Supplementary Materials

This Supplementary material elaborates on the Residual flow algorithm and provides additional experiments.

1. Comparison: Proposed approach vs. LDA (Mahalanobis) and GDA models

In this section we examine the performance of our approach compared with LDA (Mahalanobis) and GDA models. In GDA, feature activations of neural networks are modeled using Gaussian discriminant analysis, i.e. posterior of a Gaussian distribution with different mean and different covariance matrix for each class. Calculating the log-likelihood of this model is equivalent to measuring the Mahalanobis distance using a different covariance matrix for each class and adding to it the log-determinant of the class's precision matrix[10]. As in Section 3.2.1, if the feature vector is degenerate, we restrict our attention to its corresponding non-degenerate sub-vector. In LDA (Mahalanobis), the feature activations are modeled using linear discriminant analysis, i.e. posterior of a Gaussian distribution with different mean but with an identical covariance matrix for all classes. We compare these models without employing input-preprocessing. Figure 4 compares the performance of Residual Flow against LDA and GDA for the task of OOD detection. The models use ResNet trained on CIFAR-100 (in-distribution) and tested on various OOD datasets. The Figure shows that our method consistently improves upon the state-of-the-art (LDA model). Note that GDA may produce inferior results in some cases. Figures 5 and 6 show the AUROC comparison on various in- and out-of-distribution datasets of DenseNet and ResNet, respectively. The Figures affirm the observation that modeling feature activations with GDA can deteriorate performance in some cases, especially when the number of per-class training examples is limited - as in the case of CIFAR-100 (Figure 6(c)). Estimating the empirical covariance matrix for each class (GDA) suffers from high variance, exacerbated in scenarios of a small training set. By learning the residual from the LDA model, our method overcomes this limitation, resulting in consistently superior performance over stat-of-the-art.

2. Alternative Architecture

Composing a non-linear flow with linear flow blocks can be done in multiple ways. In this section, we describe an alternative residual flow architecture to the one presented in the main paper, and show that it obtains similar performance. The architecture comprises residual blocks, each composed of a single linear and several non-linear blocks. This architecture is more involved compared to the architecture in the main text, which comprises one linear flow block. We start by defining a linear flow block f_i^{lin}:

$$x_1 = z_1, \quad x_2 = x_2 \circ \exp(s_i) + t_i^T x_1,$$

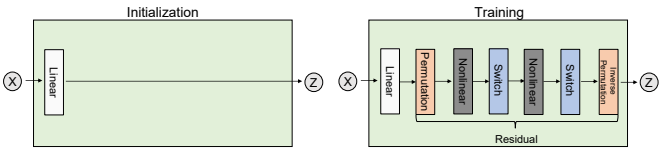
where $s_i \in \mathbb{R}^{d/2}$, $t_i \in \mathbb{R}^{d/2 \times d/2}$, and $\circ$ denotes element-wise multiplication. Here s_i and t_i are scale and translation parameters. The scale parameters are crucial here, as without them, the Jacobian determinant is a constant 1 by definition [11], making the transformation volume preserving, and limiting the expressivity of the model. Next, we compose a residual flow block f_i^{res}:

$$f_i^{res} = f_i^{lin} \cdot p_i \cdot f_{i,1}^{non-lin} \cdot r \cdot f_{i,2}^{non-lin} \cdot r \cdot p_i^{-1},$$

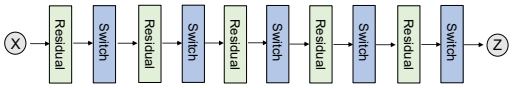
where the linear flow block f_i^{lin} was defined above, r is a switch permutation, p_i is a permutation matrix and p_i^{-1} is it's inverse, and $f_{i,1}^{non-lin}, f_{i,2}^{non-lin}$ are non-linear blocks as described in Eq. (2) in the main paper. We then compose a residual flow model as:

$$f_{res} = f_1^{res} \cdot r \cdot f_2^{res} \ldots r \cdot f_k^{res}.$$

Note that, from Eq. (2) in the main paper, when $s_i(\cdot) = 0$ and $t_i(\cdot) = 0$, the non-linear terms $f_i^{non-lin}$ are just the identity, the permutation terms cancel each other, and in that case the residual flow f^{res} is equivalent to the linear flow f^{lin}. Thus, we pre-train the residual flow by fixing the networks $s_i(\cdot)$ and $t_i(\cdot)$ to be zero, which is equivalent to fitting a Gaussian distribution model to our data[11]. In practice, setting only the last layer of the networks for $s_i(\cdot)$ and $t_i(\cdot)$ to zero is enough, and we found this to perform better in fine tuning the non-linear terms, as most of the network is not initialized to zero. Then, we fine tune the non-linear



(a) Residual Flow blocks in initialization and training.

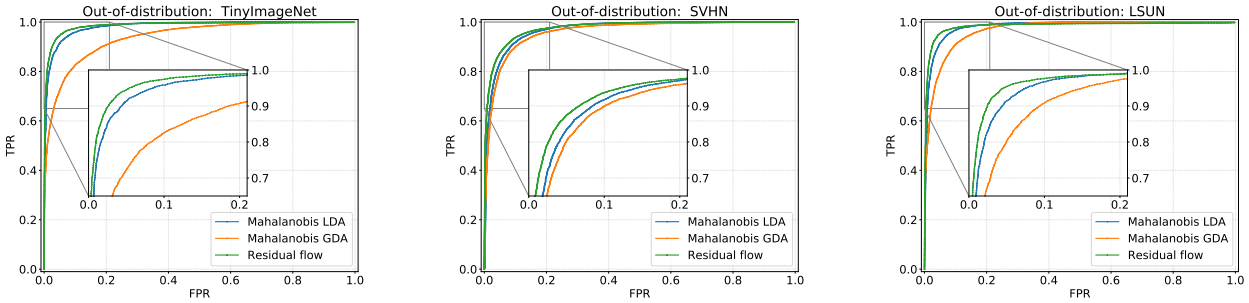


(b) The complete Residual Flow architecture $Z = f(X)$.

Figure 3. Residual Flow alternative architecture.

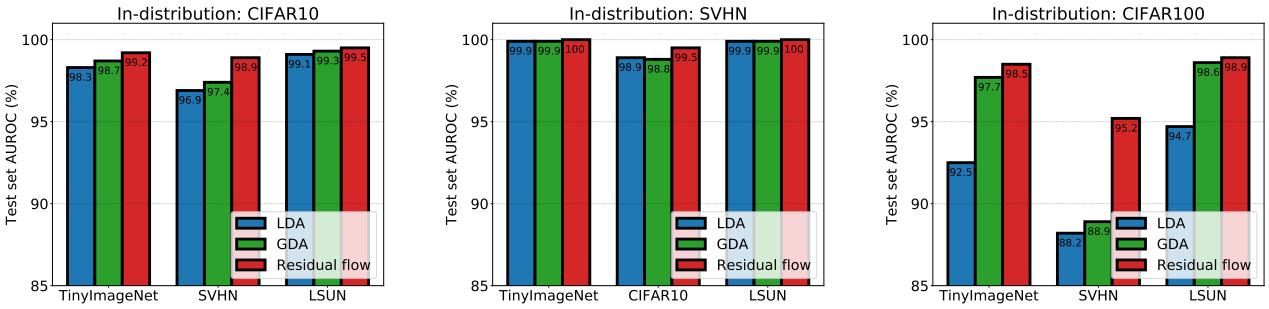
[10] We also compare our method to a GDA variant, which uses the per-class covariance matrix without the contribution of the log determinant of the precision matrix. The results are similar to those of the full GDA model, shown in Figures 5 and 6.

[11] The stopping condition for this stage is when the Kullback–Leibler divergence measure between the linear flow $\hat{p}_X$ and the Gaussian distribution calculated using the empirical covariance $\tilde{p}_X$ meets the criteria: $\mathcal{D}_{KL}(\hat{p}_X || \tilde{p}_X) < 10^{-4}$.



(a) (b) (c)

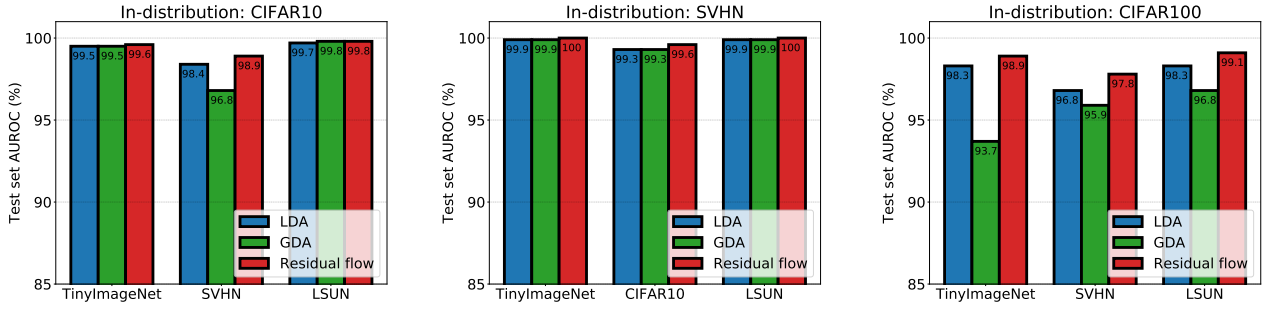
Figure 4. Receiver operating characteristic (ROC) curve comparison of our method, Mahalanobis (LDA) and GDA for the task of OOD detection. The target network is ResNet trained on CIFAR-100. We compare the three models using the following out-of-distribution datasets: (a) TinyImageNet, (b) SVHN and (c) LSUN. The x-axis and y-axis of the figures represent the false positive rate (FPR) and true positive rate (TPR), respectively.



(a) (b) (c)

Figure 5. Area under the receiver operating characteristic (AUROC) (%) curve comparison using DenseNet with 100 layers as a target network. We compare our results with LDA and GDA models across different in- and out-of-distribution datasets. The in-distribution datasets are: (a) CIFAR-10, (b) SVHN and (c) CIFAR-100, and the OOD datasets are presented on the x-axis of the figures.

components of the model to obtain a better fit to the data. Figures 3 illustrates the alternative architecture. This architecture achieves similar results to that proposed in the main paper (see Tables 3 and 4 for full comparison), but with the extra time overhead of training the linear flow. Hence, we chose to include the simpler architecture in the main paper.



(a) (b) (c)

Figure 6. Area under the receiver operating characteristic (AUROC) (%) curve comparison using ResNet with 34 layers as a target network. We compare our results with LDA and GDA models across different in- and out-of-distribution datasets. The in-distribution datasets are: (a) CIFAR-10, (b) SVHN and (c) CIFAR-100, and the OOD datasets are presented on the x-axis of the figures.

In-dist (model)	Out-of-dist	TNR at TPR 95%	AUROC	Detection accuracy	AUPR in	AUPR out
		Mahalanobis [27]/ Res-Flow without pre-processing / Res-Flow with pre-processing				
CIFAR-10 (DenseNet)	SVHN	88.4 / 92.7 / **94.4**	96.8 / 98.5 / **98.8**	92.4 / 94.0 / **94.8**	98.7 / 99.4 / **99.5**	90.4 / 96.6 / **97.6**
	ImageNet	95.4 / **97.3** / **97.3**	98.8 / **99.3** / **99.3**	95.3 / **96.3** / **96.3**	98.9 / **99.3** / **99.3**	98.7 / **99.3** / **99.3**
	LSUN	97.3 / **98.4** / **98.4**	99.0 / **99.6** / **99.6**	96.2 / **97.4** / **97.4**	99.1 / **99.5** / **99.5**	98.8 / **99.6** / **99.6**
CIFAR-100 (DenseNet)	SVHN	84.1 / 68.0 / **87.1**	96.2 / 92.8 / **96.8**	91.0 / 85.3 / **91.1**	98.6 / 96.6 / **98.6**	89.2 / 85.5 / **94.4**
	TinyImageNet	77.5 / 93.1 / **93.4**	95.4 / **98.5** / **98.5**	89.2 / 94.1 / **94.3**	95.8 / **98.4** / **98.4**	93.8 / **98.5** / **98.5**
	LSUN	69.4 / **95.3** / **95.3**	94.6 / **98.8** / **98.8**	89.2 / **95.4** / **95.4**	95.3 / **98.5** / **98.5**	92.7/ **98.9** / **98.9**
SVHN (DenseNet)	CIFAR-10	95.8 / 96.9 / **97.5**	98.8 / 99.2 / **99.3**	95.8 / 96.7 / **97.0**	95.4 / 96.9 / **97.4**	99.6 / 99.7 / **99.8**
	TinyImageNet	99.6 / **99.8** / **99.8**	99.9 / **99.9** / **99.9**	98.9 / **99.2** / **99.2**	99.6 / **99.8** / **99.8**	100.0 / **100.0** / **100.0**
	LSUN	99.7 / **99.8** / **99.8**	99.9 / **100.0** / **100.0**	99.3 / **99.5** / **99.5**	99.7 / **99.9** / **99.9**	100.0 / **100.0** / **100.0**
CIFAR-10 (ResNet)	SVHN	96.2 / 91.7 / **96.5**	99.1 / 98.3 / **99.2**	95.8 / 93.5 / **95.9**	99.6 / 99.3 / **99.7**	98.3/ 96.4 / **98.3**
	TinyImageNet	97.4 / **98.9** / 98.3	99.5 / **99.8** / 99.6	96.3 / **97.6** / 97.1	99.5 / **99.7** / 99.6	99.5 / **99.7** / 99.6
	LSUN	98.7 / **99.3** / 99.1	99.7 / **99.8** /**99.8**	97.5 / 97.8 / **97.9**	99.7 / **99.8** / **99.8**	99.7 / **99.8** / **99.8**
CIFAR-100 (ResNet)	SVHN	92.4 / 83.4 / **94.0**	98.2 / 96.5 / **98.5**	93.8 / 90.3 / **94.6**	99.2 / 98.6 / **99.3**	96.2 / 92.7 / **97.2**
	TinyImageNet	89.4 / **95.0** / **95.0**	97.9 / 98.9 / **99.9**	92.7 / **95.0** / **95.0**	97.9 / **98.9** / **98.9**	97.9 / **98.8** / **98.8**
	LSUN	92.8 / **96.2** / **96.2**	98.3 / **99.2** / 99.1	93.9 / **95.6** / **95.6**	97.9 / **99.0** / **99.0**	98.5 / **99.2** / **99.2**
SVHN (ResNet)	CIFAR-10	97.6 / **98.6** / 98.5	99.3 / **99.6** / **99.6**	96.9 / **97.8** / 97.7	97.3 / **98.2** / 98.1	99.7 / **99.9** / **99.9**
	TinyImageNet	99.7 / **99.8** / **99.8**	99.8 / **99.9** / **99.9**	99.1 / **99.4** / **99.4**	99.5 / **99.7** / **99.7**	99.9 / **100.0** / **100.0**
	LSUN	99.8 / **99.9** / **99.9**	99.9 / **100.0** / **100.0**	99.6 / **99.7** / **99.7**	99.6 / **99.7** / **99.7**	99.9 / **100.0** / **100.0**

Table 3. A comparison between residual flow implemented using the architecture described in Section 2 and Mahalanobis [27] on the task of out-of-distribution detection for image classification of various in- and out-of-distribution data sets. The hyper-parameters were tuned using a validation set of in- and out-of-distribution datasets. The values presented here are percentages and the best results are indicated in bold.

In-dist (model)	Out-of-dist	TNR at TPR 95%	AUROC	Detection accuracy	AUPR in	AUPR out
		Mahalanobis [27]/ Res-Flow without pre-processing / Res-Flow with pre-processing				
CIFAR-10 (DenseNet)	SVHN	89.6 / 75.6 / **91.7**	97.6 / 94.9 / **98.0**	92.6 / 87.8 / **93.4**	94.5 / 88.7 / **96.2**	99.0 / 97.9 / **99.1**
	TinyImageNet	94.9 / **97.3** / **97.3**	98.8 / **99.3** / **99.3**	95.0 / **96.4** / **96.4**	98.7 / **99.4** / **99.4**	98.8 / **99.3** / **99.3**
	LSUN	97.2 / **98.4** / **98.4**	99.2 / **99.6** / **99.6**	96.2 / **97.4** / **97.4**	99.3 / **99.6** / **99.6**	99.2 / **99.6** / **99.6**
CIFAR-100 (DenseNet)	SVHN	62.2 / 65.4 / **86.3**	91.8 / 91.7 / **96.4**	84.6 / 84.2 / **90.7**	82.6 / 83.9 / **94.0**	95.8 / 96.0 / **98.3**
	TinyImageNet	87.2 / **92.4** / 91.2	97.0 / **98.3** / 98.1	91.8 / **93.7** / 93.4	96.2 / **98.2** / 98.1	97.1 / **98.3** / 98.2
	LSUN	91.4 / 95.1 / **95.3**	97.9 / 98.7 / **98.8**	93.8 / 95.1 / **95.3**	98.1 / 98.5 / **98.6**	97.6 / **98.9** / **98.9**
SVHN (DenseNet)	CIFAR-10	97.5 / 96.2 / **96.5**	98.8 / 98.9 / **99.1**	96.3 / 96.1 / **96.3**	99.6 /**99.7** / **99.7**	95.1 / 96.0 / **96.5**
	TinyImageNet	99.9/ 99.7 / **99.9**	99.8 / 99.9 / **99.9**	98.9 / **99.1** / 99.0	99.9 / 99.8 / **99.9**	99.5 / **100.0** / 99.6
	LSUN	100.0 / 99.8 / **100.0**	99.9 / **99.9** / **99.9**	99.2 / **99.4** / 99.3	99.9 / 99.8 / **100.0**	99.6 / **99.9** / 99.7
CIFAR-10 (ResNet)	SVHN	75.8 /76.0 / **95.7**	95.5 / 94.2 / **98.9**	89.1 / 87.1 / **95.6**	91.0 / 97.4 / **99.4**	98.0 / 89.3 / **98.0**
	TinyIageNet	95.5 / **98.8**/ 98.5	99.0 / **99.7** / 99.6	95.4 / **97.4** / 97.1	98.6 / **99.7** / 99.6	99.1 / **99.7** / 99.6
	LSUN	98.1 / 99.5 / **99.6**	99.5 / 99.8 / **99.9**	97.2 / 98.2 / **98.5**	99.5 /**99.8** / **99.8**	99.5 / 99.8 / **99.9**
CIFAR-100 (ResNet)	SVHN	41.9 / 59.1 / **66.8**	84.4 / 90.6 / **92.4**	76.5 / 82.6 / **84.9**	69.1 / 81.0 / **83.3**	92.7 / 95.8 / **96.8**
	TinyImageNet	70.3 / 73.9 / **77.3**	87.9 / 88.8 / **89.6**	84.6 / 84.5 / **86.7**	76.8 / 78.8 / **79.2**	90.7 / 88.8 / **92.5**
	LSUN	56.6 / 66.1 / **68.1**	82.3 / **89.1** / 86.5	79.7 / **85.6** / 83.4	70.3 / **79.1** / 75.8	85.3 / 89.2 / **89.7**
SVHN (ResNet)	CIFAR-10	94.1 / **98.4** / 97.6	97.6 / **99.5** / 99.2	94.6 / **97.5** / 96.4	98.1 / **99.9** / 99.7	94.7 / **97.9** / 97.3
	TinyImageNet	99.2 / **99.9** / **99.9**	99.3 / **99.9** / **99.9**	98.8 / **99.5** / **99.5**	98.8 / **99.7** / **99.9**	98.3 / **100.0** / 99.6
	LSUN	99.9 / 99.9 / **100.0**	99.9 / **100.0** / 99.9	99.5 / **99.7** / 99.6	99.9 / 99.7 / **99.9**	98.8 / **100.0** / **100.0**

Table 4. A comparison between residual flow implemented using the architecture described in Section 2 and Mahalanobis [27] on the task of out-of-distribution detection for image classification of various in- and out-of-distribution data sets. The hyper-parameters were tuned using strictly in-distribution and adversarial (FGSM) samples. The values presented here are percentages and the best results are indicated in bold.