Abstract

The ability to perceive and understand 3D scenes is crucial for many applications in computer vision and robotics. Inverse graphics is an appealing approach to 3D scene understanding that aims to infer the 3D scene structure from 2D images. In this paper, we introduce probabilistic modeling to the inverse graphics framework to quantify uncertainty and achieve robustness in 6D pose estimation tasks. Specifically, we propose 3D Neural Embedding Likelihood (3DNEL) as a unified probabilistic model over RGB-D images, and develop efficient inference procedures on 3D scene descriptions. 3DNEL effectively combines learned neural embeddings from RGB with depth information to improve robustness in sim-to-real 6D object pose estimation from RGB-D images. Performance on the YCB-Video dataset is on par with state-of-the-art yet is much more robust in challenging regimes. In contrast to discriminative approaches, 3DNEL's probabilistic generative formulation jointly models multiple objects in a scene, quantifies uncertainty in a principled way, and handles object pose tracking under heavy occlusion. Finally, 3DNEL provides a principled framework for incorporating prior knowledge about the scene and objects, which allows natural extension to additional tasks like camera pose tracking from video.

1. Introduction

3D scene understanding is a fundamental problem in computer vision and robotics with numerous applications, including object recognition [50], robotic manipulation[51], and navigation[45]. Inverse graphics is an “analysis-by-synthesis” approach to 3D scene understanding that has found successful applications in a wide variety of tasks [16, 23, 10, 12, 26]. By synthesizing images from possible 3D descriptions of the scene and selecting the 3D scene description that best agrees with the observed image, inverse graphics offers an intuitive and appealing way to reason about the 3D structure of a scene from 2D images. However, challenges such as modeling the gap between rendered images and real-world observations and efficient inference have limited the widespread usage of 3D inverse graphics.

In this paper, we focus on 6D pose estimation, an important task in 3D scene understanding using inverse graphics that aims to infer the rigid SE(3) transformations (position and orientation) of objects in the camera frame given an image observation. We emphasize principled probabilistic modeling as a way to address the central challenges in 3D inverse graphics, and propose 3D Neural Embedding Likelihood (3DNEL). Instead of naively rendering RGB images, 3DNEL uses learned neural embeddings to predict 2D-3D correspondences from RGB and combines this with depth to robustly evaluate the agreement of scene descriptions and real-world observations. This results in a unified probabilistic model over RGB-D images that jointly models multiple objects in a scene. We additionally develop efficient inference procedures using 3DNEL, both with stochastic search for 6D object pose estimation from static RGB-D images, and with particle filtering for object pose tracking from video.

We conduct extensive experiments on the popular YCB-Video (YCB-V) dataset [51]. Our results demonstrate that...
3DNEL’s probabilistic formulation addresses 3D inverse graphics’ central challenges of bridging the gap between rendered images and real-world observations, significantly improving robustness in sim-to-real 6D pose estimation on challenging scenes with principled pose uncertainty quantification, while achieving accuracy on par with state-of-the-art (SOTA) approaches that require extensive tuning. Additionally, 3DNEL’s joint modeling of multiple objects in a scene and natural support for uncertainty quantification enables robust object pose tracking under occlusion. Furthermore, 3DNEL’s probabilistic formulation provides a principled framework for incorporating prior knowledge about the scene and objects, enabling easy extension to additional tasks like camera pose tracking from video, using principled inference in the same probabilistic model without task-specific retraining.

While the field of 6D pose estimation is currently dominated by discriminative approaches based on deep learning, our probabilistic inverse graphics approach provides a complementary alternative that offers unique advantages in terms of robustness, uncertainty quantification and support for multiple tasks due to its probabilistic generative formulation. Our main contributions are three-fold:

- We propose a probabilistic inverse graphics approach to 6D pose estimation that can naturally support uncertainty quantification, track object poses with particle filtering, and incorporate additional knowledge about the scene and objects to handle camera pose tracking without task-specific retraining.
- We conduct extensive experiments on YCB-V and perform on par with SOTA while improving robustness with significantly fewer large-error predictions.
- We show 3DNEL can handle challenging cases such as identifying pose uncertainties for symmetric objects and object pose tracking under heavy occlusion.

2. Related Work

**3D inverse graphics** Our method follows a long line of work in the analysis-by-synthesis paradigm that treats perception as the inverse problem to computer graphics [25, 52, 32, 24, 39, 28, 29]. While conceptually appealing, robustly modeling the gap between the rendered images and real-world observations, especially using appearance information, remains challenging in 3D inverse graphics. Moreover, without uncertainty estimates, even small errors in 3D scene description estimations can be catastrophic for downstream tasks. In recent years, there has been growing interest in leveraging probabilistic formulations in an inverse graphics approach for shape and scene modeling with principled uncertainty quantification [12, 21]. Our work builds on this trend, and additionally integrates appearance modeling through learned dense 2D-3D correspondences [17, 34, 15, 9, 13] with depth information in a unified probabilistic framework to allow superior sim-to-real transfer.

**Discriminative 6D object pose estimation** Discriminative approaches based on deep learning have recently yielded strong performance on 6D object pose estimation. Existing methods either directly regress poses [51, 48, 14, 49], or first establish 2D-3D correspondences [17, 34, 43, 15, 13] followed by a variant of Perspective-n-point (PnP) and random sample consensus (RANSAC) [8]. While such approaches achieve impressive results on a wide variety of datasets, their discriminative nature means there is no natural way to quantify uncertainty, and they cannot be easily extended beyond object pose estimation to additional tasks like object or camera pose tracking from video.

**Neural embeddings for dense correspondence** Many pose estimation methods directly regress 3D object coordinates at each pixel [17, 34, 43, 44] to predict dense 2D-3D correspondences. Recent works [9, 40] show that we can instead learn neural embeddings for 2D pixel and 3D surface locations, and use the embedding similarities to establish dense correspondence. Several recent pose estimation methods [13, 53] demonstrate the benefits of this approach for symmetry handling and category-level generalization. We observe we can combine such embedding similarities with a noise model on the depth information into a unified probabilistic model on RGB-D images. Specifically, we build on SurfEMB [13], and show how the additional probabilistic modeling improves both robustness and accuracy while additionally allowing principled uncertainty quantification and easy extension to additional tasks.

**Render-and-compare for pose refinement** Several recent works [30, 31, 33, 38, 35] adopt a render-and-compare approach for pose refinement, which resembles the idea of “analysis-by-synthesis” in an inverse graphics approach. However, these methods are all discriminative in nature, and train neural networks that take the rendered and real images as inputs, and either directly predict the pose transformations [30, 31, 33, 38] or predict a flow field [35]. In contrast, 3DNEL adopts a probabilistic generative formulation which allows natural support for uncertainty quantification and multiple tasks using principled inference within the same probabilistic model. Moreover, existing render-and-compare methods all consider different objects separately, while 3DNEL jointly models multiple objects in a scene.

**Sim-to-real transfer** Recent advances in photorealistic rendering and physics-based simulations [19, 6] and domain randomization [47] have yielded impressive results [13, 49, 36] in sim-to-real 6D object pose estimation. 3DNEL builds on such advances, and demonstrates that principled probabilistic modeling of the noise distribution between rendered and real-world data can further improve robustness and accuracy in sim-to-real transfer.

**Uncertainty Quantification and Pose Tracking** Several works [5, 41, 37, 46] propose to quantify pose uncer-
tainties, especially for rotations, to achieve robust performance in ambiguous settings such as symmetric objects and heavily occluded scenes. In our work, we demonstrate how 3DNL naturally supports such uncertainty quantification, and additionally show how this helps enable the challenging task of object pose tracking under occlusion.

3. Methods

3.1. Preliminaries

Probabilistic inverse graphics 3D inverse graphics formulates the perception problem as searching for the 3D scene description that can be rendered by a graphics engine to best reconstruct the input image. We propose a likelihood \( P(\text{Observed RGB-D Image} | 3D \text{ scene description}) \) that can assess how well an observed RGB-D image is explained by a 3D scene description. We define a 3D scene description in terms of the number \( N \) of objects in the scene, their classes \( t_1, \cdots, t_N \in \{ 1, \cdots, M \} \), and their corresponding poses \( \mathcal{D} = (\mathbf{P}_1, \cdots, \mathbf{P}_N) \) where \( \mathbf{P}_1, \cdots, \mathbf{P}_N \in \text{SE}(3) \). Each object is associated with a textured mesh, which captures the 3D shape and appearance information of the object. We assume uniform prior distributions over object poses (uniform over a bounded volume for position and uniform on \( \mathbb{SO}(3) \) for orientation). Note that our probabilistic formulation jointly models all objects in the scene, as opposed to many existing probabilistic models where different objects are considered separately.

Noise model on depth information We use the probabilistic model \( P_{\text{depth}}(c | \mathbf{C}; r) = \frac{1}{4\pi r^2} \exp(-\frac{1}{2r}) \) from 3DPM [12] as our noise model on depth information. \( P_{\text{depth}} \) is a uniform distribution in a radius-\( r \) ball centered at a rendered point \( c \in \mathbb{R}^3 \), and models the small spatial displacements in the observed point \( c \in \mathbb{R}^3 \). \( r \) is a hyperparameter that controls the variance of the noise model.

Noise model on RGB information Instead of directly operating on RGB images, we leverage similarity measurements of learned neural embeddings for 2D pixel and 3D surface locations [9, 40, 13, 53] to specify the noise model on RGB information. As a concrete example, we reuse components from SurfEMB [13] to highlight how 3DNL’s principled probabilistic modeling brings added benefits on robustness and uncertainty quantification. But in Section 4.4 we illustrate how 3DNL can leverage other learned neural embeddings and similarity measurements.

For each object class \( t \in \{ 1, \cdots, M \} \), SurfEMB learns two neural embedding models: (1) a query embedding model which maps an RGB image \( \mathbf{I} \) to a set of query embedding maps \( \mathbf{Q}_i \), one for each object class, and (2) a key embedding model \( g_t : \mathbb{R}^3 \mapsto \mathbb{R}^E \) which maps each 3D location \( x \in \mathbb{R}^3 \) (object frame coordinate) on the object surface to a key embedding \( g_t(x) \in \mathbb{R}^E \). Given a pixel with query embedding \( q \in \mathbb{R}^E \), SurfEMB measures the similarity between the query and the key embeddings using a surface distribution \( P_{\text{RGB}}(q_t(x) | q, t) = \exp(q^T g_t(x)) \) that describes which point \( x \) on the object surface the given pixel corresponds to. Importantly, these models can be trained entirely from synthetic data (with photorealistic rendering, physics-based simulations and domain randomization). See Appendix A for a more detailed review.

3.2. 3D Neural Embedding Likelihood (3DNEL)

Processing 3D scene description for 3DNEL evaluation For a given 3D scene description, we use a 3D graphics engine to render it into: (1) A rendered point cloud image \( \hat{C} \), where \( \hat{C}_{i,j} \in \mathbb{R}^3 \) represents the camera frame coordinate at pixel \( (i, j) \). (2) A semantic segmentation map \( \hat{S} \) where \( \hat{S}_{i,j} \in \{ 0, 1, \cdots, M \} \) represents the class to which the pixel \( (i, j) \) belongs. Here 0 represents background. (3) An object coordinate image \( \hat{X} \) where \( \hat{X}_{i,j} \in \mathbb{R}^3 \) represents the object frame coordinate at pixel \( (i, j) \) of the object of class \( \hat{S}_{i,j} \).

Processing RGB-D image for 3DNEL evaluation For an observed RGB image \( \mathbf{I} \) and depth image, we use the learned query embedding models to obtain \( M \) sets of query embedding maps \( \mathbf{Q}_i, t \in \{ 1, \cdots, M \} \), one for each object class, where \( \mathbf{Q}_i \in \mathbb{R}^E \) represents the query embedding at pixel \( (i, j) \), and use camera intrinsics to unproject the depth image into an observed point cloud image \( \mathbf{C} \), where \( C_{i,j} \in \mathbb{R}^3 \) represents the camera frame coordinate at pixel \( (i, j) \).

3DNEL evaluation Figure 1 visualizes 3DNEL evaluation using processed 3D scene descriptions and observed RGB-D images. 3DNEL combines the noise model \( P_{\text{depth}} \) on depth information and the dense 2D-3D correspondence distribution \( P_{\text{RGB}} \), and jointly models multiple objects in a scene through a mixture model formulation. This results in a unified probabilistic model on real RGB-D images.

Intuitively, we assess how well each pixel \( (i, j) \) in the observed point cloud image \( \hat{C} \) is explained by a pixel \( (\hat{i}, \hat{j}) \) in the rendered point cloud \( \hat{C} \), by combining the noise model \( P_{\text{depth}} \) on depth and the noise model \( P_{\text{RGB}} \) on RGB. To jointly model multiple objects in a scene, we assume each pixel \( (i, j) \) in \( \hat{C} \) can be explained by multiple pixels in \( \hat{C} \).

We formalize this with a mixture model formulation, where the mixture component associated with the rendered pixel \( (\hat{i}, \hat{j}) \) combines \( P_{\text{depth}} \) and \( P_{\text{RGB}} \) to assess how well the observed pixel \( (i, j) \) is explained by the rendered pixel \( (\hat{i}, \hat{j}) \). To model background pixels in \( \hat{C} \), we assume the observed point cloud image \( \hat{C} \) resides in a bounded region of volume \( B \), and introduce a uniform distribution \( P_{BG}(c; B) = 1/B \) on the bounded region with mixture probability \( \epsilon \) as an additional mixture component for background modeling. Representing the total number of non-background pixels in the rendered images as \( K = \sum_{i,j} 1[\hat{S}_{i,j} > 0] \), the mixture probability for the mixture component associated with rendered pixel \( (\hat{i}, \hat{j}) \) is given by

\[
\]
Figure 1. **Evaluating 3DNEL** 3DNEL defines the probability of an observed RGB-D image conditioned on a 3D scene description. We first render the 3D scene description into: (1) a depth image, which is transformed to a rendered point cloud image, (2) a semantic segmentation map, and (3) the object coordinate image (each pixel contains the object frame coordinate of the object surface point from which the pixel originates). The object coordinate image is transformed, via the key models, into key embeddings. The observed depth is transformed into an observed point cloud image. The 3DNEL Energy Function (Equation 1) is evaluated using the rendered point cloud image, semantic segmentation, key embeddings, the observed RGB image, and query embeddings.

\[
\begin{align*}
(1 - \epsilon)/K. \quad \text{Since the query embedding at a pixel depends on the entire image I, the mixture components are not properly normalized. This leads to the following energy-based formulation:} & \quad \mathbb{P}_{3\text{DNEL}}(I, C|D) \text{ is proportional to} \\
\prod_{c} \left( \epsilon \mathbb{P}_{BG}(c; B) + \frac{1 - \epsilon}{K} \sum_{\epsilon, s > 0} \mathbb{P}_{\text{depth}}(c|\epsilon; r)^2 \mathbb{P}_{\text{RGB}}(g(B)(\tilde{x}^*_i), q^*_i, \tilde{s}) \right)
\end{align*}
\]

where we denote \( C_{i,j} \) by \( c \), \( \tilde{C}_{i,j} \) by \( \tilde{c} \), \( \tilde{S}_{i,j} \) by \( \tilde{s} \), \( \tilde{X}_{i,j} \) by \( \tilde{x} \), and \( Q^*_i \) by \( q^*_i \). The product is over all observed pixels, and the sum is over all non-background rendered pixels. \( \epsilon, B \) and \( r \) are hyper-parameters that we pick in the experiments. See Appendix B for more details.

### 3.3. Inferring the 3D scene description

**Stochastic search with 3DNEL** Given an observed RGB-D image (represented as \( I \) for the RGB image and \( C \) for the observed point cloud) and a 3D scene description (with object poses \( D = (P_1, \ldots, P_N) \)), 3DNEL evaluates the likelihood \( \mathbb{P}(I, C|D) \) using Equation 1 as described in Section 3.2. We develop an OpenGL-based parallel renderer, and a JAX [4] based likelihood evaluation using the rendered outputs. This allows efficient parallel evaluation of the likelihood of an observed RGB-D image for hundreds of 3D scene descriptions on modern GPUs.

We design a stochastic search procedure with 3DNEL to infer the 3D scene description from an observed RGB-D image. Given the current 3D scene description \( D_t \), the stochastic search procedure is an iterative process where at each iteration, we propose \( K \) candidate poses \( \hat{P}_1, \ldots, \hat{P}_K \) for a randomly picked object \( i \in \{1, \ldots, N\} \). We evaluate in parallel the likelihood of \( K \) 3D scene descriptions obtained by replacing the pose \( P_i \) of object \( i \) in \( D \) with each of the \( K \) candidate poses, and identify the candidate pose with the highest likelihood. We update \( P_i \) to this candidate pose if this increases the likelihood.

We consider 3 types of pose proposals: (1) pose hypotheses proposal proposes a pre-specified set of pose hypotheses (obtained either from the coarse enumerative procedure in Section 3.4 or from a different pose estimation method); (2) ICP proposal uses ICP to align the object to the observed point cloud, and proposes a set of candidate poses sampled from a Gaussian-von Mises–Fisher (Gaussian-VMF) distribution centered around the aligned pose; (3) random walk proposal proposes a set of candidate poses sampled from a Gaussian-VMF distribution centered around the object’s current pose. The Gaussian-VMF distribution means a multivariate Gaussian centered at the current position and a VMF distribution centered at the current orientation.

Given a set of coarse pose estimations as pose hypotheses, the stochastic search procedure with 3DNEL can be used for pose refinement. As we demonstrate in Section 4.1, 3DNEL’s joint modeling of multiple objects in a scene and principled combination of RGB and depth information allows such pose refinement process to further improve robustness and accuracy of previous SOTA.

**Particle filtering for object pose tracking from video** We formulate the problem of object pose tracking from video as probabilistic inference in a state-space model. At each timestep \( t = 1, \ldots, T \), we have the 3D scene description \( D_t = (P_1^{(t)}, \ldots, P_N^{(t)}) \) as the latent state and the RGB-D image \( I_t, C_t \) as the observed variable. We use a
the following state space model \( P \) from the entire RGB-D image into a 3D accumulator space. The predicted distances as radiuses to aggregate information around it, and cast votes towards all points on spheres with between an observed point and possibly present keypoints using farthest point sampling, spherical voting leverages given a set of keypoints sampled from the object surface through the mixture model formulation enables robust estimation on this challenging scene with two similar-looking clamps.

We coarsely discretize the object pose space, and heuristically score the discretized poses using the aggregated information to output top scoring pose hypotheses. Refer to Appendix C for a detailed description of the process.

Our coarse enumerative pose hypotheses generation combines depth information with dense 2D-3D correspondences, and can be implemented efficiently on the GPU (we use Taichi [22]). As we show in Section 4.1, it performs competitively even without separate 2D detections, and can additionally leverage available 2D detections to filter out noisy query embeddings and restrict voting to only the relevant image regions to further improve performance.

### 3DNEL multi-stage inverse graphics pipeline (MSIGP)
We design a MSIGP based on 3DNL for sim-to-real 6D object pose estimation. We generate a set of pose hypotheses for each object class using the above coarse enumerative procedure, and initialize the 3D scene description with the top scoring pose hypothesis for each object class. Starting from the initial 3D scene description, we use the stochastic search procedure as described in Section 3.3 to infer the 3D scene description

\[
\tilde{P}_1, \ldots, \tilde{P}_N = \arg \max_{P_1, \ldots, P_N} \mathbb{P}(I, C | P_1, \ldots, P_N)
\]

We start with the pose hypotheses proposal, followed by the ICP proposal, before finally applying the random walk proposal. For each type of proposal, we go through all the objects once. See Figure 2 for an illustration of the 3DNL MSIGP. We follow [14] and use [27] to fill in missing depth. See Appendix D for details.

**Training** To highlight that performance improvements are coming entirely from 3DNL’s probabilistic formulation, we use publicly released pretrained SurfEMB models,
and also use SurfEMB as our primary baseline. These include query models, key models, mask predictors for different object classes, and an additional 2D detector from CosyPose [30], all trained entirely from synthetic data. 

Comparison with SurfEMB SurfEMB proposes a pose estimation pipeline based on probabilistic modeling of dense 2D-3D correspondences: it samples 2D-3D correspondences on image crops from object detection, generates pose hypotheses using PhP, scores the pose hypotheses using its training objective, and does gradient-based refinement. It does all of the above separately for individual objects using only RGB information. Depth is only used in median depth comparison for heuristic pose refinement.

In contrast, 3DNL’s probabilistic inverse graphics approach focuses on explaining the entire scene, and the proposed 3DNL MSGP differs from SurfEMB’s pose estimation pipeline in a few key ways: (1) 3DNL MSGP jointly models all objects which improves robustness. (2) 3DNL MSGP generates pose hypotheses using a coarse enumerative procedure, which we demonstrate to be faster and better in Section 4. (3) 3DNL MSGP combines correspondence and depth in a principled way in all aspects of the modeling (likelihood formulation, hypotheses generation and refinement) and improves performance.

4. Experiments

In this section, we aim to answer the following questions: (1) Can 3DNL achieve improved robustness in challenging sim-to-real setups compared to more discriminative baselines? (2) Does 3DNL perform on par with SOTA? (3) Can 3DNL be additionally used to quantify uncertainty, as well as for object and camera pose tracking?

4.1. Sim-to-real object pose estimation

Evaluation We follow the evaluation protocol of the Benchmark for 6D Object Pose Estimation (BOP) challenge [19]. The task is to estimate the 6D poses of objects in a scene from a single RGB-D image, assuming knowledge of the number of instances of each object class in the scene. For a predicted pose, we calculate three error metrics: Visible Surface Discrepancy (VSD) [18, 19], Maximum Symmetry-Aware Surface Distance (MSSD) [7], and Maximum Symmetry-Aware Projection Distance [3]. Average recalls AR_VSD, AR_MSSD, AR_MSPD are computed for each error metric across a range of error thresholds. The aggregate Average Recall (as reported in Table 1) is the average of AR_VSD, AR_MSSD, and AR_MSPD.

Baselines We use SurfEMB as our main baseline for a series of detailed analysis. We additionally include the sim-to-real performance of several recent SOTA 6D object pose estimation method [30, 35, 14, 31, 36] as context. For [30, 35], we use the publicly available codebase to retrain on only synthetic data, and re-evaluate their performance.

Robustness Figure 3 illustrates how 3DNL’s probabilistic formulation significantly reduces large-error pose estimates when compared with SurfEMB and improves robustness. Across all YCB-V test images, there are 4123 object instances. Each point on the scatter plots corresponds to an object instance, and the point’s x and y coordinates are the pose prediction error of 3DNL MSGP and SurfEMB, respectively. Points above the dashed line correspond to object instances for which 3DNL MSGP had a lower prediction error than SurfEMB. Figure 3(a) shows the scatter plot for all 4123 predictions and Figure 3(c) shows the scatter plots for 6 representative object classes. Figure 3(b) shows, across a range of error thresholds, the number of pose predictions with error above that threshold.

Qualitatively, we observe 3DNL’s probabilistic formulation is especially helpful in challenging situations: (1) Scenes (e.g. Figure 2) with similar-looking objects. SurfEMB makes per-object predictions and incorrectly predicts both clamps to be in the back, while 3DNL jointly models all objects in the scene and makes correct prediction. (2) Scenes (e.g. Figure 3(d)) with objects like the red bow where RGB alone is not informative enough. With a principled combination of RGB and depth information, 3DNL MSGP can reliably correct a large number of related errors. (3) Scenes (e.g. Figure 3(e)) with missing 2D detections. SurfEMB cannot recover from missing 2D detections, while 3DNL MSGP can robustly aggregate information from the entire image and reliably avoid such errors.

---

1 Code to reproduce the results in this section is available at https://github.com/deepmind/threednel.
Identifying pose uncertainties Pose uncertainty may arise from partial observability, viewpoint ambiguity, and inherent symmetries of the object. 3DNEL can naturally quantify such pose uncertainties due to its probabilistic formulation. Figure 4(a) illustrates how 3DNEL identifies the pose uncertainty of the red bowl due to its inherent symmetry. Figures 4(b)(c) consider the red mug in YCB objects. 3DNEL can accurately capture that while there is no pose uncertainty when the mug handle is visible, there are a range of equally likely poses when the mug handle is not visible.

Comparison with baselines In Table 1, we report the Average Recall for 3DNEL MSIGP and representative recent baselines. 3DNEL MSIGP significantly outperforms SurfEMB despite using the same underlying models, highlighting the benefits of 3DNEEL’s principled probabilistic modeling. In addition, 3DNEL MSIGP achieves results that are on par with SOTA, and outperforms all the included baselines in the sim-to-real setup by a large margin, except concurrent work GDRNPP [36] which requires extensive tuning.

Abalations We report additional ablations in Table 1.

We evaluate No RGB in Likelihood, where we drop $P_{RGB}$ in Equation 1. and No Depth in Likelihood where we replace $1(||C_{i,j} - \tilde{C}_{i,j}|| \leq r)$ in Equation 1 with a fixed $3 \times 3$ 2D patch. Both are substantially worse.

We evaluate SurfEMB initialization + stochastic search, which replaces the coarse enumerative pose hypotheses generation with pose hypotheses from SurfEMB, and initializes with SurfEMB predictions. This leads to 2.73% improvement over SurfEMB but is still worse than 3DNEL MSIGP, illustrating the value of our stochastic search and coarse enumerative pose hypotheses generation.

We evaluate No 2D Detection, where we do not use 2D detections for pose hypotheses generation. This uses much less information than all the other baselines, but outperforms strong baselines like CosyPose in the sim-to-real setup, demonstrating the ability of our coarse enumerative pose hypotheses generation to robustly aggregate information from entire noisy query embedding images.

We evaluate the effects of the three different proposals used in our stochastic search, by separately removing the pose hypotheses proposal, ICP proposal and random walk proposal (No pose hypotheses/ICP/random walk proposal). Performances remain strong, but drop to various extent, illustrating the contributions from all three proposals.

Inference speed Inverse graphics approaches are traditionally computationally expensive. To speed up inference, we downsample input images by a factor of 0.25. We fully leverage recent hardware advances to develop efficient GPU
3DNEL naturally quantifies pose uncertainty in the scene. 3DNEL identifies pose uncertainty for the red bowl due to its inherent symmetry, and accurately captures the range of equally likely poses for the red mug when its handle is not visible.

implementations. We use a single NVIDIA A100 GPU for our experiments. When tested in the same setup, SurfEMB reported taking 1.2s for pose hypotheses generation using PnP+RANSAC, while our coarse enumerative pose hypotheses generation takes 0.2s with better results. Our stochastic search with 3DNEL runs at a similar speed to SurfEMB’s pose refinement, and averages around 1s per object. Most of our implementation (except parallel rendering with OpenGL) is written in Python, using a mix of JAX, PyTorch and Taichi. We empirically observe that intercommunication between different packages creates additional overhead. We expect a compiled implementation using a single framework to further speed up inference.

Inference hyperparameters For the above experiments, we pick inference hyperparameters\(^2\) by visually inspecting inference results on a small number of real training images outside the test set. To verify 3DNEL can robustly apply to a variety of object types and scene configurations, we evaluate on two additional datasets using the same hyperparameters. On TUD-L \([20]\), 3DNEL improves SurfEMB by 2.7\% (88.1\% vs 85.4\%). On LM-O \([2]\), due to the small objects in the scenes, we increase the downsampling factor from 0.25 to 0.6. Keeping all other hyperparameters the same, 3DNEL improves SurfEMB by 0.7\% (76.7\% vs 76.0\%).

4.2. Object pose tracking under occlusion

We apply 3DNEL under the particle filtering framework as described in Section 3.3 for object pose tracking under occlusion. Figure 5(a) visualizes tracking with 3DNEL with 200 particles on a representative YCB-V video, where the tomato can gets fully occluded before reappearing. Existing per-object likelihood \([5]\) cannot handle such cases without ad hoc occlusion modeling. In contrast, 3DNEL’s joint modeling of multiple objects in a scene naturally handles occlusion through rendering and can reliably track through occlusion.

In Figure 5(a), the tomato can is briefly occluded by a narrow occluder, and the estimated posterior from particle filtering indicates there is little uncertainty about where the tomato can is even when it is fully occluded. However, accurate uncertainty quantification is important for tracking objects through extended occlusion. To illustrate this point, we generate a synthetic video in which a sugar box moves from left to right and becomes fully occluded by a cracker box. Figure 5(b) visualizes tracking with 3DNEL with 400 particles in this challenging video. We observe that 3DNEL can accurately quantify uncertainty with particle filtering: the estimated posterior concentrates on the actual pose when the sugar box is visible, yet spreads to cover a range of possible poses when the sugar box becomes occluded. Such modeling of the full posterior helps 3DNEL to regain track when the sugar box reappears, after which the posterior again concentrates on the actual pose. We observe that if we instead use a smaller number of particles (e.g. 50), we would not be able to accurately represent the uncertainty introduced by the occlusion and would lose track.

4.3. Extension to camera pose tracking from video

We demonstrate that 3DNEL’s probabilistic formulation provides a principled framework for incorporating prior knowledge about the scene and objects, and enables easy extension to camera pose tracking from video using probabilistic inference in the same model without task specific retraining. We extend our single-frame 3DNEL MSIGP to the multi-timestep setup by introducing a dynamics prior that samples the object pose at time \(t + 1\) from a Gaussian-VMF
3DNEL's probabilistic formulation enables robust object pose tracking under occlusion with particle filtering. Green dots visualize the particles used to estimate posterior distributions in particle filtering. (a) 3DNEL can track objects through heavy occlusions. (b) 3DNEL can accurately quantify uncertainty (shown by the spread-out particles), which enables tracking through extended occlusions.

Table 2. Extending 3DNEL to camera pose tracking improves performance compared to single-frame setups. We apply 3DNEL to camera pose tracking from video. We demonstrate that 3DNEL's probabilistic formulation allows us to incorporate additional knowledge that the scene is static and leverage temporal information to significantly improve pose estimation accuracy over the single frame setting.

distribution restricted to poses with position at most 3cm away from the object pose at time \( t \). We initialize object poses at the first frame to ground truth annotations to avoid introducing systematic errors. We again apply stochastic search with 3DNEL, using just ICP and random walk proposals. However, we further assume we know the scene is static and only the camera moves, which translates into jointly updating all object poses by the same amount in a scene. Table 2 shows that the same inference procedure can readily handle such extensions, taking into account the dynamics prior and the knowledge of a static scene within the same probabilistic model. We observe comprehensive improvements over single frame predictions.

4.4. Alternative similarity measurements

So far our experiments reuse components from SurfEMB to highlight the added benefits of robustness and uncertainty quantification from 3DNEL's principled probabilistic modeling. However, 3DNEL can leverage any learned neural embeddings and similarity measurements. To illustrate this, we demonstrate applying 3DNEL to a simple few-shot pose estimation task in simulation using similarity measurements based on learned neural embeddings from DINoV2 [42], a recently released vision foundation model.

Given 10 training images of a novel object from different perspectives, we reconstruct a 3D mesh of the object using bundle adjustment. For each training image, we sample 2000 pixels on the object surface, and record their corresponding object frame coordinates and DINoV2 embeddings. For a given 3D scene description, we obtain the key embedding of a rendered pixel by finding the closest sampled 3D surface point from the training images and taking the corresponding DINoV2 embedding. We use a VMF distribution with concentration parameter 100.0 on normalized DINoV2 embeddings as our similarity measurement \( P_{\text{RGB}} \).

We evaluate using the sugar box from YCB-V. We generate a test set consisting of 100 RGB-D images of the sugar box at a fixed position directly in front of the camera, but in randomly sampled orientations. We find that 3DNEL+DINOv2 can accurately infer the object’s orientation with an average rotation error of 0.839 degrees. These results suggest a promising future direction to combine the rich, general neural representations from powerful pretrained vision foundation models with the robustness and interpretability of structured probabilistic models.

5. Conclusion

In conclusion, we propose a probabilistic inverse graphics approach to pose estimation. We leverage learned neural embeddings and depth information to model likelihood of observed RGB-D images given 3D scene descriptions, and build efficient inference procedures for both pose estimation and tracking. Our approach achieves performance on par with SOTA in sim-to-real setups on YCB-V, and can more robustly handle challenging scenes. Finally, thanks to our probabilistic formulation, we can jointly model all object poses in the scene and easily extend to additional tasks such as uncertainty quantification and camera tracking.
References


