NeRF-Frenemy: Co-Opting Adversarial Learning for Autonomy-Directed Co-Design

Stanley Lewis, Bahaa Aldeeb, Anthony Opipari, Elizabeth A. Olson, Cameron Kisailus, Odest Chadwicke Jenkins[1]
Robotics Department
University of Michigan
Ann Arbor, Michigan 48109
Email: {stanlew, baldeeb, topipari, lizolson, kisailus, ocj}@umich.edu

***Abstract*—As the presence of robotics in industries such as warehousing and manufacturing grows, the need arises to optimize product design for downstream autonomous tasks. For example, when considering object segmentation or pose regression, minimizing featureless or symmetric regions of an object can improve the quality of these estimations. Adding visual fiducial markers can provide landmarks for these tasks, however they can become warped on deformable packaging or distract from designed branding of an object. To address this gap, our proposed framework, *NeRF-Frenemy*, incorporates techniques introduced by the adversarial machine learning community, but in a cooperative manner to improve the fidelity of manipulation-focused perception tasks. NeRF-Frenemy optimizes a neural radiance field (NeRF) representation of an object against a given pre-trained perception model by seeking a minimal perturbation to the implicit space. The resulting changes in the objects' appearance from these alterations to the implicit space can be realized to a modified object appearance which will improve the given model's performance on the object. In this work, we show an initial result of this approach on a member of the YCB Dataset against the image segmentation portion of the PoseCNN model. The project webpage is available at: https://progress.eecs.umich.edu/projects/nerf-frenemy.**

I. Introduction

As the commercialization of robotics in industries such as grocery stores (Amazon Go), robotic warehouses (Amazon/Kiva, Boston Dynamics, Fetch Robotics), and autonomous truck driving (Tusimple, Waymo, Gatik) become more common, manufacturers will need to ensure their product designs are compatible with both human and robotic users. One solution is for commercial products to include fiducial markers on their packaging, similar to the current utilization of UPC barcodes in commercial environments. However, this approach would interfere with branding: UPC's can be placed in discreetly while fiducial markers must be placed in visually prominent locations around the product.

Our work reexamines the promise of slight modifications to the product's appearance to assist in these tasks, but by creating these visual changes with the downstream network in mind. This work is also inspired by modern computer-aided product design workflows that utilize a topology optimization stage to reduce material costs while ensuring physical performance constraints are met. Analogously, NeRF-Frenemy seeks to improve perception accuracy while minimizing material or color modifications.

Deeply learned methods such as PoseCNN [17] have demonstrated state-of-the-art performance on tasks such as feature detection and pose estimation. In attempting to better understand why deep learning models are performing well, the concept of adversarial attacks was presented [14]. By having access to a trained differentiable model one can use gradient methods to alter model inputs rather than the model's parameters to dramatically decrease model performance. In contrast, the present study does not aim to fool a network as adversarial approaches do - instead, we utilize counter adversarial approaches to make an object's design more compatible with task-specific differentiable estimators. To allow for the redesign of an object, we use a NeRF as a differentiable implicit representation. Using a NeRF allows us to generate representative RGB-D renderings in a scalable and differentiable manner. This representation facilitates direct optimizations of the object's geometry and color information conditioned on a task-specific differentiable model.

II. Related Work

A. Object Pose Estimation

Estimating the 6-DoF poses (position and orientation) of objects in space is a crucial prerequisite for many robotic manipulation tasks. To aid robotics systems with this task, visual fiducial markers can be attached as physical tags to the object as visual landmarks [11, 15, 8]. When regressing pose without such markers, deep neural networks have achieved state of the art accuracy performance and are commonly applied in robotic manipulation settings [17, 2, 6, 13]. The prevalence of deep neural networks for pose estimation motivates the present paper, which sets out to understand how pose estimation performance can be improved by modifying the objects themselves.

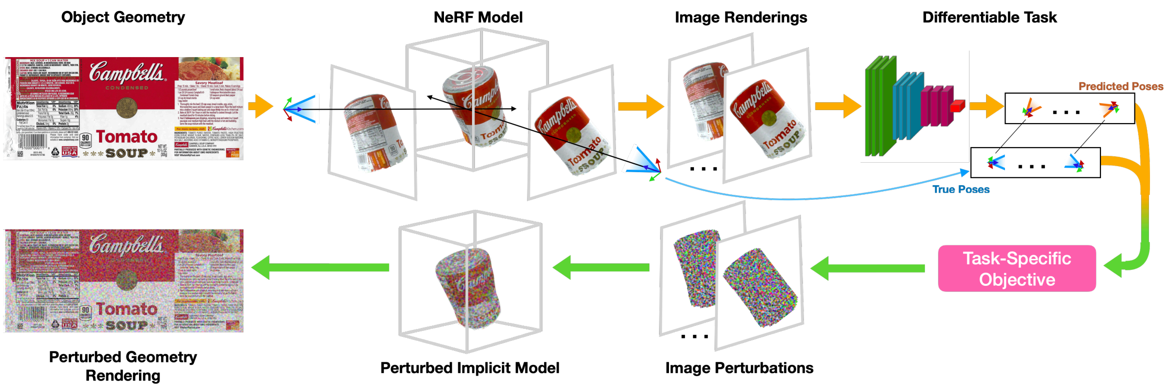

Fig. 1. NeRF-Frenemy architecture diagram. The object geometry and initial coloring is encoded in the NeRF model, which then renders the object at the ground truth pose. NeRF-Frenemy finds a minimal perturbation that maximizes the performance of a pre-trained model on a given task (e.g. object segmentation)

B. Adversarial Examples

It has been shown that only minimal input perturbations are required to produce large changes in neural network inferences [14, 5]. These adversarial approaches have been demonstrated to be able to bridge the sim-to-real gap and work on physically deployed networks [3].

Although most of the work on adversarial attacks focuses on classification, recent work demonstrates the benefit of adversarial training for improving 6-DoF object pose estimation [19] and human pose estimation [1, 7, 16].

This work co-opts the adversarial approach to instead use gradients to influence the inputs constructively rather than antagonistically. The goal is to minimally refine an object's colors or geometry in order to maximize performance against a task-specific model.

C. 3D Object Representations

There are many different options for representing an object in an implicit, differentiable manner. These options include occupancy nets as in Mescheder et al. [9], deep signed distance functions as in Park et al. [12], Neural Radiance Fields as in Mildenhall et al. [10] or plenoptic voxel grids as in Yu et al. [18]. This work utilizes a representation similar to FastNeRF as proposed by Garbin et al. [4] in order to improve rendering time.

III. Method

The proposed NeRFrenemy framework consists of three primary components: a base NeRF capable of rendering the original object, a perturbation NeRF containing the modifications to the original object, and a fixed a priori estimator we seek to optimize against. We perform a forward render and estimation pass followed by a backward pass to optimize the perturbation NeRF as illustrated in Fig. 1.

While our method in theory could be applicable to any differentiable estimator, we focus on PoseCNN due to its general prevalence within the robotics community, access to a high quality open source implementation with pretrained models, and abundance of available data with which to test. In particular, our study focuses on the image segmentation output of PoseCNN.

A. Forward Pass

The purpose of the forward pass is to generate synthetic input data for the task-specific estimator. NeRF-based renderers operate against an implicit space by evaluating a multi-layered perceptron network (MLP) conditioned on spatial location [x, y, z] along with view azimuth and elevation [θ, ϕ] at various locations along projected epipolar rays in order to produce color and density [$\hat{r}$, $\hat{g}$, $\hat{b}$, $\hat{\sigma}$] estimates. These estimates are then combined to produce per-pixel color and depth estimates via the volumetric rendering function. For a full description of this process, see Mildenhall et al. [10].

In this study, we assume the task-specific optimization will affect a single, known object in the scene. Thus, in our study a single NeRF model is learned only of the object we seek to optimize.

If background information is necessary for the task-specific estimator ($G(\cdot)$) to function (which is the case for PoseCNN), then the rendering ($\hat{C}$) can be composited onto an observation C to produce $\hat{C}_{comp}$ either by using the ground truth mask Z from the original dataset that produced C, or by taking only the pixels from $\hat{C}$ which contain depth information (which serves as an estimate of Z). Because we intend to perform optimization against a perturbation and not the full object, we need to utilize two separate NeRF renderers, one which contains the base object's geometry information (F_{base}), and another which contains the current perturbation state ($F_{perturb}$). $F_{perturb}$ is initially trained to produce zeros, so that no perturbation is applied at the beginning of the optimization process.

After the rendering is produced, it is fed through the PoseCNN or other task-specific network $G(\hat{C}_{comp})$ to produce the output prediction $\hat{Y}$, which for this work is a segmentation of the input image.

Fig. 2. An example rendering from the trained NeRF prior to learning perturbations.

B. Backward Geometry Modification

Subsequent to the forward pass, we optimize the perturbation network according to the loss function described in Eq. (1). L_{task} represents the loss function for the task-specific model (e.g. intersection-over-union, Euclidean norm, or another custom metric), $\|\hat{Y}_{perturb}\|$ represents the Euclidean norm of the perturbation NeRF's output, and λ is a regularization weight that is hand-tuned.

$$L(\hat{Y}_{comp}, \hat{Y}_{perturb}, Y) = \lambda * \|\hat{Y}_{perturb}\| + L_{task}(\hat{Y}_{comp}, Y) \tag{1}$$

In essence, the goal of this loss function is to minimize the loss on the task-specific model while simultaneously minimizing perturbation magnitude. This approach is common in adversarial learning, excepting that the chosen L_{task} for adversarial objectives is often chosen to be a loss towards an incorrect answer, as opposed to the constructive one in this work. The gradient of L is then calculated with respect to the perturbation NeRF's parameters, and passed to an Adam optimizer for the update step.

IV. Results

This section presents a preliminary result for the proposed method. We initially train a FastNeRF style model on a soup can using a subset of data from the YCB-Video Dataset which contains no clutter affecting the soup can. No other efforts were made to account for scene variances in lighting, pose biases, or other known issues related to NeRF training. As a result, the renderings of the can are not photo-realistic - although they do remain recognizable as the relevant object to a human. An example rendering of the can from the final trained model is shown in Fig. 2.

After training, we selected an unseen frame from the YCB-Video Dataset and allowed the proposed pipeline to perturb only the $[R, G, B]$ outputs of the base NeRF. This constraint ensures that only the color information was changed by the pipeline and not the geometry. Due to the memory limitations associated with the RTX 3070 GPU used in this experiment, it was not possible to perform a rendering of the entire soup can object and perform optimization against the perturbation NeRF. Therefore, $N = 12000$ epipolar rays were uniformly randomly sampled from the ground truth segmentation mask, Z, and composited onto the original YCB-Video frame. The pre-trained model from PoseCNN for the soup can was used for the task-specific model, in which the task-specific loss was computed as shown in Equation 2, where $\hat{Z}$ is the predicted segmentation probability from the PoseCNN model, and i is the indexing variable for a uniformly sampled pixel.

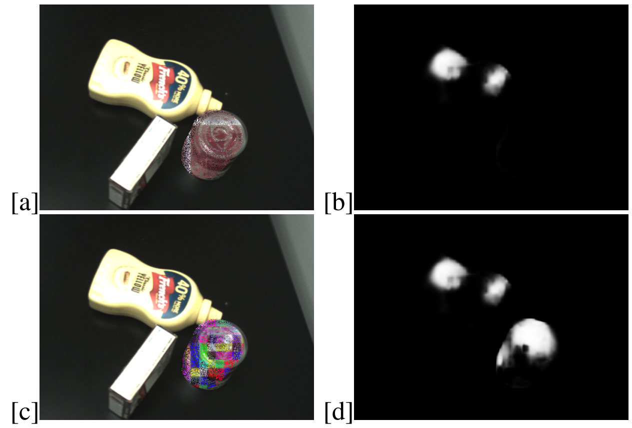

Fig. 3. (a) At iteration 0, the composited scene of the original coloring of the soup can and (b) the corresponding segmentation output of the PoseCNN model. (c) The scene at iteration 999 visualized with a proposed change in coloring for the soup can from NeRF-Frenemy and (d) the new output of the PoseCNN segmentation model based on the perturbed coloring. Note the segmentation for the can, while previously missed completely, is able to be partially produced.

$$L(\hat{Z}) = \frac{\sum_{i=0}^{N}(1 - \hat{Z}_i)}{N} \tag{2}$$

In this experiment, the learning rate for the Adam optimizer was set to 0.75 and λ was set to $1e - 6$. The optimization routine was run for 1000 iterations. Figure 3 shows the composited renderings and segmentation visualizations for the first and final iteration of the training process. In these results, it is clear that the network is successfully optimizing against the underlying PoseCNN network, as the object is not detected at all initially outside of some false positives by the mustard bottle, before being largely discovered by the final iteration. However, the final perturbation values remain very large (as evidenced by the final rendering looking very different from the original can) so further loss formulation and hyper-parameter tuning is required.

V. Conclusion

Preliminary experimentation has shown that NeRF-Frenemy can yield renderings that improve on a chosen task's performance metric. Further work remains to be done to optimize NeRF-Frenemy's rendering fidelity, and also to ensure that this approach can generalize sufficiently to bridge the sim-to-real gap.

References

[1] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1212–1221, 2017.

[2] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao-blackwellized particle filter for 6d object pose tracking. In *Robotics: Science and Systems (RSS)*, 2019.

[3] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[4] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3003–3013, 2021.

[7] Naman Jain, Sahil Shah, Abhishek Kumar, and Arjun Jain. On the robustness of human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 29–38, 2019.

[8] Luis A Mateos. Apriltags 3d: dynamic fiducial markers for robust pose estimation in highly reflective environments and indirect communication in swarm robotics. *arXiv preprint arXiv:2001.08622*, 2020.

[9] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[10] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[11] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. doi: 10.1109/ICRA.2011.5979561.

[12] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[13] Nuno Pereira and Luís A Alexandre. Maskedfusion: Mask-based 6d object pose estimation. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 71–78. IEEE, 2020.

[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[15] Daniel Wagner and Dieter Schmalstieg. Artoolkitplus for pose trackin on mobile devices. In *12th Computer Vision Winter Workshop 07*, pages 139–146. ., 2007. Computer Vision Winter Workshop : CVWW 2007, CVWW 2007 ; Conference date: 06-02-2007 Through 08-02-2007.

[16] Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11855–11864, 2021.

[17] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018.

[18] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021.

[19] Jinlai Zhang, Weiming Li, Shuang Liang, Hao Wang, and Jihong Zhu. Adversarial samples for deep monocular 6d object pose estimation. *arXiv preprint arXiv:2203.00302*, 2022.