leveraging off-the-shelf pose-aware generative models.

Keywords: category-level object pose estimation, analysis-by-synthesis

1 Introduction

Object pose estimation is a fundamental research problem that aims to estimate the 6 DoF pose of an object from a given observation. To enable broad applications in augmented reality and robotics, it is essential that the object pose estimation methods allow for generalizing to unseen objects and being applicable to widely used sensors, e.g., monocular cameras. Thus, there is a growing interest in the challenging task of category-level object pose estimation based on a single monocular image [8].

As a classic idea in computer vision, analysis-by-synthesis has recently shown competitive performance in object pose estimation [8, 25, 46, 54]. This line of

[*] Co-corresponding author. [**] Corresponding author.

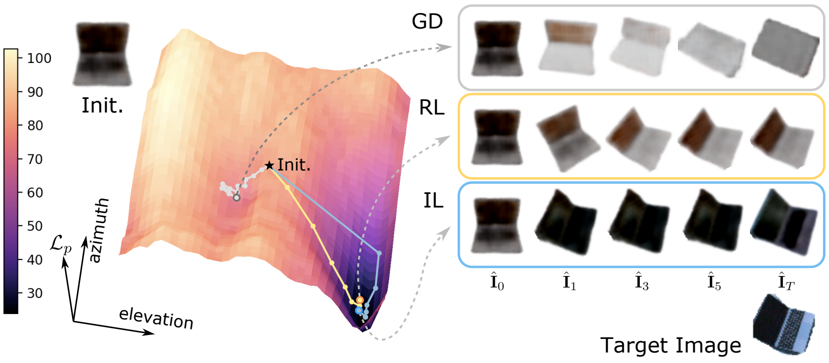


Fig. 1: **Inference of Analysis-by-Synthesis.** We illustrate the perceptual loss between the synthesized image and the target image, calculated over a grid of azimuth and elevation. We further show the navigation trajectory of gradient descent (GD), reinforcement learning (RL) and imitation learning (IL) given the same initialization, including the synthesized images generated at multiple steps. Note that GD converges to a local minimum due to the non-convex loss landscape. RL and IL converge in the correct direction while IL converges faster.

approaches leverages a forward synthesis model that can be controlled by a low-dimensional input, e.g., object pose, and infers the pose via render-and-compare. Given an observation image, multiple images can be synthesized under different object poses, and the one that best matches the observation is selected. While earlier methods only apply to instances of known CAD models taking a graphics renderer as the forward model [25,46], recent works extend this idea to category-level object pose estimation leveraging pose-aware generative models and demonstrate superior performance compared to direct pose regression [8,54].

In this paper, we advocate analysis-by-synthesis but identify a major limitation of existing approaches: it is non-trivial to efficiently retrieve the pose that best reproduces the target observation. Fig. 1 illustrates that existing methods based on gradient descent (GD) are sensitive to initialization and are prone to convergence problems. This is due to the fact that the objective function, i.e., the difference between the synthesized image and the observation, is highly non-convex. Leveraging multiple initial poses is a common remedy for this problem [8,38]. However, this is time-consuming and computationally expensive.

Intending to analyze and improve the *inference* process of analysis-by-synthesis, we view the inference as a visual navigation task, where an agent uses visual input to take a sequence of actions to reach its own goal [58]. This perspective allows us to take inspirations from the visual navigation literature and compare different navigation policies. This formulation leads to our main question: *what is a good navigation policy for category-level object pose estimation?*

To answer this question, we systematically compare different navigation policies. Taking the pose-aware generative model as a *simulator*, we explore common strategies in visual navigation, including reinforcement learning (RL) [33,41,58] and imitation learning (IL) [24,43]. We also study the behavior of GD as a one-step greedy strategy within the same framework. Specifically, we first investigate how design choices, i.e., planning horizon and loss function, affect the behavior of navigation policies. Next, we compare all strategies wrt. convergence, robustness, and efficiency and make the following observations: 1) Both RL and IL remarkably alleviate convergence problems compared to GD as shown in Fig. 1. Despite easily getting stuck in local minima, GD yields a more precise prediction given a good initialization; 2) GD tends to be more robust against disturbance of brightness and shift on the target image; 3) Both RL and IL are more efficient than GD during inference. Compared to RL, IL requires less training time but is less competitive when trained with off-policy data. However, IL achieves similar or even better performance than RL when augmented with on-policy data. Based on these observations, we suggest to combine IL's convergence and efficiency with GD's precision and robustness. We demonstrate that this simple hybrid approach achieves superior performance on category-level pose estimation.

We summarize our contributions as follows: i) We propose to view the inference process of analysis-by-synthesis as a visual navigation task, leveraging the pose-aware generative model as a simulator to provide training data without manual labeling. ii) We conduct thorough comparisons between GD, RL, and IL in terms of convergence, robustness and efficiency. Based on our observations we suggest a simple combination of IL and GD that is effective and efficient. iii) We compare different strategies to state-of-the-art methods on category-level object pose estimation on synthetic and real-world datasets. Our hybrid approach shows competitive performance and consistently improves the inference process of different pose-aware generative models. Our code is released at https://github.com/wrld/visual_navigation_pose_estimation.git.

2 Related Work

Object Pose Estimation: Extensive studies have been conducted for object pose estimation of known *instances* [10,12,19,22,26–28,35,39,40,53]. Only recently, there has been a growing interest in a more general task of *category-level* 6 DoF object pose estimation for unseen instances in a specific category [6,7,44,49–51]. These methods achieve promising results via establishing correspondences across different objects [49–51] or direct regression [6]. In this paper, we are interested in category-level object pose estimation leveraging a pose-aware generative model, eliminating the need for intermediate correspondences compared to [49–51]. In contrast to direct regression methods [6], we model the problem as a long-horizon navigation task to approach the goal sequentially via a set of relative updates. Moreover, all aforementioned methods for category-level object pose estimation are applied to RGB-D images while we focus on a single RGB image-based solution.

Analysis-by-Synthesis for Object Pose Estimation: It is a classical idea to analyze a signal by reproducing it using a synthesis model, which is referred to as analysis-by-synthesis [55]. It has been successfully applied to many tasks, including human pose estimation [31], object recognition [16], and scene understanding [20,34]. A few methods leverage this idea for instance-level object pose estimation [25,46], but are not applicable to unseen instances.

With the rapid progress of pose-aware generative models that allow for generating 2D images under controllable object poses [5,18,29,36,37,45], analysis-by-synthesis approaches are extended to category-level object pose estimation recently [8,38,54]. Among them, a few works demonstrate promising results using a single RGB image [8,54]. As the generative models are differentiable, all these approaches leverage gradient descent to infer the object pose. Due to the non-convex objective function, gradient-based optimization suffers from convergence problems. While iNeRF [54] constrains the initialization range during inference, LatentFusion [38] and Chen et al. [8] start from multiple pose candidates and keep the one that best aligns with the observation. However, it is computationally expensive, and the computing time increases wrt. the number of initial poses. Another idea is to leverage an encoder to provide a better initialization [8,13]. Note that the common underlying idea of these methods is to improve the initialization. In contrast, we focus on analyzing and improving the policy for updating the pose.

Visual Navigation: By sufficiently interacting with the simulation environment, RL has demonstrated superior performance in long-horizon decision tasks in visual navigation [33,41,58]. IL is also commonly adopted when expert demonstrations are available [4,24,43]. In contrast to all aforementioned methods, we propose to adopt a pose-aware generative model as a simulator, where the agent navigates in the input space of a pose-aware generative model for category-level pose estimation. Exploiting RL/IL to learn the gradient is similar to meta gradient descent methods [2]. Compared to existing meta-GD methods, we utilize the simulator to provide explicit supervision towards the global optimum.

Inversion of Generative Models: The idea of analysis-by-synthesis is also closely related to inversion of generative models [1,47,57], see [52] for a detailed survey. While all these works focus on enabling the editing of a real image by searching its corresponding latent code, we leverage pose-aware generative models to estimate category-level object poses. In our task, the inverting process is more sensitive to initialization and prone to convergence problems.

3 Object Pose Estimation as Visual Navigation

Our goal is to improve the inference procedure of the analysis-by-synthesis pipeline for category-level object pose estimation. In the following, we first formulate the problem as a visual navigation task in Section 3.1. Next, we present several navigation policies, including gradient descent (Section 3.2), reinforcement learning (Section 3.3) and imitation learning (Section 3.4).

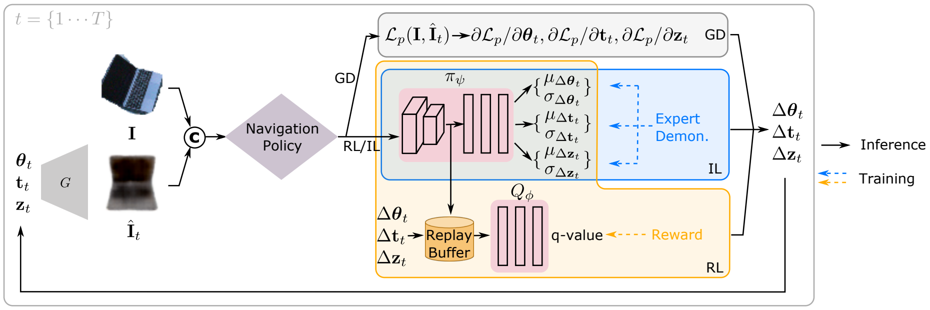

$t = \{1 \cdots T\}$
$\boldsymbol{\theta}_t$
$\mathbf{t}_t$
$\mathbf{z}_t$
G
$\mathbf{I}$
$\hat{\mathbf{I}}_t$
c
Navigation Policy
GD
RL/IL
$\mathcal{L}_p(\mathbf{I}, \hat{\mathbf{I}}_t) \rightarrow \partial \mathcal{L}_p / \partial \boldsymbol{\theta}_t, \partial \mathcal{L}_p / \partial \mathbf{t}_t, \partial \mathcal{L}_p / \partial \mathbf{z}_t$
GD
π_ψ
$\{\mu_{\Delta\boldsymbol{\theta}_t}, \sigma_{\Delta\boldsymbol{\theta}_t}\}$
$\{\mu_{\Delta\mathbf{t}_t}, \sigma_{\Delta\mathbf{t}_t}\}$
$\{\mu_{\Delta\mathbf{z}_t}, \sigma_{\Delta\mathbf{z}_t}\}$
Expert Demon.
IL
Q_ϕ
$\Delta\boldsymbol{\theta}_t$
$\Delta\mathbf{t}_t$
$\Delta\mathbf{z}_t$
Replay Buffer
q-value
Reward
RL
$\Delta\boldsymbol{\theta}_t$
$\Delta\mathbf{t}_t$
$\Delta\mathbf{z}_t$
Inference
Training

Fig. 2: **Category-Level Object Pose Estimation as Visual Navigation.** We illustrate the visual navigation pipeline, where an agent uses visual input (the synthesized image $\hat{\mathbf{I}}$) to reach a target state $\boldsymbol{\theta}^*, \mathbf{t}^*, \mathbf{z}^*$ via iteratively taking T steps of actions. At a time step t, the navigation policy takes as input the synthesized image $\hat{\mathbf{I}}_t = G(\boldsymbol{\theta}_t, \mathbf{t}_t, \mathbf{z}_t)$ and the target image $\mathbf{I}$, to update the state $\boldsymbol{\theta}_t, \mathbf{t}_t, \mathbf{z}_t$ via $\Delta\mathbf{R}_t, \Delta\mathbf{t}_t, \Delta\mathbf{z}_t$. We evaluate and compare three different strategies as the navigation policy, including gradient descent (GD), reinforcement learning (RL) and imitation learning (IL). Note that G is fixed and the GD policy does not contain any trainable parameters. Both RL and IL learn the policy via a network parameterized by ψ, while supervised by different signals.

3.1 Problem Formulation

We aim for 6 DoF category-level object pose estimation from a single image via analysis-by-synthesis. The idea is to sequentially update a set of input variables, e.g., object pose, shape and appearance, of a forward synthesis model, until the generated image best matches the target. The corresponding pose is then selected as the prediction. We view this sequential procedure as a long-horizon visual navigation task as illustrated in Fig. 2. Formally, we model this problem as a Markov decision process (MDP). Let $\mathbf{I}$ denote the target image, and $\hat{\mathbf{I}} = G(\boldsymbol{\theta}, \mathbf{t}, \mathbf{z})$ denote a synthesized image generated by a pose-aware generative model G. Here, G takes as input the object's rotation $\mathbf{R}_{\boldsymbol{\theta}} \in SO(3)$, which is parametrized using Euler angles $\boldsymbol{\theta} = [\boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \boldsymbol{\theta}_z]$, translation $\mathbf{t} \in \mathbb{R}^3$, and a latent code $\mathbf{z}$ for its shape and appearance. Note that G is fixed during the navigation process. The state at a step $t \in [0, T]$ is $\boldsymbol{\theta}_t, \mathbf{t}_t, \mathbf{z}_t$ with a linear state transition:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \Delta\boldsymbol{\theta}_t, \;\; \mathbf{t}_{t+1} = \mathbf{t}_t + \Delta\mathbf{t}_t, \;\; \mathbf{z}_{t+1} = \mathbf{z}_t + \Delta\mathbf{z}_t \tag{1}$$

We further consider the forward synthesis model as the observation function, and the observation $\mathbf{o}_t$ at a step t is the synthesized image $\hat{\mathbf{I}}_t$ combined with the target $\mathbf{I}$. Given an initial state $\boldsymbol{\theta}_0, \mathbf{t}_0, \mathbf{z}_0$, the agent iteratively takes T steps of actions towards reaching the goal state $\boldsymbol{\theta}^*, \mathbf{t}^*, \mathbf{z}^*$ that can best reproduce $\mathbf{I}$. At each step, an action $\mathbf{a}_t := \Delta\boldsymbol{\theta}_t, \Delta\mathbf{t}_t, \Delta\mathbf{z}_t$ is taken following the policy $\pi(\mathbf{a}_t | \mathbf{o}_t)$. This formulation leads to a key question: what is a good navigation policy $\pi(\mathbf{a}_t | \mathbf{o}_t)$? We now briefly discuss three different strategies in this unified pipeline.

3.2 Gradient Descent

When G is differentiable as considered in this paper, it is straightforward to update the input variables using gradient descent (GD) [8,38,54], yielding the following policy:

$$\pi(\mathbf{a}_t|\mathbf{o}_t) = -\lambda\frac{\partial\mathcal{L}_p}{\partial\boldsymbol{\theta}_t}, -\lambda\frac{\partial\mathcal{L}_p}{\partial\mathbf{t}_t}, -\lambda\frac{\partial\mathcal{L}_p}{\partial\mathbf{z}_t} \tag{2}$$

where λ controls the speed of gradient descent. $\mathcal{L}_p$ is the perceptual loss [21] following Chen et al. [8], which measures the discrepancy between the observed image $\mathbf{I}$ and the synthesized image $\hat{\mathbf{I}}$. Here, the policy $\pi(\mathbf{a}_t|\mathbf{o}_t)$ does not contain any trainable parameters, thus can be directly applied without training. On the other hand, the agent may easily get stuck in a local minimum due to the non-convex loss landscape as shown in Fig. 1. Thus, the final performance highly depends on the initial state $\boldsymbol{\theta}_0, \mathbf{t}_0, \mathbf{z}_0$. The main reason is that the agent cannot look into the future to plan for a long-term reward.

3.3 Reinforcement Learning

In contrast to GD, RL allows the agent to explore the environment to maximize an accumulated reward over multiple steps. The RL policy is inspired by [46], where RL is adopted for instance-level pose estimation. While this requires a known CAD model, we apply RL for category-level object pose estimation and recover both the object and its pose simultaneously.

Specifically, we apply Soft Actor-Critic to train the RL agent [15]. As illustrated in Fig. 2, it consists of a policy network $\pi_\psi(\mathbf{a}_t|\mathbf{o}_t)$ to produce a stochastic policy and a Q-value function $Q_\phi(\mathbf{o}_t, \mathbf{a}_t)$ to inform how good the policy is, with ψ and ϕ denoting the parameters of the networks, respectively. The policy network is trained to maximize the expected sum of future discounted rewards $\mathbb{E}[\sum_{t=0}^{T}\gamma^t r_t]$ approximated by $Q_\phi(\mathbf{o}_t, \mathbf{a}_t)$, where T is the number of steps, γ is a discount factor and r_t is a reward at each step:

$$\begin{aligned} r_t = &-\lambda_1\|\mathbf{q}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_t) - \mathbf{q}(\Delta\boldsymbol{\theta}_t)\|_2^2 - \lambda_2\|(\mathbf{t}^* - \mathbf{t}_t) - \Delta\mathbf{t}_t\|_2^2 \\ &- \lambda_3\|(\mathbf{z}^* - \mathbf{z}_t) - \Delta\mathbf{z}_t\|_2^2 \end{aligned} \tag{3}$$

where $\mathbf{q}(\boldsymbol{\theta})$ denotes the quaternion of Euler angles $\boldsymbol{\theta}$, and the weight parameters $\lambda_1 = 10.0, \lambda_2 = 5.0, \lambda_3 = 1.0$ are set to balance the contributions of each term. Here, we assume the target state $\boldsymbol{\theta}^*, \mathbf{t}^*, \mathbf{z}^*$ is available when training the policy network. For example, it can be obtained by randomly sampling a target image from the simulator as $\mathbf{I} = G(\boldsymbol{\theta}^*, \mathbf{t}^*, \mathbf{z}^*)$.

There are two major differences comparing RL to GD. Firstly, the reward in (3) provides cues for global convergence through direct comparison to the best possible action, while the perception loss in (2) does not necessarily lead to an update towards reaching the global optimum. Secondly, the RL policy aims to maximize the predicted accumulated reward in future steps, while the GD policy greedily minimizes a one-step loss. Therefore, the RL policy is expected to be less prone to local minima.

Training Efficiency: The training of the Soft Actor-Critic is based on an experience replay buffer that stores a set of state-action reward pairs. During early training, unconverged policies can lead an agent to random locations, filling the experience replay buffer with low-reward samples and causing inefficient learning. We improve the training efficiency of RL in two ways. Following the relabeling strategy [3], we first relabel the final synthesized image of a failure trial as the target image, turning it into a successful trial. We further manually sample successful trials. Adding both, the relabeled and the manually sampled trajectories, to the experience replay buffer allows for speeding up the training. We refer to the supplementary for ablation study of the training efficiency.

3.4 Imitation Learning

As expert demonstrations are easily accessible from the simulator, an alternative is to directly learn a policy network via imitation learning.

Behavior Cloning: Taking the same network structure in RL, we directly train a policy network $\pi_\psi(\mathbf{a}_t|\mathbf{o}_t)$ via Behavior Cloning (BC) [4], see Fig. 2. In this supervised setting, the policy network is trained with one-step supervision, forcing it to reach the goal in one update. The loss can be formulated as follow:

$$\begin{aligned}\mathcal{L}_{IL} =&\lambda_1\|\mathbf{q}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_t) - \mathbf{q}(\Delta\boldsymbol{\theta}_t)\|_2^2 + \lambda_2\|(\mathbf{t}^* - \mathbf{t}_t) - \Delta\mathbf{t}_t\|_2^2 \\ &+ \lambda_3\|(\mathbf{z}^* - \mathbf{z}) - \Delta\mathbf{z}\|_2^2\end{aligned} \tag{4}$$

where the weight parameters $\lambda_1 = 10.0, \lambda_2 = 5.0, \lambda_3 = 1.0$ are the same as (3). Again, we assume the target state $\boldsymbol{\theta}^*, \mathbf{t}^*, \mathbf{z}^*$ is available during training. Although the agent is supervised by the one-step update, it can be applied iteratively to reach the goal during inference.

DAgger: One disadvantage of BC compared to RL is that the i.i.d. assumption of BC is violated when applied iteratively during inference, as the synthesized image at step $t+1$ depends on previous predictions at step t. Therefore, BC suffers from drift when supervised by off-policy data only. This can be avoided by Dataset Aggregation (DAgger) [42], which extends the training dataset on-the-fly by collecting data under the current policy. Specifically, given a predicted action $\Delta\boldsymbol{\theta}, \Delta\mathbf{t}, \Delta\mathbf{z}$, the synthesized image $G(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}, \mathbf{t} + \Delta\mathbf{t}, \mathbf{z} + \Delta\mathbf{z})$ and its corresponding action label are aggregated to the training set. While differing in the training set, the same loss as BC (4) is adopted. Note that DAgger usually requires an expert to annotate such on-policy data. However, in our case, the pose-aware generative model can provide labeled on-policy data for free.

4 Implementation Details

4.1 Pose-Aware Generative Model

In principle, our navigation framework is compatible with any pose-aware generative model, and we provide two examples in this paper. For both, we use trained networks by the authors and fix them throughout our work.

Voxel-based Generative Model: Chen et al. [8] propose a pose-aware generative model using a voxel-based structure. This model is trained in a supervised manner, taking images of single objects and corresponding poses as supervision. Once trained, it could generate images by controlling the camera/object pose and a latent code $\mathbf{z}$.

GRAF: Generative Radiance Fields (GRAF) [45] is a generative model for radiance fields [32] for high-resolution 3D-aware image synthesis. In contrast to the voxel-based generative model, it learns from *unposed* 2D images using the adversarial loss [14]. GRAF can generate images conditioned on the camera/object pose and two latent codes for object shape and appearance, respectively. We consider these two latent codes jointly as our $\mathbf{z}$. Note that adopting unsupervised generative models such as GRAF means that the pose estimation can be achieved without collecting posed images. Neither the generative model nor the navigation agent requires real-world posed images for training.

4.2 Navigation Policy

Architecture: Following Chen et al. [8], we define the rotation Euler angles $\boldsymbol{\theta}$ by azimuth, elevation and in-plane rotation. The translation $\mathbf{t}$ is represented by horizontal and vertical shifts in the image plane, and the scale factor along the z-axis which aligns with the principle axis of the camera. Following the official implementations, we set the dimension of $\mathbf{z}$ to 16 for the voxel-based generator [8] and 256 for GRAF [45]. More details about the network architecture of our RL and IL policies can be found in the supplementary.

Inference: For GD, we set the number of update steps $T = 50$ and leverage the Adam optimizer [23] using the same learning rate with Chen et al. [8]. For RL and IL, we observe that they converge faster and thus use update steps $T = 10$. We initialize $\boldsymbol{\theta}_0, \mathbf{t}_0$ as the mean pose of all objects within one category. For the latent code, we set $\mathbf{z}_0 = \mathbf{0}$ for all strategies when using the voxel-based generative model, meaning that the agent starts from the mean appearance. We experimentally observe that $\mathbf{z}$ is harder to estimate for GRAF due to its higher dimension. Thereby we initialize $\mathbf{z}$ using an encoder similar to Chen et al. [8]. Note that GRAF relies on volumetric rendering and is memory intensive when calculating gradients on high-resolution images. Inspired by the patch-discriminator used in GRAF, we use image patches as inputs to the GD policy. RL and IL policies are applied to the full image as both do not back-propagate through the GRAF generator and thus are more memory efficient.

5 Experiments

In this section, we first analyze how design choices affect the performance of the navigation strategies in Section 5.1. Next, we systematically compare all three strategies in terms of convergence, robustness and efficiency in Section 5.2.

Finally, we compare these strategies to state-of-the-art approaches for category-level pose estimation in Section 5.3 on both synthetic and real-world datasets.

Dataset: We first evaluate on *REAL275* [50], a standard dataset for benchmarking category-level object pose estimation. We follow the official split of [50] to evaluate on 2760 real-world images, including 6 categories (camera, can, bottle, bowl, laptop and mug). Here, we use the voxel-based generative model provided by Chen et al. [8] as our simulator. Note that this voxel-based generator is trained on synthetic images only. To investigate the performance gap between synthetic and real, we also test on synthetic images in one category (laptop) used for training the generator.

Additionally, we evaluate on *Cars* [11] and *Faces* [30] used in GRAF when using GRAF as the pose-aware generative model. As the poses of the Cars dataset are available, we evaluate on 2000 images used for training GRAF. For the real-world Faces dataset where poses are not available, we sample 2000 images from GRAF as target images for quantitative evaluation and show qualitative evaluation using real-world face images.

We train an RL/IL policy network on each category individually. Since RL and IL are well-known to be sensitive to different random seeds [17], we apply 10 random seeds and report the standard variation for experiments in Section 5.1 and Section 5.2. For REAL275, we train on synthetic images used for training the voxel-based generator as their poses are available. As for Cars and Faces, we randomly generate samples from GRAF for training.

Metrics: We follow the evaluation protocol of NOCS [50] to evaluate average precision (AP) at different error thresholds. For the REAL275 dataset where original images contain multiple objects with background, NOCS considers object detection, classification and pose estimation jointly. Following Chen et al. [8], we use the trained Mask-RCNN network provided by NOCS to detect and segment objects, such that a single-object target image without background is provided to our visual navigation pipeline. This ensures fair comparison to Chen et al. [8] and NOCS as all methods rely on the same network for pre-processing. Following Chen et al. [8], the rotation and translation errors are evaluated as:

$$e_{\mathbf{R}} = arccos\frac{Tr(\mathbf{R}^* \cdot \mathbf{R}_T^{-1}) - 1}{2}, \qquad e_{\mathbf{t}} = \|\mathbf{t}^* - \mathbf{t}_T\|_2 \tag{5}$$

where Tr represents the trace of a matrix, $\mathbf{R}_T$ and $\mathbf{t}_T$ denote the final prediction. Following NOCS [50], the rotation along the axis of symmetry is not penalized for symmetric object categories, i.e., bottle, bowl and can.

5.1 How are Policies Affected by Design Choices?

When adopting a trainable policy such as RL or IL, there are several open questions to design choices: Is it necessary to reach the goal in multi-steps? Is it important to recover the latent code $\mathbf{z}^*$ while we are interested in pose estimation only? We first investigate these questions on the laptop category in REAL275 and its corresponding synthetic images for training the voxel-based generator.

Multi-Step v.s. Single-Step: We investigate whether it is beneficial to perform sequential updates during inference when adopting a trainable policy. Specifically, we compare two variants of IL, behavior cloning ($\mathrm{IL_{BC}}$) and DAgger ($\mathrm{IL_{DA}}$). Both variants are supervised by one-step demonstrations during training, while applied for one step or multiple steps during inference. In this comparison, we refer to multi-step as using $T = 10$ steps. As for RL that is usually applied for making sequential decisions, we observe a degenerated performance when using single-step and report results in the supplementary.

Fig. 3a shows rotation AP on both synthetic and real-world target images. Interestingly, $\mathrm{IL_{BC}}$ and $\mathrm{IL_{DA}}$ perform similar at single-step inference. However, their behaviors diverge when taking multiple steps: $\mathrm{IL_{BC}}$ is degraded while $\mathrm{IL_{DA}}$ is improved. As $\mathrm{IL_{BC}}$ is trained with off-policy data, it diverges when applied iteratively. In contrast, $\mathrm{IL_{DA}}$ overcomes this problem by adding on-policy data. Furthermore, the gap between single-step and multiple-step becomes more prominent when transferred to the real world. It suggests that the multi-step inference helps to overcome the synthetic-to-real gap when leveraging proper training data.

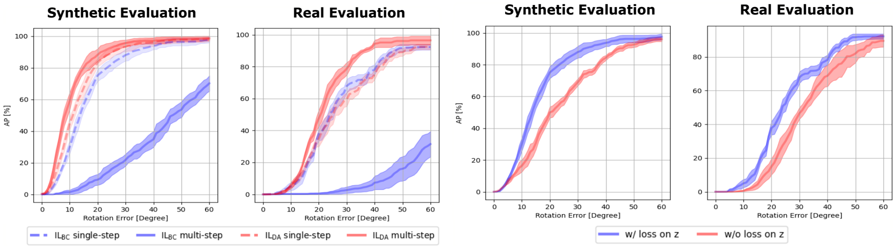
Prediction of Latent Code: Taking the simple single-step $\mathrm{IL_{BC}}$ as an example, we study whether it makes a difference to recover $\mathbf{z}^*$ or not. Specifically, we train another $\mathrm{IL_{BC}}$ policy using the same network architecture but omitting $\|(\mathbf{z}^* - \mathbf{z}) - \Delta\mathbf{z}\|_2^2$ in (4). Fig. 3b compares the rotation AP of $\mathrm{IL_{BC}}$ trained with and without loss on $\Delta\mathbf{z}$. Surprisingly, adding the loss on $\Delta\mathbf{z}$ improves the pose estimation accuracy, especially when the target is real-world images. This finding is interesting as iterative update is not applied here, i.e., $\Delta\mathbf{z}$ does not directly affect the final prediction of $\mathbf{R}, \mathbf{t}$. We hypothesize that recovering $\mathbf{z}^*$ acts as an auxiliary task which can boost the performance of related tasks [56], i.e., the prediction of $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$.

Discussions: We observe that the long-horizon navigation policy can be beneficial despite the trainable policy making reasonable predictions in a single step. However, it is important to take the multi-step inference into account during training, e.g., via on-policy training data in IL. Further, recovering $\mathbf{z}^*$ acts as a multi-task constraint that helps to improve the performance of pose estimation.

5.2 What is a Good Navigation Policy?

We now compare all strategies, GD, RL and $\mathrm{IL_{BC}}$ and $\mathrm{IL_{DA}}$. Based on previous analysis, we adopt single-step inference for $\mathrm{IL_{BC}}$ and multi-step for $\mathrm{IL_{DA}}$.

Convergence: We first evaluate GD, RL, $\mathrm{IL_{BC}}$ and $\mathrm{IL_{DA}}$ in how the initialization affects the pose estimation. To this goal, we manually control the relative pose between the initial state and the target. We evaluate on Cars where the variation of the target poses is the largest. Specifically, we keep the relative translation fixed, and increase the relative azimuth angle from 10° to 180° with an interval of 10°. We compare the rotation precision AP_{10° and AP_{30° in Fig. 4a. As can be seen, GD achieves the best precision in the range $[10^\circ, 40^\circ]$, demonstrating that GD is more precise given a good initialization. This is because



(a) Multi-Step v.s. Single-Step.

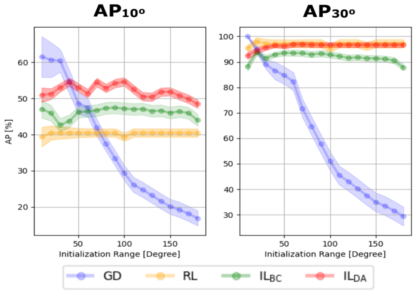
(b) Prediction of Latent Code.

Fig. 3: **Effect of Design Choices.** (a) We compare $\mathrm{IL}_{\mathrm{BC}}$ and $\mathrm{IL}_{\mathrm{DA}}$ on the laptop category on synthetic and real-world images. (b) We compare single-step $\mathrm{IL}_{\mathrm{BC}}$ on the laptop category with and without loss on $\Delta\mathbf{z}$ on both synthetic and real-world target images.

that GD is guaranteed to converge to the global optimum of L_p given a good initialization. However, the precision of GD degrades significantly when the initialized angle deviates further from the target. In contrast, RL, $\mathrm{IL}_{\mathrm{BC}}$ and $\mathrm{IL}_{\mathrm{DA}}$ are not affected, maintaining almost the same performance in different initialization conditions. Further note that $\mathrm{IL}_{\mathrm{DA}}$ trained with on-policy data achieves superior performance compared to RL while $\mathrm{IL}_{\mathrm{BC}}$ is less competitive (at AP_{30°).

Robustness: Inspired by Chen et al. [8], we compare the robustness of GD, RL, $\mathrm{IL}_{\mathrm{BC}}$ and $\mathrm{IL}_{\mathrm{DA}}$ against variations of brightness, occlusion and shift in the target image. This is evaluated on the synthetic images of the laptop category. As shown in Fig. 4c, GD is more robust against disturbance in brightness and shift compared to other strategies. One possible explanation is that the perceptual loss is more robust, as it is calculated based on VGG [48] pretrained on ImageNet [9]. In contrast, both RL and IL policy networks are trained on synthetic images only, thus being less robust against the domain shift. All methods struggle to some extent in terms of occlusions, which can be a disadvantage of the analysis-by-synthesis pipeline. Note that neither RL nor IL policy is trained with data augmentation regarding brightness, occlusion, or shift. We expect better robustness when trained with augmentation against these disturbances.

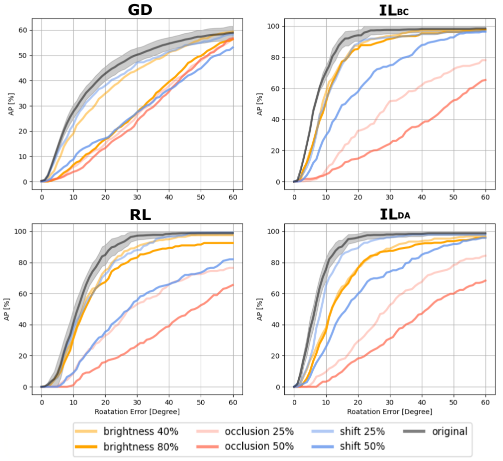
Efficiency: Fig. 4b shows the training and inference time of different strategies on the same device NVIDIA RTX 3090. For GD, we further evaluate the inference time using multiple different initial states. This strategy is used in Chen et al. [8] to avoid converging to local minima. Despite that GD does not require training, its inference takes longer compared to RL, $\mathrm{IL}_{\mathrm{BC}}$ and $\mathrm{IL}_{\mathrm{DA}}$. Furthermore, the time cost of GD increases wrt. the number of initial states. Taking 32 initial states as used in Chen et al. [8] requires almost 7 seconds for one target image. For trainable policies, the training time of both IL variants is less compared to the RL policy thanks to the direct supervision. During inference, $\mathrm{IL}_{\mathrm{BC}}$ is the most efficient method as we take only a single-step update when using $\mathrm{IL}_{\mathrm{BC}}$. RL and $\mathrm{IL}_{\mathrm{DA}}$ take longer, but the overhead is acceptable.



(a) Convergence wrt. initialization.

Strategy	GD	GD_{16}	GD_{32}	RL	IL_{BC}	IL_{DA}
Training (h)	–	–	–	25	2	6
Inference (s)	0.16	2.31	4.68	0.007	0.0008	0.007

(b) Efficiency.



(c) Robustness.

Fig. 4: **Convergence, Robustness and Efficiency** of different navigation policies. (a) Rotation AP_{10° and AP_{30° given different initial states. (b) Total training time and averaged inference time on a single image, both evaluated using the voxel-based generative model at the image resolution of 64 × 64. (c) Rotation AP of navigation policies against disturbance in brightness, occlusion and shift.

Discussions: Our analysis shows that GD achieves the best precision given a good initialization and better robustness against brightness and shift. However, it easily gets stuck in local minima when the initial state is far from the target. Solving this problem by adopting multiple initial states sacrifices efficiency. On the other hand, RL and IL policies are efficient and less prone to local minima. When augmented with on-policy data, IL_{DA} performs similar or even better compared to RL while requiring less training time. Therefore, we suggest to combine GD's precision and robustness with the convergence and efficiency of IL_{DA} by applying a few steps of GD (e.g., $T = 10$) after IL_{DA}. We show results of this simple hybrid method in the next section.

5.3 Comparison to the State-of-the-Art

Baselines: We now compare different strategies to baselines for category-level object pose estimation. We first evaluate a simple baseline following Chen et al. [8], where a VGG16 [48] is adopted to regress the object pose from an RGB image directly. We then compare to state-of-the-art analysis-by-synthesis approaches, including iNeRF [54] and Chen et al. [8]. Both approaches follow the principle of the GD policy but improve from different aspects: iNeRF samples image patches in interested regions to calculate the loss while [8] starts from 32 different initial states. We also consider NOCS [50] as a reference on the REAL275 dataset. Note that NOCS does not apply to Cars and Faces due to the lack of supervision.

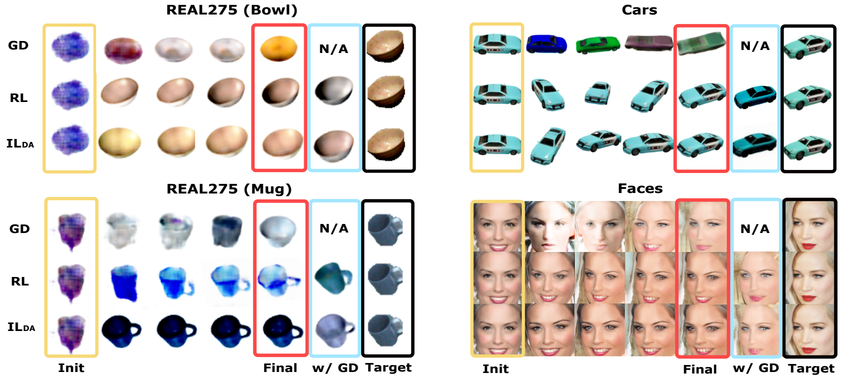


Fig. 5: **Qualitative Comparison** of different strategies, including the initialization, the optimization process, the final image and the target image. For RL and $\mathrm{IL}_{\mathrm{DA}}$, we further show the synthesized image after adding 10 steps of GD.

Moreover, NOCS is trained jointly on synthetic and real-world RGB-D images, while our method is trained on synthetic RGB images only.

Results: The quantitative results are shown in Table 1. Firstly, we observe that VGG is outperformed by other methods, suggesting that it is difficult to regress the pose from a single RGB image directly. For iNeRF, it is interesting that it outperforms GD based on interest region sampling, even when GD is applied to the full image on REAL275. However, it struggles to overcome the convergence problem given the uncurated initialization range as in this paper, particularly on the Cars dataset where the initial azimuth is within the range of $[-180°, 180°]$. Chen et al. [8] significantly outperforms GD by leveraging 32 initial states, but is time-consuming as shown in Fig. 4b.

In contrast, RL and $\mathrm{IL}_{\mathrm{DA}}$ achieve competitive performance compared to [8] but are remarkably more efficient. Moreover, the simple hybrid approaches, RL w/ GD and $\mathrm{IL}_{\mathrm{DA}}$ w/ GD, often lead to better performance. As RL/$\mathrm{IL}_{\mathrm{DA}}$ provides a fairly good start, the subsequent GD converges in only a few steps. This reduces the required steps of "w/ GD" to $T = 10$ in contrast to $T = 50$ in the standard GD, and the inference time of RL, $\mathrm{IL}_{\mathrm{BC}}$, $\mathrm{IL}_{\mathrm{DA}}$ change to 0.033s, 0.027s, 0.033s respectively. The hybrid approach is still much more efficient compared to [8] (0.033s v.s. 4.68s). Note that $\mathrm{IL}_{\mathrm{DA}}$ w/ GD yields the best performance among all RGB based approaches and sometimes even achieves comparable performance to NOCS based on RGB-D input. Interestingly, the hybrid approach sometimes worsens the translation performance, e.g., on the asymmetry categories of REAL275. This might be due to the scale ambiguity of the generative model. Here, NOCS achieves superior performance in translation leveraging depth maps. Lastly, it is worth noting that our learned policies consistently improve the inference of different pose-aware generators. This brings hope to apply our method to more advanced synthesis models for more challenging tasks.

Dataset	Metric	NOCS* [50]	VGG [48]	iNeRF [54]	Chen [8]	**GD**	**RL**	**IL$_{\text{DA}}$**	**RL w/ GD**	**IL$_{\text{DA}}$ w/ GD**
REAL275 Dataset (Symmetry)	AP_{10°	32.8	6.4	21.0	24.0	20.5	18.6	21.6	$\underline{24.8}$	**25.0**
	AP_{30°	66.5	34.8	88.7	92.1	86.2	91.7	$\underline{93.6}$	92.5	**94.2**
	AP_{60°	99.3	76.3	97.1	**99.9**	96.7	98.8	$\underline{99.6}$	$\underline{99.6}$	**99.9**
	AP_{5cm}	93.4	7.8	11.9	12.7	11.9	11.8	12.4	$\underline{13.2}$	**14.6**
	AP_{10cm}	95.0	23.7	26.1	27.4	24.5	23.9	**29.1**	27.2	$\underline{28.8}$
	AP_{15cm}	97.3	38.1	43.8	**46.9**	41.4	39.5	42.3	42.6	$\underline{46.4}$
REAL275 Dataset (Asymmetry)	AP_{10°	20.5	0.6	5.1	**6.9**	5.0	5.1	4.8	6.5	$\underline{6.8}$
	AP_{30°	55.5	12.4	43.1	$\underline{59.5}$	21.1	51.6	53.5	58.7	**60.0**
	AP_{60°	93.3	35.1	62.8	79.2	35.0	74.5	$\underline{80.6}$	76.3	**82.3**
	AP_{5cm}	87.7	10.3	7.7	12.1	9.8	9.7	**17.5**	$\underline{12.8}$	12.5
	AP_{10cm}	98.2	38.1	33.2	42.4	27.7	41.7	**52.2**	42.2	$\underline{46.8}$
	AP_{15cm}	99.5	61.8	48.7	$\underline{73.1}$	50.6	71.6	**75.8**	73.0	72.8
Cars Dataset	AP_{10°	\	5.6	31.7	51.8	21.3	42.4	47.6	$\underline{62.9}$	**65.3**
	AP_{30°	\	15.4	45.4	85.5	33.7	92.8	93.5	$\underline{93.8}$	**94.1**
	AP_{60°	\	32.8	56.6	93.7	37.4	94.2	$\underline{98.2}$	97.7	**98.8**
	AP_{1cm}	\	8.9	12.4	$\underline{35.7}$	9.6	27.2	35.5	29.1	**36.7**
	AP_{3cm}	\	35.2	41.6	$\underline{75.8}$	32.7	71.8	**76.3**	72.5	75.6
	AP_{6cm}	\	52.1	68.3	85.7	60.1	91.8	$\underline{92.0}$	91.4	**92.9**
Faces Dataset	AP_{5°	\	5.3	4.6	24.8	2.0	17.3	**25.8**	15.4	$\underline{23.1}$
	AP_{15°	\	32.8	42.6	88.7	35.9	84.2	$\underline{89.5}$	88.4	**90.8**
	AP_{30°	\	71.1	80.9	92.5	81.2	98.6	98.3	**99.5**	$\underline{99.1}$
	AP_{1cm}	\	11.0	18.2	$\underline{27.4}$	14.3	**27.9**	25.9	26.7	25.3
	AP_{3cm}	\	41.0	59.6	**92.6**	53.8	86.3	90.1	87.8	$\underline{91.5}$
	AP_{6cm}	\	72.9	85.7	$\underline{98.5}$	82.3	97.2	97.7	$\underline{98.5}$	**99.5**
Mean	AP_{rot}	\	27.4	48.3	66.6	39.7	64.2	67.2	$\underline{68.0}$	**70.0**
	AP_{tran}	\	33.4	38.1	52.5	34.9	50.0	**53.9**	51.4	$\underline{53.6}$

*NOCS is based on RGB-D while the others are based on RGB images.

TABLE 1: **Quantitative Comparison** of category-level pose estimation on different datasets.

We further show the qualitative comparison of different navigation strategies in Fig. 5. Note that RL and IL both allow for converging to the correct pose starting from a bad initialization, e.g., the car example. Furthermore, adding 10 steps of GD helps to refine the object pose, see the mug category of REAL275. More qualitative comparisons are provided in the supplementary.

6 Conclusions

In this paper, we formulate the category-level object pose estimation problem as a long-horizon visual navigation task. We experimentally analyze three different navigation policies in terms of convergence, robustness and efficiency. Based on our analysis, we come up with a simple yet effective hybrid approach that enhances the convergence of existing analysis-by-synthesis approaches without sacrificing the efficiency. We further show that it improves the inference of different pose-aware generative models. However, the scale ambiguity of monocular images remains unsolved, thus estimating correct translation is particularly challenging. We plan to tackle these challenges in the future.

Acknowledgement: This work is supported in NSFC under grant U21B2004, and partially supported by Shenzhen Portion of Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone under HZQB-KCZYB-20200089, the HK RGC under T42-409/18-R and 14202918, the Multi-Scale Medical Robotics Centre, InnoHK, and the VC Fund 4930745 of the CUHK T Stone Robotics Institute.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 4
2. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. Advances in neural information processing systems **29** (2016) 4
3. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., Zaremba, W.: Hindsight experience replay. In: Advances in Neural Information Processing Systems (NeurIPS) (2017) 7
4. Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., Zieba, K.: End to end learning for self-driving cars. arXiv.org **1604.07316** (2016) 4, 7
5. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: Pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021) 4
6. Chen, D., Li, J., Wang, Z., Xu, K.: Learning canonical shape space for category-level 6d object pose and size estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 3
7. Chen, W., Jia, X., Chang, H.J., Duan, J., Shen, L., Leonardis, A.: Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021) 3
8. Chen, X., Dong, Z., Song, J., Geiger, A., Hilliges, O.: Category level object pose estimation via neural analysis-by-synthesis. In: Proc. of the European Conf. on Computer Vision (ECCV) (2020) 1, 2, 4, 6, 8, 9, 11, 12, 13, 14
9. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2009) 11
10. Do, T., Pham, T., Cai, M., Reid, I.: Lienet: Real-time monocular object instance 6d pose estimation. In: Proc. of the British Machine Vision Conf. (BMVC) (2018) 3
11. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Proc. Conf. on Robot Learning (CoRL) (2017) 9
12. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2010) 3
13. Duggal, S., Wang, Z., Ma, W.C., Manivasagam, S., Liang, J., Wang, S., Urtasun, R.: Secrets of 3d implicit object shape reconstruction in the wild. arXiv.org **2101.06860** (2021) 4
14. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS) (2014) 8
15. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proc. of the International Conf. on Machine learning (ICML) (2018) 6
16. Hejrati, M., Ramanan, D.: Analysis by synthesis: 3d object recognition by object reconstruction. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2014) 4

17. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D.: Deep reinforcement learning that matters. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018) 9
18. Henzler, P., Mitra, N.J., Ritschel, T.: Escaping plato's cave: 3d shape from adversarial rendering. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 4
19. Hu, Y., Hugonot, J., Fua, P., Salzmann, M.: Segmentation-driven 6d object pose estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 3
20. Isola, P., Liu, C.: Scene collaging: Analysis and synthesis of natural images with semantic layers. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2013) 4
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proc. of the European Conf. on Computer Vision (ECCV) (2016) 6
22. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2017) 3
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) Proc. of the International Conf. on Learning Representations (ICLR) (2015) 8
24. Kretzschmar, H., Spies, M., Sprunk, C., Burgard, W.: Socially compliant mobile robot navigation via inverse reinforcement learning. International Journal of Robotics Research (IJRR) **35**(11), 1289–1307 (2016) 3, 4
25. Krull, A., Brachmann, E., Michel, F., Yang, M.Y., Gumhold, S., Rother, C.: Learning analysis-by-synthesis for 6d pose estimation in RGB-D images. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2015) 1, 2, 4
26. Krull, A., Brachmann, E., Nowozin, S., Michel, F., Shotton, J., Rother, C.: Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017) 3
27. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: Deepim: Deep iterative matching for 6d pose estimation. In: Proc. of the European Conf. on Computer Vision (ECCV) (2018) 3
28. Li, Z., Wang, G., Ji, X.: CDPN: coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 3
29. Liao, Y., Schwarz, K., Mescheder, L.M., Geiger, A.: Towards unsupervised learning of generative models for 3d controllable image synthesis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 4
30. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2015) 9
31. Loper, M.M., Black, M.J.: Opendr: An approximate differentiable renderer. In: Proc. of the European Conf. on Computer Vision (ECCV) (2014) 4
32. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Proc. of the European Conf. on Computer Vision (ECCV) (2020) 8
33. Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., Hadsell, R.: Learning to

navigate in complex environments. In: Proc. of the International Conf. on Learning Representations (ICLR) (2017) 3, 4

34. Moreno, P., Williams, C.K.I., Nash, C., Kohli, P.: Overcoming occlusion with inverse graphics. In: Proc. of the European Conf. on Computer Vision (ECCV) Workshops (2016) 4
35. Muñoz, E., Konishi, Y., Murino, V., Del Bue, A.: Fast 6d pose estimation for texture-less objects from a single rgb image. In: Proc. IEEE International Conf. on Robotics and Automation (ICRA) (2016) 3
36. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: Hologan: Unsupervised learning of 3d representations from natural images. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 4
37. Niemeyer, M., Geiger, A.: GIRAFFE: representing scenes as compositional generative neural feature fields. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021) 4
38. Park, K., Mousavian, A., Xiang, Y., Fox, D.: Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 2, 4, 6
39. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 3
40. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 3
41. Pfeiffer, M., Shukla, S., Turchetta, M., Cadena, C., Krause, A., Siegwart, R., Nieto, J.I.: Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations. IEEE Robotics and Automation Letters (RA-L) **3**(4), 4423–4430 (2018) 3, 4
42. Ross, S., Gordon, G.J., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Conference on Artificial Intelligence and Statistics (AISTATS) (2011) 7
43. Ross, S., Melik-Barkhudarov, N., Shankar, K.S., Wendel, A., Dey, D., Bagnell, J.A., Hebert, M.: Learning monocular reactive UAV control in cluttered natural environments. In: Proc. IEEE International Conf. on Robotics and Automation (ICRA) (2013) 3, 4
44. Sahin, C., Kim, T.: Category-level 6d object pose recovery in depth images. In: Leal-Taixé, L., Roth, S. (eds.) Proc. of the European Conf. on Computer Vision (ECCV) Workshops (2018) 3
45. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. Advances in Neural Information Processing Systems (NeurIPS) (2020) 4, 8
46. Shao, J., Jiang, Y., Wang, G., Li, Z., Ji, X.: PFRL: pose-free reinforcement learning for 6d pose estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 1, 2, 4, 6
47. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 4
48. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proc. of the International Conf. on Learning Representations (ICLR) (2015) 11, 12, 14

49. Tian, M., Ang, M.H., Lee, G.H.: Shape prior deformation for categorical 6d object pose and size estimation. In: Proc. of the European Conf. on Computer Vision (ECCV) (2020) 3
50. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 3, 9, 12, 14
51. Wang, J., Chen, K., Dou, Q.: Category-level 6d object pose estimation via cascaded relation and recurrent reconstruction networks. In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS) (2021) 3
52. Xia, W., Zhang, Y., Yang, Y., Xue, J., Zhou, B., Yang, M.: GAN inversion: A survey. arXiv.org **2101.05278** (2021) 4
53. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In: Proc. Robotics: Science and Systems (RSS) (2018) 3
54. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: inerf: Inverting neural radiance fields for pose estimation. In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS) (2021) 1, 2, 4, 6, 12, 14
55. Yuille, A., Kersten, D.: Vision as bayesian inference: analysis by synthesis? Trends in Cognitive Sciences **10**(7), 301–308 (2006) 4
56. Zamir, A.R., Sax, A., Shen, W.B., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018) 10
57. Zhu, J., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Proc. of the European Conf. on Computer Vision (ECCV) (2016) 4
58. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: Proc. IEEE International Conf. on Robotics and Automation (ICRA) (2017) 2, 3, 4