Density-based Feasibility Learning with Normalizing Flows for Introspective Robotic Assembly

Anonymous

***Abstract*—*Machine Learning (ML)* models in *Robotic Assembly Sequence Planning* (RASP) need to be introspective on the predicted solutions, i.e. whether they are feasible or not, to circumvent potential efficiency degradation. Previous works need both feasible and infeasible examples during training. However, the infeasible ones are hard to collect sufficiently when re-training is required for swift adaptation to new product variants. In this work, we propose a density-based feasibility learning method that requires only feasible examples. Concretely, we formulate the feasibility learning problem as Out-of-Distribution (OOD) detection with *Normalizing Flows* (NF), which are powerful generative models for estimating complex probability distributions. Empirically, the proposed method is demonstrated on robotic assembly use cases and outperforms other single-class baselines in detecting infeasible assemblies. We further investigate the internal working mechanism of our method and show that a large memory saving can be obtained based on an advanced variant of NF.**

I. Introduction

To embrace the trend of shorter product life cycles and greater customization, RASP empowered with ML models for productivity enhancement has received more attention over the past years [2, 12, 10, 22]. However, *data-driven* models are reported to behave unreliably with inputs differing from the training distribution, e.g., assemblies with distinct customization [15]. In other words, the assembly robot is *unaware* of the predicted solution's feasibility, which requires an intrinsic understanding of the geometry of assemblies and the capability of the robotic system [11]. This lack of introspection can lead to prolonged planning time induced from re-planning after failed execution of an infeasible plan. To address this issue, feasibility learning has been studied [18, 6, 19, 20, 2] based on a setting with *infeasible assemblies included*. We argue that this setting is undesirable in practice because of the risk of incomplete coverage of all possible infeasible cases and high time costs for generating sufficient infeasible training cases. These aggravate the situation when flexible and efficient adaptation across different product variants is required.

To establish introspection for assembly robots with *only feasible assemblies* in mind, we seek to model the feasibility of an assembly with NF, which are a powerful class of generative models excelling at density estimation [5]. Concretely, we train the NF model with *Maximum Likelihood Estimation* (MLE) based on *feasible assemblies alone* to estimate the density of *In-Distribution* (ID) data, i.e. feasible assemblies. Hence, infeasible assemblies can be detected via a lower predicted likelihood as *Out-of-Distribution* (OOD).

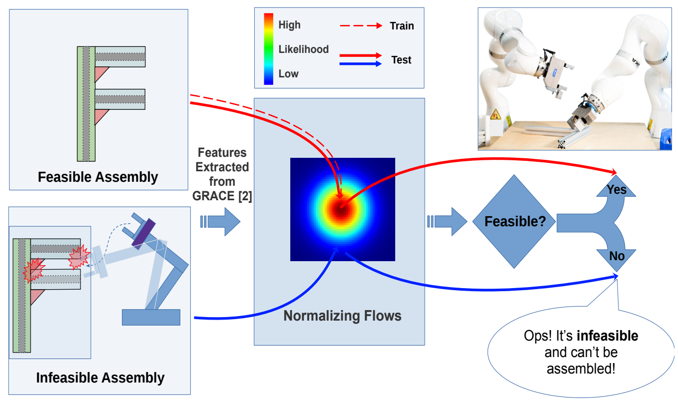

Fig. 1: **Overview of the proposed method** on an assembly scenario with a dual-armed robotic system (used in our setting). The distribution of feasible assemblies is modeled during training with NF. In test time, infeasible assemblies are identified by their low-likelihood.

We examine the proposed idea in a robotic assembly use case, in which different types of aluminum profiles are assembled with a dual-armed robot to create target structures (see Fig. 1). We collected assembly data in simulation and trained the NF on features of *only feasible* assemblies extracted from the *Graph Assembly Processing Networks* (GRACE) proposed in [2]. The NF model is then used to predict the likelihood of test data which includes both feasible and infeasible assemblies. As we learn the feasibility by estimating the density of feasible cases, the predicted outputs from NF represent how likely the given assemblies are feasible. Based on a threshold selected on a validation set, we can then detect infeasible assemblies. Empirically, we demonstrate better results with the proposed method against other baselines on detecting infeasible assemblies in terms of *Area Under the Receiver Operating Characteristic Curve* (AUROC) in the setting where only feasible assemblies are available. We further investigate the major contributing factors of NF and significantly decrease the memory costs (i.e., number of network layers) by employing a more elaborate base distribution [16].

II. Related Work

A. Feasibility Learning

The major body of work on feasibility learning is concentrated on plan or action feasibility learning in TAMP, while our goal is to learn the feasibility of assemblies directly by distilling the knowledge of assembly geometry and capability of the robot system. Wells et al. [18] trained a feature-based SVM model to directly predict the feasibility of an action sequence based on experience, which is hard to scale to scenarios with different numbers and types of objects. Driess et al. [6] and a recent follow-up [19] predict if a mixed-integer program can find a feasible motion for a required action based on visual input. Besides, Yang et al. [20] predict a plan's feasibility with a transformer-based architecture using multi-model input embeddings. Different from us, these methods work in a two-class setting, requiring failing action sequences to be included in the training set and then use binary feasibility classifiers.

B. Normalizing Flows for Out-of-Distribution Detection

NF [4] are a family of deep generative models with expressive modeling capability for complex data distributions where both sampling and density evaluation can be efficient and exact. Among a diverse set of flow architectures, Affine Coupling Flows [5] have gained huge popularity for their scalability to big data with high dimensionality and efficiency for both forward and inverse evaluation. These merits make NF more practically advantageous for OOD detection [8] when compared with other more principled but run-time inefficient uncertainty estimation methods [9]. In the context of task-relevant OOD detection, the practice of PostNet [3] of operating on feature embeddings, provides a more reasonable modeling ability. The potentials of NF for OOD detection have been demonstrated in other domains [13, 21], inspiring us to use them for feasibility learning.

III. Method

A. Problem Setting

Our goal is to predict the feasibility of assemblies relying only on feasible ones by formulating the problem as an OOD detection. Given a data-set $\mathcal{D}$ of N feature embeddings of feasible assemblies $\{\mathbf{a}_i\}_{i=1}^{N}$, where $\mathbf{a}_i \in \mathcal{R}^h$ is drawn from an unknown distribution $P_{feasible}$ with *Probability Density Function* (PDF) p_f, a density estimator, denoted by $q_\theta : \mathcal{R}^h \rightarrow \mathcal{R}$, approximates the true p_f with MLE for its parameters θ based on $\mathcal{D}$. During inference, given a threshold $\delta \in \mathcal{R}$, the feature of a test assembly $\hat{\mathbf{a}}_i$ is classified as OOD, i.e. infeasible, if $q_\theta(\hat{\mathbf{a}}_i) < \delta$, otherwise as ID, i.e. feasible.

B. Density-based Learning with NF

In this work, NF are used to estimate the density of feasible assemblies. NF, denoted by $f_\theta : \mathcal{R}^h \rightarrow \mathcal{R}^h$, are defined by a chain of *diffeomorphisms* (invertible and differentiable mappings) that transform a base distribution $p(\mathbf{z}),\ \mathbf{z} \in \mathcal{R}^h$ (e.g. an isotropic Gaussian) to the data distribution q_θ (in our case p_f). Based on the Change-of-Variables formula, the likelihood of an embedding of an assembly is obtained by

$$q_\theta(\mathbf{a}) = p(f_\theta^{-1}(\mathbf{a})) \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{a})}{\partial \mathbf{a}} \right) \right| \tag{1}$$

θ is optimized with MLE based on feasible data only, where the log likelihood is defined as:

$$\log q_\theta(\mathbf{a}) = \log p(f_\theta^{-1}(\mathbf{a})) + \log \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{a})}{\partial \mathbf{a}} \right) \right| \tag{2}$$

To this end, the inverse flow f^{-1} and the log determinant of the Jacobian need to be tractable and efficient. We employ the Real-NVP [5] that is composed of multiple layers of affine coupling flows. As the input to the NF, a data-set of feature embeddings for feasible assemblies $\mathcal{D}$ is extracted from a pre-trained GRACE [2], which represents each assembly structure as a graph of its parts and their respective surfaces. To create a single feature embedding per assembly, a channel-wise mean pooling is applied on the graph's part nodes. Different to previous works, the dimension of this embedding is independent of the number of assembly parts.

During inference, given a test assembly embedding, the trained NF q_θ predicts a log-likelihood score and determines its feasibility based on a pre-defined threshold δ, which we selected with a validation set.

IV. Experiments

A. Data-set

We applied an in-house simulation software to randomly generate synthetic assemblies, each with 5 or 6 aluminum parts. The software was tasked with putting together these structures with brute-force search while considering geometry restrictions and those imposed by the capabilities of the dual-armed robotic system *KUKA LBR Med* (seen in Fig. 1). We label structures that were successfully assembled as feasible and ones for which the software failed as infeasible. The resulting data-set consists of 6036 5-parts and 2865 6-parts assemblies. For the training set, we used feasible-labeled assemblies alone. The validation and testing sets were balanced with both feasible and infeasible assemblies[1].

B. Implementation Details

We pre-trained GRACE [2] with its default parameters to retrieve a 94-dimensions embedding per assembly. We implemented the NF model using [17] and experimented with Gaussian and Resampling [16] base distributions. For training the NF, we chose a batch size of 32 and a learning rate of $1e-5$ with Adam optimizer. The number of coupling flows was chosen with hyper-parameter search on a validation set. Each affine coupling flow contained 4 layers with 94 hidden channels per layer.

We measure the separation between the feasibility classes with the binary classification metrics *False Positive Rate* (FPR) and *True Positive Rate* (TPR) to derive an AUROC score. In this setting, a positive instance is a feasible assembly and a negative an infeasible one.

[1]This is still a single-class training setting since the validation set is only used for model selection.

Classifier	AUROC (↑) 5-parts	6-parts
GRACE + NF, Gaussian dist., 749 layers (ours)	**0.85**	**0.83**
GRACE + NF, Resampling dist., 109 layers (ours)	0.83	-
OC-SVM [14]	0.74	0.59
GRACE [2]	0.61	0.57

TABLE I: Feasibility classifiers AUROC score on balanced test sets of 5- and 6-part assemblies.

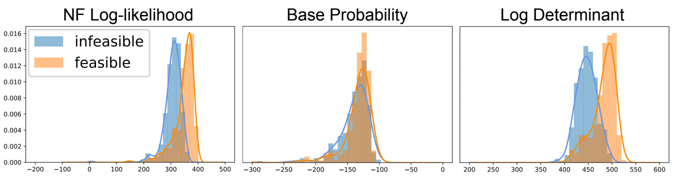

Fig. 2: **NF log-likelihoods for feasible and infeasible assemblies** with 5-parts (left), is a sum of the base probability (middle) and the transformation matrices log determinant (right). Best viewed in color.

C. Results

In Table I, we compare our method to baselines on predicting the feasibility of 5- and 6-part assemblies. The NF model with Gaussian base distribution achieves the highest score with a deep 749-layered network, outperforming the *One-class SVM* (OC-SVM) [14] and the naive GRACE [2]. In this setting, GRACE predicts an assembly sequence for a test instance and infer the assembly's feasibility based on the success of its sequencing process. More practically relevant, the NF variant with the more expressive Resampling base distribution [16] can reach comparably good results with a much smaller network (109 vs. 749 layers). This benefit of memory efficiency is highly relevant for robotic systems with only restricted computation resources (e.g., mobile manipulators).

D. Discussion

For an insight into how NF works on feasibility learning, we study the impacts of the flow transformations from the perspectives of two quantities: 1. likelihoods; 2. sample coordinates. While the former represents the density estimation ability of NF, the latter provides us a hint on how NF shifts the samples from the flow input space into its latent space.

a) Likelihoods Ablation: The NF log-likelihood estimation in Eq. 2 is a sum of two terms: the density of the base distribution and the log-determinant of the Jacobian of the flow transformation. To understand the contribution of each of these to the model's estimation, we plot their values separately for the model with Gaussian base distribution in Fig. 2. As expected, the determinants are the main contributing factor to the final scores, whereas the values produced by the base distribution act as a normalization term.

b) Samples Visualization: We visualize the coordinates of the embeddings in the input space (as created by the GRACE feature extractor) and in the NF latent space with

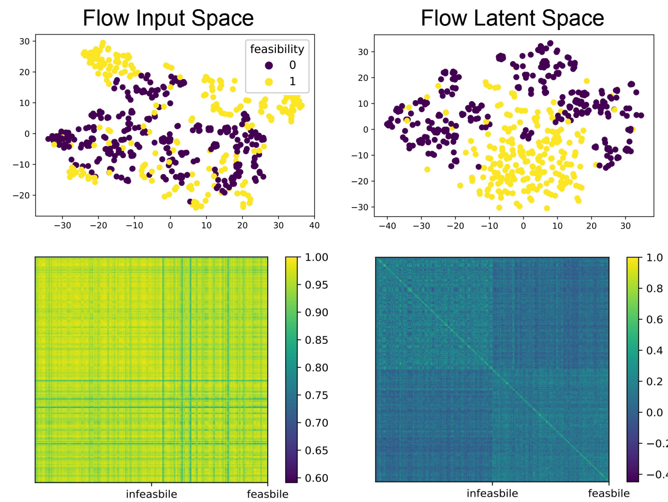

Fig. 3: **Samples visualization in NF input (left) and latent (right) spaces**. At the top, t-SNE shows that samples mapped by NF are "normalized", pulled together to a compact cluster. At the bottom, *Cosine Similarities* between feasible and infeasible assemblies are more distinct after the transformation, verifying the "normalization". Best viewed in color.

t-distributed Stochastic Neighbor Embedding (t-SNE) and similarity matrices (Fig. 3). As shown in the t-SNE visualization, the samples of feasible assemblies are pulled together and hence clustered more compactly when compared to those in the input space before the flow transformation. This is verified again in the similarity matrices at the bottom, where the distances between feasible samples are smaller than those of infeasible ones after. These results show us that the flow transformation indeed *"normalizes"* the inputs in terms of both likelihood computation and geometrical coordinates. This observation also confirms the finding of better OOD detection performance in the flow latent space [7], which is worth exploring for more effective feasibility learning algorithms, which we leave for future work. Besides, a further improvement could be archived by encouraging the feature extractor GRACE to grasp semantics that are more closely related to the feasibility task, as suggested by [8].

V. Conclusion

In this work, we seek to address feasibility prediction for data-driven methods in RASP with NF relying only on feasible examples. With the formulation of density-based OOD detection, we develop an effective feasibility prediction algorithm based on feature embeddings from a pre-trained processing network. The empirical experiments on detecting infeasible assemblies in simulation present promising results, which outperform the baselines. We further dug into the internal working mechanism of NF for this use case and found insightful observations, which can provide more understanding to inspire other researchers for further improvements in this direction. For future research, we suggest introducing explainability into this setting with a gradient map in respect to the input, which can guide the user in altering the structure and enable its assembly, i.e., counter-factual explanation [1].

References

[1] Matan Atad, Vitalii Dmytrenko, Yitong Li, Xinyue Zhang, Matthias Keicher, Jan Kirschke, Bene Wiestler, Ashkan Khakzar, and Nassir Navab. Chexplaining in style: Counterfactual explanations for chest x-rays using stylegan. *arXiv preprint arXiv:2207.07553*, 2022.

[2] Matan Atad, Jianxiang Feng, Ismael Rodríguez, Maximilian Durner, and Rudolph Triebel. Efficient and feasible robotic assembly sequence planning via graph representation learning. *arXiv preprint arXiv:2303.10135*, 2023.

[3] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33: 1356–1367, 2020.

[4] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[6] Danny Driess, Ozgur Oguz, Jung-Su Ha, and Marc Toussaint. Deep visual heuristics: Learning feasibility of mixed-integer programs for manipulation planning. In *2020 IEEE ICRA*, pages 9563–9569. IEEE, 2020.

[7] Dihong Jiang, Sun Sun, and Yaoliang Yu. Revisiting flow generative models for out-of-distribution detection. In *ICLR*, 2022.

[8] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.

[9] Jongseok Lee, Matthias Humt, Jianxiang Feng, and Rudolph Triebel. Estimating model uncertainty of neural networks in sparse information form. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 5702–5713. PMLR, 13–18 Jul 2020.

[10] Lin Ma, Jiangtao Gong, Hao Xu, Hao Chen, Hao Zhao, Wenbing Huang, and Guyue Zhou. Planning assembly sequence with graph transformer. *arXiv preprint arXiv:2210.05236*, 2022.

[11] Ismael Rodrıguez, Korbinian Nottensteiner, Daniel Leidner, Michael Kaßecker, Freek Stulp, and Alin Albu-Schäffer. Iteratively refined feasibility checks in robotic assembly sequence planning. *IEEE RAL*, 4(2):1416–1423, 2019.

[12] Ismael Rodríguez, Korbinian Nottensteiner, Daniel Leidner, Maximilian Durner, Freek Stulp, and Alin Albu-Schäffer. Pattern recognition for knowledge transfer in robotic assembly sequence planning. *IEEE RAL*, 5(2): 3666–3673, 2020.

[13] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF WACV*, pages 1907–1916, 2021.

[14] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999.

[15] Rohan Sinha, Apoorva Sharma, Somrita Banerjee, Thomas Lew, Rachel Luo, Spencer M Richards, Yixiao Sun, Edward Schmerling, and Marco Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.

[16] Vincent Stimper, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Resampling base distributions of normalizing flows. In *AISTATS*, pages 4915–4936. PMLR, 2022.

[17] Vincent Stimper, David Liu, Andrew Campbell, Vincent Berenz, Lukas Ryll, Bernhard Schölkopf, and José Miguel Hernández-Lobato. normflows: A PyTorch Package for Normalizing Flows. *arXiv preprint arXiv:2302.12014*, 2023.

[18] Andrew M Wells, Neil T Dantam, Anshumali Shrivastava, and Lydia E Kavraki. Learning feasibility for task and motion planning in tabletop environments. *IEEE RAL*, 4(2):1255–1262, 2019.

[19] Lei Xu, Tianyu Ren, Georgia Chalvatzaki, and Jan Peters. Accelerating integrated task and motion planning with neural feasibility checking. *arXiv preprint arXiv:2203.10568*, 2022.

[20] Zhutian Yang, Caelan Reed Garrett, and Dieter Fox. Sequence-based plan feasibility prediction for efficient task and motion planning. *arXiv preprint arXiv:2211.01576*, 2022.

[21] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *Computer Vision–ECCV 2020, Proceedings, Part III 16*, pages 102–117. Springer, 2020.

[22] M. Zhao, X. Guo, X.and Zhang, Y. Fang, and Y. Ou. Aspw-drl: assembly sequence planning for workpieces via a deep reinforcement learning approach. *Assembly Automation*, 40:65–75, 2020. ISSN 0144-5154.