Topology-Matching Normalizing Flows for Out-of-Distribution Detection in Robot Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract: To facilitate reliable deployments of autonomous robots in the real world, Out-of-Distribution (OOD) detection capabilities are often required. A powerful approach for OOD detection is based on density estimation with Normalizing Flows (NFs). However, we find that prior work with NFs attempts to match the complex target distribution topologically with naïve base distributions leading to adverse implications. In this work, we circumvent this topological mismatch using expressive class-conditional base distributions that we train with an information-theoretic objective to match the required topology. The proposed method enjoys the merits of wide compatibility with existing learned models, efficient runtime, and low memory overhead while enhancing the OOD detection performance. We demonstrate the benefits of our method in density estimation, 2D object detection benchmarks and in particular, showcase the applicability in a real-robot deployment.

Keywords: Normalizing Flows, Out-of-Distribution, Robotic Introspection

1 Introduction

The reliable identification of Out-of-Distribution (OOD) data, which is not well represented in the training set, poses a pressing challenge on the path towards trustworthy open-world robotic systems such as self-driving cars [1], delivery drones [2] or healthcare robots [3]. For example, with widespread adoption in the perception pipeline, existing object detectors have been reported to overconfidently misclassify an OOD object into a known class, which might obfuscate the decision-making module and eventually cause catastrophic consequences in safety-critical scenarios [1, 4, 5].

Normalizing Flows (NFs) are a popular class of generative models [6, 7, 8, 9] that may be used for OOD detection. NFs represent complex probability distributions [10] with a learnable series of transformations from a simple base distribution to a complex target distribution. However, NFs' expressivity [11, 12, 13] and numerical stability [14, 15] is limited by a fundamental constraint: the supports of the base and target distribution should preserve *similar topological properties* (Definition 3.3.10 in Runde [16]). The topological properties subsume different geometrical characteristics of the target distribution, including its continuity, the number of connected components, or the number of modes. Increasing the capacity of the transformation may mitigate this constraint. Yet, this raises computation and memory demands [11, 17, 12]. An alternative to overcome the topological mismatch is to increase the flexibility of the base distribution, which is surprisingly under-explored in the OOD detection literature.

Therefore, we propose to equip NFs with efficient but flexible base distributions for OOD detection in robot learning. Concretely, we replace the frequently used uni-modal Gaussian base distribution with the Conditional Resampled Base Distributions (cRSB), a class-conditional version of a learnable base distribution for mitigating the topological problem in NFs – Resampled Base Distributions (RSB) [13]. cRSB can learn the required topological properties, like adapting the number of modes, to match the unknown topological structure of the latent class-specific target distribution (Figure 1). Moreover, we adapt our cRSB with an adapted Information Bottleneck (IB) objective [18] to bal-

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

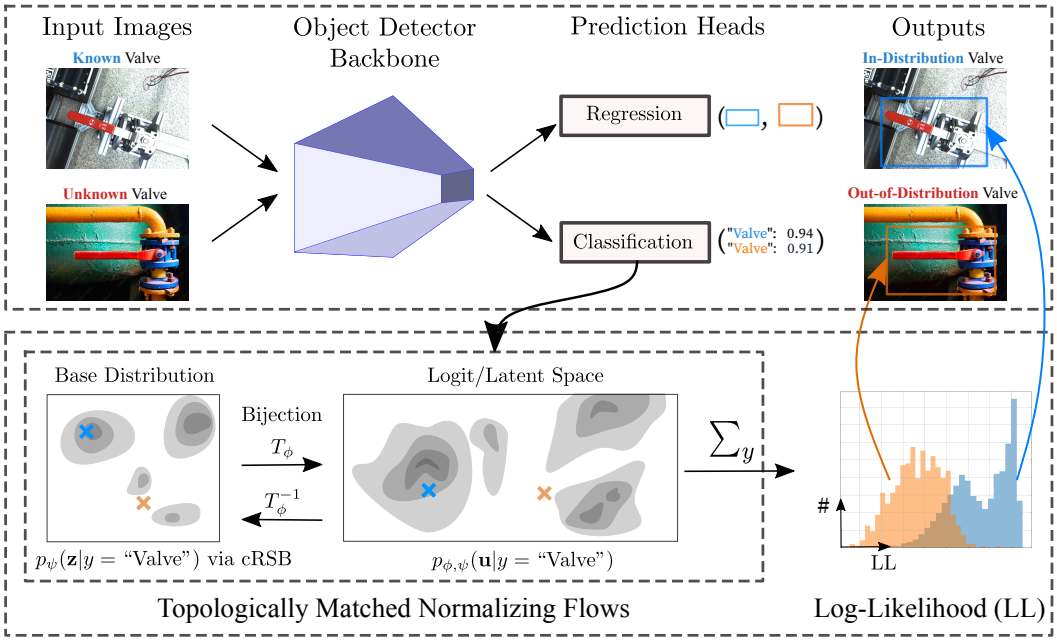


Figure 1: **Proposed architecture**. We overcome the topological mismatch problem in NFs to accurately model In-Distribution (ID) density. That is, the cRSB base distribution $p_\psi(\mathbf{z})$ can, e.g., adapt the numbers of modes to match latent distribution $p(\mathbf{u}|\mathbf{y})$. Therefore, we identify OOD objects by low predicted log-likelihoods more reliably than before (best viewed in color).

ance fusing class-conditional information with the marginalized density estimation capabilities in NFs. IB [19] is an information-theoretic objective to incorporate task-specific details e.g. class conditions, which are commonly ignored in pure generative modeling. This delivers a topology in the base distribution that is more accurately aligned to the one in the target distribution (see Figure 3).

Our OOD detection approach using topology-matching NFs is powerful and yet resource-efficient for open-set object detection. It is applicable to diverse object detectors (e.g. Faster-RCNN [20] and Yolov7 [21] in this work) with minor changes and no loss of prediction performance. Moreover, our approach is sampling-free, i.e., only a single forward pass is required for efficient test-time inference while keeping the space memory tractable. In our experiments, we empirically validate the efficacy of the proposed idea using synthetic density estimation and real-world object detection based on two adapted public benchmark datasets (Pascal-VOC [22] and COCO [23]). Here, within standardized benchmarks, we demonstrate improvements over the state-of-the-art. To further validate the applicability in robotics, we deploy an object detector equipped with the proposed method on an aerial manipulation robot in inspection and maintenance scenarios.

Contributions. Our main contribution is the NFs-based OOD detection method that adapts to the topological structure of the (unknown) target distribution while taking class-conditional information into account. We show that training with IB yields effective representation with superior OOD detection capabilities. We conduct a comprehensive empirical evaluation using both synthetic density estimation and public object detection datasets followed by a real-world robot deployment, which overall shows the effectiveness of the proposed approach.

2 Method

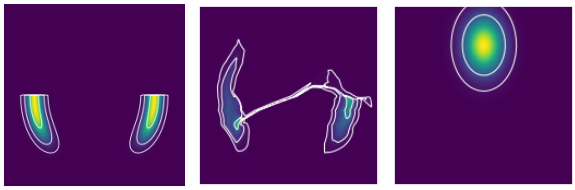
Problem Formulation Given an image $\mathbf{x} \in \mathcal{X}$ and a trained object detector F_θ that localizes a set of objects with corresponding bounding box coordinates $\mathbf{b}_i \in \mathcal{R}^4$ as well as class label $y_i \in \mathcal{Y} = \{1, 2, ..., C\}$, the task is to distinguish if $(\mathbf{x}, \mathbf{b}_i, y_i)$ is ID, i.e., drawn from $\mathcal{P}_{id}$, or OOD, i.e., belongs to the unknown distribution $\mathcal{P}_{ood}$. As discussed, a powerful OOD detection can be

obtained via density estimation using NFs. This density estimator identifies OOD objects with low likelihoods after being trained *only* on data drawn from $\mathcal{P}_{id}$. Following relevant prior [24, 25], we use the semantically rich logit space (pre-softmax layer) for density estimation. However, our method generalizes to other (high-dimensional) latent feature spaces.

NFs are known to be universal distribution approximators [10]. That is, they can model a complex target distribution $p(\mathbf{u})$ on a space $\mathcal{R}^d$ by defining $\mathbf{u}$ as a transformation $T_\phi : \mathcal{R}^d \to \mathcal{R}^d$ parameterized by ϕ from a well-defined base distribution $p_\psi(\mathbf{z})$ wiht parameters ψ:

$$\mathbf{u} = T_\phi(\mathbf{z}) \;\text{ where }\; \mathbf{z} \sim p_\psi(\mathbf{z}) \tag{1}$$

where $\mathbf{z} \in \mathcal{R}^d$ and p_ψ is commonly chosen as a uni-modal Gaussian. By designing T_ϕ to be a *diffeomorphism*, that is, a bijection where both T_ϕ and T_ϕ^{-1} are differentiable. We can compute the likelihood of the input $\mathbf{u}$ *exactly* based on the change-of-variables formula [26]:



(a) $p(\mathbf{u}|y=0)$ (b) $p_{\phi,\psi}(\mathbf{u}|y{=}0)$ (c) $p_\psi(\mathbf{z}|y=0)$

Figure 2: Filament connect modes in the modeled class-conditional distribution (b) if using (trainable) uni-modal base (c) for the multi-modal target (a).

$$p(\mathbf{u}) = p_\psi(T_\phi^{-1}(\mathbf{u}))|\det(J_{T_\phi^{-1}}(\mathbf{u}))|\,, \tag{2}$$

where $J_{T_\phi^{-1}}(\mathbf{u}) \in \mathcal{R}^{d\times d}$ is the Jacobian of the inverse T_ϕ^{-1} with respect to $\mathbf{u}$. When the target distribution is unknown but samples thereof are available, we can estimate the parameter (ϕ, ψ) by minimizing the forward Kullaback-Leibler Divergence (KLD), which is equivalent to maximizing the expected Log-Likelihood (LL).

Topological Mismatch However, since the base distribution $p_\psi(\mathbf{z})$ is usually a uni-modal Gaussian (e.g. Figure 2c) and T_ϕ is a diffeomorphism, problems arise for modeling data distribution with different topological properties. These include well-separated multi-modal distributions or distributions with disconnected components (e.g. Figure 2a). For example, one can see that this leads to density filaments between the modes in Figure 2b. Cornish et al. [11] have shown that flows require a bijection with *infinite bi-Lipshitz constant* if modeling a target distribution with disconnected support using a unimodal base distribution. Besides the diminishing modeling performance, this renders the bijection to be numerically "non-invertible", thus, causing optimization instability during training and unreliability of likelihood calculation.

2.1 Conditional Resampled Base Distributions

One possible partial mitigation is by enriching the expressiveness of the flows. For example, by (a) increasing the number of layers or parameters, (b) using more complex base distributions, or (c) employing multiple NFs. It is important to note that especially (a) and (c) may escalate the computational cost and memory burden, which is unfavorable for robotic systems with restricted computing resources. Moreover, scaling the normalizing flow's expressivity, (a) or (c), often does not increase the stability of the optimization [15] or the likelihood calculation. For these reasons, we pursue (b) and attempt to compensate for the complexity of the transformation with the elasticity of the base distribution. In other words, we use a more flexible but efficient base distribution to trade off a costly but sufficiently expressive bijection of the normalizing flow. This way we aim to capture desirable topological properties of the target distribution [17].

Following prior work [27], to model the fidelitous distribution of data with task-specific conditions, e.g. class labels, we use a class-conditional base distribution. This way we get similar benefits like combining multiple conditional flows (c), however, without having to burden the computational cost on marginalization over classes which requires repeated evaluation of the flows. Even though a class-conditional distribution can specialize on a smaller fraction of the dataset containing similar instances, in real-world applications, it will surely manifest in a multi-modal distribution.

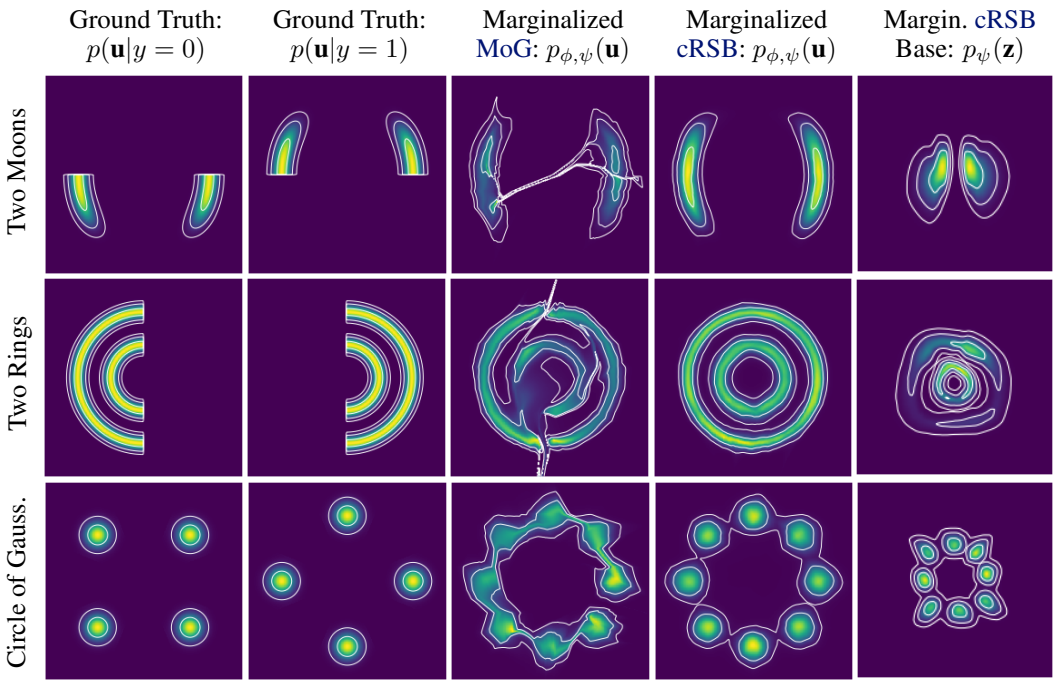


Figure 3: Visualization of density estimation using Real NVP with class conditional MoG, where each class is modeled by a uni-modal Gaussian, and cRSB as well as the class-marginalized density for the base distribution of cRSB.

Therefore, we propose to capture the complex topological properties in the target distribution with a more expressive base distribution instead of the uni-model Gaussian. To the end, we introduce Conditional Resampled Base Distributions (cRSB) by extending a powerful unconditional base distribution Resampled Base Distributions (RSB) [13] with class-conditional modeling. RSB deforms a uni-modal Gaussian in a learnable manner to obtain more complex distributions via Learned accept/reject sampling (LARS) [28]. LARS iteratively re-weighs samples drawn from a proposal distribution $\pi(\mathbf{z})$, e.g. a standard Gaussian, through a learned acceptance function $a_\psi : \mathcal{R}^d \to [0, 1]$. To reduce the computation cost in practice, this process is truncated by accepting the T-th samples if the previous $T-1$ samples get rejected. To take into account class-conditional information, we conditionalize the learnable acceptance function $a_\psi(\mathbf{z}|y)$. As a result, we have the conditional base distribution:

$$p_\psi(\mathbf{z}|y) = (1-\alpha_T)\frac{a_\psi(\mathbf{z}|y)\pi(\mathbf{z})}{Z_k} + \alpha_T \pi(\mathbf{z}), \tag{3}$$

where $a_\psi : \mathcal{R}^d \to [0,1]^C$ and $\alpha_T = (1-Z)^{T-1}$, where Z is the normalization factor for $\pi(\mathbf{z})a_\psi(\mathbf{z}|y)$. This factor can be estimated via Monte Carlo Sampling.

In Figure 3, we contrast the density estimation capabilities of Normalizing Flows (NFs) with the common Mixture of Gaussians (MoG) [8, 27] base distribution and our cRSB on three tasks with class-conditional structure using an appropriate learning objective (see next section). We find that our cRSB learns appropriate topology-matching base distributions (right outer column) and as a result, the respective NFs do not have adverse effects like filaments between the modes.

2.2 Training with Information Bottleneck

Unfortunately, directly training NFs with a conditional base distribution can lead to underperformance as observed in experiments (see Table 2 and appendix) and reported by Fetaya et al. [27]. This attributes to the lack of explicit control for the balance between generative and discriminative modeling in the likelihood-based training objective of NFs. To alleviate this, we train the normalizing flow with a class-conditional base distribution using the so-called Information Bottleneck (IB) objective [19]. The IB minimizes the Mutual Information (MI) $I(U, Z)$ between U and Z, while

simultaneously maximizing the MI $I(Z, Y)$ between Z and Y. Intuitively, the IB trades off between the objectives of modeling the class conditional information $p(u|y)$ with the marginalized density $p(u)$, thus allowing to leverage the class-conditional structure to facilitate more effective density estimation for data characterized with semantic classes.

However, the IB is not directly applicable to latent class-conditional distributions in NFs since the bijection T_ϕ is lossless by design. Thus, for trading off the class-conditional information with density estimation capabilities, we adapt the approach proposed by Ardizzone et al. [18] for our cRSB. Specifically, we inject a small amount of noise ϵ into the input U. This yields an asymptotically exact version of IB and resolves to (more details in appendix):

$$\mathcal{L}_{\text{IBNF}} = \hat{I}(U, Z) - \beta \hat{I}(Z, Y) \tag{4}$$

$$\hat{I}(Z, Y) = \mathbb{E}_{p(y)} \left[\log \frac{p_\psi(\mathbf{z}|y)p(y)}{\sum_{y\prime}(p_\psi(\mathbf{z}|y\prime)p(y\prime))} \right] \tag{5}$$

$$\hat{I}(U, Z) = \mathbb{E}_{p(\mathbf{u}),p(\epsilon)} \left[-\log \sum_{y\prime}(p_\psi(\mathbf{z}|y\prime)) - \log |\det(J_{T_\phi^{-1}}(\mathbf{u}+\epsilon))| \right], \tag{6}$$

where $\mathbf{z} = T_\phi^{-1}(\mathbf{u} + \epsilon)$, $p(\epsilon) = \mathcal{N}(0, \sigma^2 \mathcal{I}_d)$ is a zero-meaned Gaussian with variance σ^2, and β trades off class information and density estimation.

With the flexible conditional base distributions defined in Eq. 3, we can train *topology matching* NFs with IB by substituting cRSB into the conditional base probability $p_\phi(\mathbf{z}|y)$ in Eq. 5. More noteworthy, we observed that the IB is able to regularize the acceptance rate learning for cRSB to better assimilate the topological structure of the target distribution, leading to an overall improved performance on accurately approximating the complex target distribution (see Figure 4).

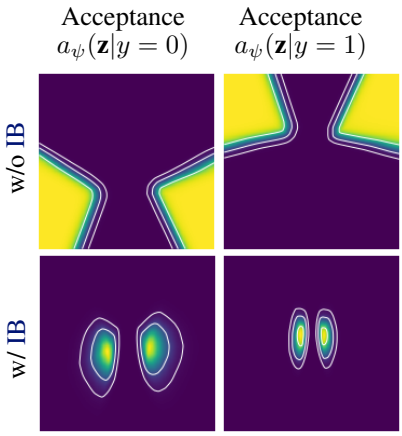


Figure 4: cRSB acceptance rate $a_\psi(\mathbf{z})$ w/o and w/ IB training for Two Moons.

2.3 Detecting OOD Objects

During test time, we detect the Out-of-Distribution (OOD) data based on the Log-Likelihood (LL) estimated from the topology-matching NFs trained with IB as described in previous sub-sections. We *marginalize* the density over classes for the base distribution defined in Eq. 3 and compute the final Log-Likelihood (LL) based on the change-of-variables formula given the logits $\mathbf{u}'$ from the test image and the predicted object bounding box $\{\mathbf{x}', \mathbf{b}'\}$:

$$\text{LL}_{test}(\mathbf{u}') = \log \sum_{y\prime}(p_\psi(T_\phi^{-1}(\mathbf{u}')|y\prime)) - \log |\det(J_{T_\phi^{-1}}(\mathbf{u}'))|. \tag{7}$$

We then expect LL for In-Distribution (ID) objects to be higher than OOD ones. A threshold to distinguish ID and OOD objects can be estimated from the training set.

3 Related Work

Normalizing Flows NFs [29] are a popular class of deep generative models. NFs has shown applicability in a variety of areas such as image generation [30, 31], uncertainty estimation [32, 33, 34] and OOD detection [6, 35]. For NFs, one trend has been designing expressive flow-based architectures. Notable examples are affine coupling flows [30, 31], auto-regressive flows [36, 37], invertible ResNet blocks [38] and ODEs-based maps [39]. The major focus of these works is on reducing computing requirements for Jacobian computations while ensuring that each mapping is invertible.

Another research direction, currently emerging, is on addressing the topological mismatch [29, 10] of NFs. Targeting this problem, some existing works attempt to increase the learning capacity of the transformation via mixture models [40], latent variable models [11, 41] or injecting carefully

specified randomness [42, 12]. These methods are limited in their applicability to robotics because they either increase memory consumption by expanding the width of transformations or approximate the exact likelihood. Recently, these constraints have been addressed by improving the expressivity of the base distribution [13, 17]. In this paper, we build upon this class of methods since they only add slight computation overheads and thus are better suited for applications in robotics.

Normalizing Flows for OOD Detection NFs have been a widely adapted for OOD detection due to its superior density estimation [43]. For example, NFs have demonstrated encouraging OOD detection results with additional refinements for raw data [44, 45, 46] or directly based on task-relevant embeddings [6, 7]. Recently hybrid models [47, 7, 48] have also shown remarkable performance gain on OOD detection by modeling the joint distribution of both data and its class labels. Such works suggest that class labels can provide useful information. However, directly performing class conditional modeling with NFs for OOD detection results in performance degradation. Tishby et al. [19], Ardizzone et al. [18] mitigate such performance degradation by utilizing IB for training NFs. This explicitly controls the trade-off between generative and discriminative modeling [9]. However, these works on OOD detection utilize NFs without much concern for the fundamental topological problem as the first citizen. Therefore, complementary to the aforementioned approaches, we examine the problem of topological mismatch of NFs for OOD detection tasks.

OOD Detection in Object Detectors OOD detection research has focused on image classification tasks [43], which may be limited in relevance to robotic vision. In robotics, we may often need both categorization and localization of objects of interest. Therefore, in the paper, we focus on object detection in open-set conditions. In this domain, uncertainty estimation has been considered propitious for OOD detection but suffered from computation burdens on runtime [49, 50, 51] or memory costs [52]. To address this, one direction is to apply uncertainty estimation techniques for object detection [51, 2], while another popular approach is to explicitly formulate the problem as OOD detection tasks [25, 53, 8, 54, 55]. Amongst them, NFs has been utilized as an expressive density estimator [8, 54]. However, despite the encouraging results, these approaches have not examined the problem of topological mismatch in NFs. As this might prevent additional performance improvements, this work examines the topology-matching NFs for OOD detection in object detectors.

4 Experiments

We next demonstrate the efficacy of our method. First, we evaluate on synthetic density estimation for distributions with distinct topological properties. We then evaluate the OOD detection performance on two object-detection data-sets adapted from their public counterparts [22, 23] for open-set (OS) experiments: Pascal-VOC-OS and MS-COCO-OS. To showcase the practicality, we deploy the object detector equipped with the proposed method on a real aerial manipulation robot to distinguish synthetic and real objects of an industrial valve and a pipe inspection robot, along with the run-time and memory analysis.

Datasets For density estimation, there are three synthetic datasets: two moons, two rings, and a circle of Gaussians. For OOD detection for objection detection, since existing object detection datasets are not ready for fair evaluation [4], we strictly follow the experimental protocol in [25]. In this setting, the first 15 classes and 50 classes are selected as known classes from the Pascal-VOC and MS-COCO datasets respectively, while the remaining classes are set as unknown classes. Therefore, the training sets are then filtered to *keep only known classes*. The original test sets are used for testing as they have both known and unknown classes. For real robot deployment, we use synthetic images of two classes (valve and a crawler robot) [56] for training and real images for testing before deploying on the robot.

Metrics For density estimation, we employ the Kullaback-Leibler Divergence (KLD) between the target and the model distributions. On OOD detection for object detection, we want to identify the OOD objects that are falsely detected and classified as known objects by the detector. To evaluate this ability, we use the Area Under Receiver Operation Curve (AUROC) as a metric with varying

Table 2: OOD detection performance on Pascal-VOC-OS and MS-COCO-OS datasets for different methods based on the Faster-RCNN from 3 random runs. The highest values are marked in **bold** and the second highest in *italics*.

	Pascal-VOC-OS				MS-COCO-OS			
	AUROC	TPR at 5%FPR	TPR at 10%FPR	TPR at 20%FPR	AUROC	TPR at 5%FPR	TPR at 10%FPR	TPR at 20%FPR
Softmax	0.901	60.1	72.8	83.1	0.882	61.3	70.6	78.1
Entropy	0.905	59.8	72.9	82.9	0.903	61.2	70.6	80.2
Ensemble Softmax	0.885	47.8	72.6	83.1	0.898	66.2	73.5	82.3
Ensemble Entropy	0.887	47.8	72.5	83.1	0.906	66.2	73.5	82.3
GMMDet [25]	0.931	70.7	*80.5*	*89.3*	0.924	69.5	80.2	87.9
Flows Gaussian	0.915 ± 0.002	72.2 ± 0.75	77.8 ± 0.89	86.1 ± 0.67	0.924 ± 0.001	68.2 ± 0.73	81.2 ± 0.61	89.4 ± 0.04
Flows GMM	0.919 ± 0.002	69.0 ± 2.4	77.0 ± 2.5	86.5 ± 1.2	0.925 ± 0.001	68.3 ± 0.30	80.5 ± 0.50	89.6 ± 0.05
Flows RSB [13]	0.924 ± 0.003	72.8 ± 0.88	79.3 ± 1.0	87.1 ± 0.82	0.925 ± 0.001	68.6 ± 0.87	81.3 ± 0.31	89.5 ± 0.34
Flows MoG_CLS [27]	0.923 ± 0.001	69.2 ± 1.5	78.2 ± 1.3	88.5 ± 0.82	*0.930* ± 0.001	68.5 ± 0.73	*82.2* ± 0.31	*89.7* ± 0.30
Flows MoG_IB [8]	*0.934* ± 0.002	73.1 ± 1.3	79.6 ± 0.6	87.8 ± 0.2	0.924 ± 0.002	*71.1* ± 0.9	79.6 ± 0.46	88.6 ± 0.63
Flows cRSB_CLS	0.919 ± 0.001	72.5 ± 0.37	78.8 ± 0.27	86.8 ± 0.42	0.924 ± 0.001	68.3 ± 0.14	81.1 ± 0.30	89.3 ± 0.18
Flows cRSB_IB (ours)	**0.946** ± 0.003	**78.5** ± 0.97	**84.0** ± 0.83	**90.8** ± 0.76	**0.934** ± 0.002	**73.3** ± 2.0	**84.3** ± 0.40	**91.3** ± 0.28

thresholds. To note that we set the true positives as the correct predictions and false positives as the incorrect predictions. Moreover, we also report the True Positive Rate(TPR) at different False Positive Rate (FPR) (5%, 10%, 20%), as they represent the performance of the potential operating points for safety-critical applications, which requires the FPR to be sufficiently low.

Implementation Details For density estimation, we use a Real NVP [30] with four layers for the cRSB and five layers for the MoG with randomly initialized trainable mean and variance. For the NSF we use two layers each since the cRSB overhead is negligible here. For object detection, we implement our method with the Glow [31] architecture and train our flows based on logits generated by the pre-trained Faster-RCNN [20] object detectors provided by Miller et al. [25] for a fair comparison. For real robot deployment, we use the one-stage object detector Yolov7 [21]. More details are available in the supplementary materials.

4.1 Density Estimation

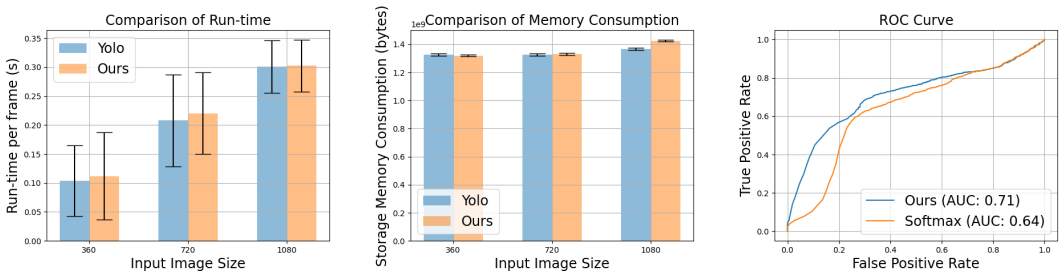
We compare the density estimation performance in Table 1 and provide qualitative results in Figure 3. We find that the cRSB base distribution consistently outperforms the class-conditional Mixture of Gaussians (MoG). The improvements from cRSB are able to generalize across two different NFs architectures, i.e. Real NVP and NSFs.

Table 1: Performance on density estimation for different flow architectures w.r.t. KLD, i.e., $D_{\mathrm{KL}}(p(\mathbf{u}, y)||p_{\phi,\psi}(\mathbf{u}, y))$. Better base distribution is highlighted in bold.

Flow architecture	Real NVP		NSFs	
Base distribution	MoG	cRSB	MoG	cRSB
Two Moons	1.179	**1.066**	0.909	**0.906**
Two Rings	2.032	**1.704**	1.647	**1.602**
Circle of Gaussians	2.335	**1.667**	1.766	**1.653**

4.2 OOD Detection in Object Detection

We compare the flow-based methods with GMMDet [25], Softmax, Entropy and, their Deep Ensemble variants with five models trained from different parameter initialization [52]. Among the flow-based methods, we compare across different base distributions, including unconditional ones (uni-modal Gaussian, MoG, RSB) and their conditional variants (MoG_CLS, cRSB_CLS) [27] and MoG trained with IB (MoG_IB) [8, 18]. From Table 2, we can observe that flows with uni-modal Gaussian are able to provide satisfactory performance while flows with more expressive base distributions such as MoG and RSB can bring more benefits on Pascal-VOC-OS than MS-COCO-OS. When trained with IB, the more flexible conditional base distributions (cRSB_IB) can mostly have better performance gains (on both Pascal VOC and COCO) than its strong competitor (MoG_IB) (only on COCO) in comparison with their counterparts without IB (MoG_CLS). These results demonstrate the effectiveness of cRSB with IB for OOD detection in complicated 2D object detection tasks.



(a) (b) (c)

Figure 5: Results from experiments on a real robot. Run-time, memory consumption, and ROC curve are reported. Compared to vanilla yolo, the proposed method does not yield significant computational costs, while providing performance gains in OOD detection.

4.3 Real Robot Deployment

Next, we validate the suitability of the proposed method for applications in robotics. To do so, we study our method by grounding it in an application scenario of robotic inspection and maintenance. Within such field robotic applications, the OOD objects appear routinely due to the challenges of outdoor environments, which is crucial to avoid false positives. Computational efficiency is another important requirement. The results are reported in Figure 5. First, we compare the runtime and space memory consumption against a vanilla Yolo v7 using the NVIDIA's embedded GPU called Jetson Orin. Our approach is then applied to Yolov7 by training an additional OOD detector based on our topology-matching NFs. The results indicate that the computational overhead of having an OOD detector is relatively small when compared to the vanilla Yolov7. We further show significant performance gains on OOD detection in Figure 5c, where the proposed method can notably achieve higher TPR around the low FPR region, which are commonly used as operating points for the robot. Overall, these experiments validate our claim that our method features efficient runtime inference and cost-effective memory consumption.

5 Limitations

The quality of features can majorly determine the performance of OOD detection. As reported by previous work [25, 8], learning more compact and centralized features can often lead to increased performance for OOD detection while feature collapse can be harmful to OOD detection. Another limiting factor of NFs is their computational cost. We mitigate the topological mismatch by enhancing the expressiveness of the base distribution. However, enhancing the base distribution also comes at a cost. Therefore, we seek to minimize the added computation overhead by balancing the cost of the bijection and base distribution. As a result, our method comes at negligible cost, when we equip the efficient Yolov7 object detector as shown in Figure 5.

6 Conclusion

To endow robots with introspective awareness against OOD data, we propose the effective yet lightweight conditional Resampled Base Distribution (cRSB) for NFs. With our learnable cRSB base distribution, NFs are able to mitigate the fundamental topological mismatch problem, facilitating more effective OOD detection capabilities. We present empirical evidence that NFs equipped with our cRSB and trained with IB objective achieve superior performance both quantitatively and qualitatively. To demonstrate the run-time efficiency and minimum memory overheads, we deployed on a real-robot system. Overall, we expect our encouraging and practical-relevant results stemming from an enriched base distribution to push forward the direction of NFs-based OOD detection in robot learning.

References

[1] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena. Out-of-distribution detection for automotive perception. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2938–2943. IEEE, 2021.

[2] J. Lee, J. Feng, M. Humt, M. G. Müller, and R. Triebel. Trust your robots! predictive uncertainty estimation of neural networks with sparse gaussian processes. In *Conference on Robot Learning*, pages 1168–1179. PMLR, 2022.

[3] J. Feng, J. Lee, M. Durner, and R. Triebel. Bayesian active learning for sim-to-real robotic perception. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10820–10827. IEEE, 2022.

[4] A. Dhamija, M. Gunther, J. Ventura, and T. Boult. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[5] R. Sinha, A. Sharma, S. Banerjee, T. Lew, R. Luo, S. M. Richards, Y. Sun, E. Schmerling, and M. Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.

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

[7] H. Zhang, A. Li, J. Guo, and Y. Guo. Hybrid models for open set recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 102–117. Springer, 2020.

[8] R. Li, C. Zhang, H. Zhou, C. Shi, and Y. Luo. Out-of-distribution identification: Let detector tell which i am not sure. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 638–654. Springer, 2022.

[9] R. Mackowiak, L. Ardizzone, U. Kothe, and C. Rother. Generative classifiers as a basis for trustworthy image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2971–2981, 2021.

[10] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[11] R. Cornish, A. Caterini, G. Deligiannidis, and A. Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pages 2133–2143. PMLR, 2020.

[12] H. Wu, J. Köhler, and F. Noé. Stochastic normalizing flows. *Advances in Neural Information Processing Systems*, 33:5933–5944, 2020.

[13] V. Stimper, B. Schölkopf, and J. M. Hernández-Lobato. Resampling base distributions of normalizing flows. In *International Conference on Artificial Intelligence and Statistics*, pages 4915–4936. PMLR, 2022.

[14] J. Behrmann, P. Vicol, K.-C. Wang, R. Grosse, and J.-H. Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, 2021.

[15] P. Hagemann and S. Neumayer. Stabilizing invertible neural networks using mixture models. *Inverse Problems*, 37(8):085002, 2021.

[16] V. Runde. *A taste of topology/*. Springer,, New Delhi:, 2005. Includes bibliographical references and index.

[17] P. Jaini, I. Kobyzev, Y. Yu, and M. Brubaker. Tails of lipschitz triangular flows. In *International Conference on Machine Learning*, pages 4673–4681. PMLR, 2020.

[18] L. Ardizzone, R. Mackowiak, C. Rother, and U. Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. *Advances in Neural Information Processing Systems*, 33:7828–7840, 2020.

[19] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[22] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. URL http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWWZ10.

[23] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2014. URL http://arxiv.org/abs/1405.0312. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

[24] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li. Mitigating neural network overconfidence with logit normalization. *International Conference on Machine Learning (ICML)*, 2022.

[25] D. Miller, N. Sünderhauf, M. Milford, and F. Dayoub. Uncertainty for identifying open-set errors in visual object detection. *IEEE Robotics and Automation Letters*, 7(1):215–222, 2021.

[26] V. I. Bogachev and M. A. S. Ruas. *Measure theory*, volume 1. Springer, 2007.

[27] E. Fetaya, J.-H. Jacobsen, W. Grathwohl, and R. Zemel. Understanding the limitations of conditional generative models. *arXiv preprint arXiv:1906.01171*, 2019.

[28] M. Bauer and A. Mnih. Resampled priors for variational autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 66–75. PMLR, 2019.

[29] I. Kobyzev, S. J. Prince, and M. A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 3964–3979, 2020.

[30] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[31] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

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

[33] B. Charpentier, O. Borchert, D. Zügner, S. Geisler, and S. Günnemann. Natural posterior network: Deep bayesian uncertainty for exponential family distributions. *arXiv preprint arXiv:2105.04471*, 2021.

[34] J. Postels, H. Blum, Y. Strümpler, C. Cadena, R. Siegwart, L. Van Gool, and F. Tombari. The hidden uncertainty in a neural networks activations. *arXiv preprint arXiv:2012.03082*, 2020.

[35] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[36] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018.

[37] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

[38] R. T. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

[39] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[40] J. Postels, M. Liu, R. Spezialetti, L. Van Gool, and F. Tombari. Go with the flows: Mixtures of normalizing flows for point cloud generation and reconstruction. In *2021 International Conference on 3D Vision (3DV)*, pages 1249–1258. IEEE, 2021.

[41] L. Dinh, J. Sohl-Dickstein, H. Larochelle, and R. Pascanu. A rad approach to deep mixture models. *arXiv preprint arXiv:1903.07714*, 2019.

[42] D. Nielsen, P. Jaini, E. Hoogeboom, O. Winther, and M. Welling. Survae flows: Surjections to bridge the gap between vaes and flows. *Advances in Neural Information Processing Systems*, 33:12685–12696, 2020.

[43] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[44] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019.

[45] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*, 2019.

[46] D. Jiang, S. Sun, and Y. Yu. Revisiting flow generative models for out-of-distribution detection. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=6y2KBh-0Fd9.

[47] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, pages 4723–4732. PMLR, 2019.

[48] S. Cao and Z. Zhang. Deep hybrid models for out-of-distribution detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4723–4733, 2022. doi:10.1109/CVPR52688.2022.00469.

[49] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018.

[50] J. Lee, M. Humt, J. Feng, and R. Triebel. Estimating model uncertainty of neural networks in sparse information form. In *International Conference on Machine Learning*, pages 5702–5713. PMLR, 2020.

[51] A. Harakeh, M. Smart, and S. L. Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–93. IEEE, 2020.

[52] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[53] X. Du, G. Gozum, Y. Ming, and Y. Li. SIREN: Shaping representations for detecting out-of-distribution objects. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=8E8tgnYlmN.

[54] N. Kumar, S. Šegvić, A. Eslami, and S. Gumhold. Normalizing flow based feature synthesis for outlier-aware object detection. *arXiv preprint arXiv:2302.07106*, 2023.

[55] X. Du, X. Wang, G. Gozum, and Y. Li. Unknown-aware object detection: Learning what you don't know from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13678–13688, 2022.

[56] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. W. Knauer, K. H. Strobl, M. Humt, and R. Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023.