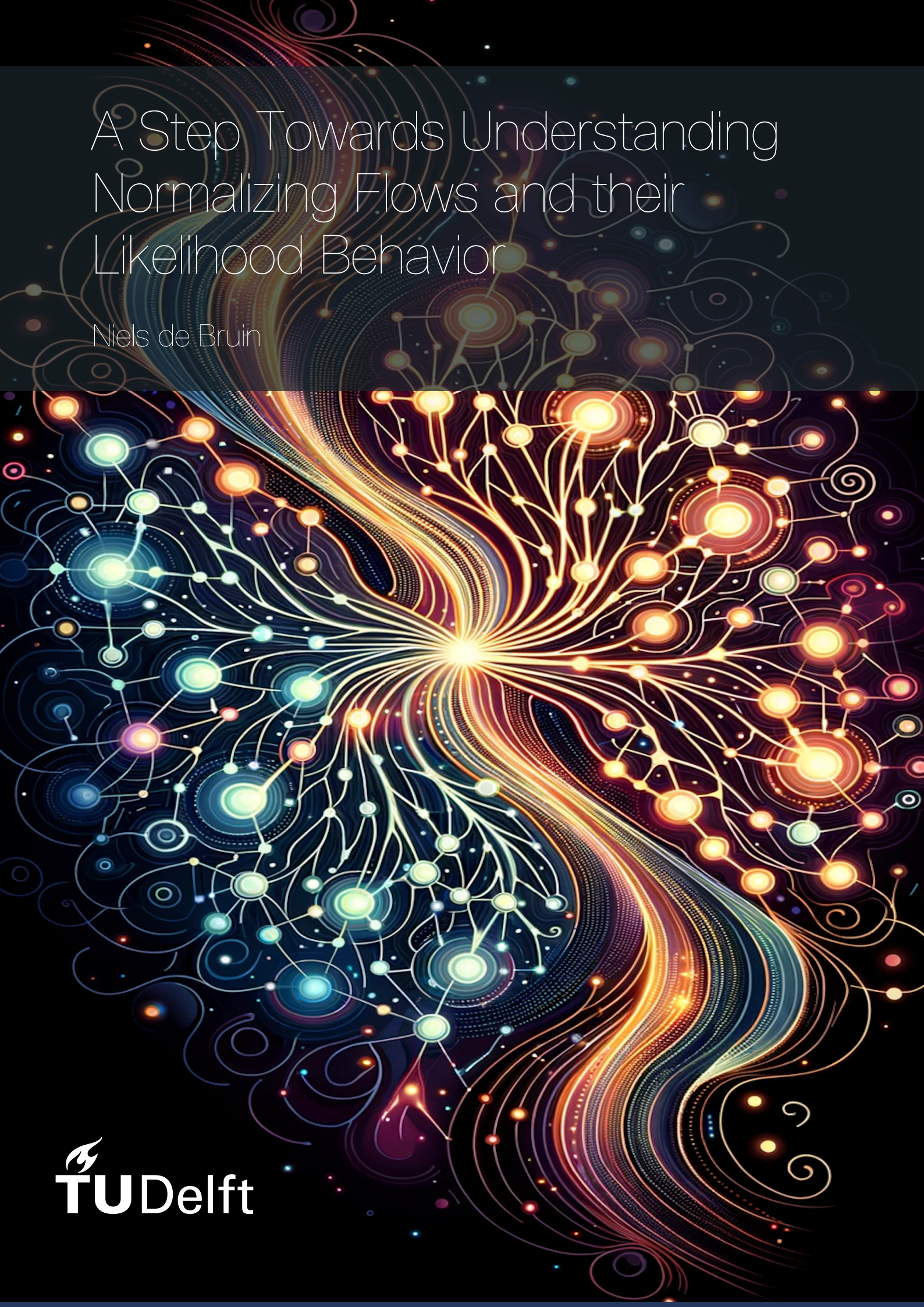
A Step Towards Understanding Normalizing Flows and their Likelihood Behavior

Niels de Bruin



TU Delft

A Step Towards Understanding Normalizing Flows and their Likelihood Behavior

THESIS REPORT

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Niels de Bruin

to be defended publicly on February 16, 2024 at 9:30

Thesis Committee:

Chair:	Dr. J. van Gemert, Faculty EEMCS, TU Delft
University supervisor:	Prof. dr. M. Loog, iCIS, Radboud University
Committee Member:	Dr. ir. Bidarra, Faculty EEMCS, TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Pattern Recognition and Bioinformatics
Department of Intelligent Systems
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Preface

There are many different flavors of artificial intelligence (AI), but none has raised the awareness of both capabilities and potential risks as much as generative AI. This has made the importance of in-depth understanding of these models more important. Unfortunately, the latter has not kept up with the rate at which the capabilities of these models developed. This thesis report details my research focused on normalizing flows, a deep generative model, with a particular emphasis on understanding its likelihood behavior. The overlying theme in this research was simplification. I designed a set of controlled test cases to explore model behavior and extrapolated how these results might help explain the normalizing flow behavior of real-world datasets. The thesis paper contains all the necessary background information for understanding this work. For interested readers, a supplementary appendix outlining some of the more complex architecture extensions of the normalizing flow model.

This thesis was conducted within the pattern recognition group of the EEMCS faculty under the supervision of Prof. dr. Marco Loog. I want to thank all members of the group for allowing me to join the weekly discussions. I am particularly thankful to Marco for all the guidance and support during the somewhat extended period of this thesis. During this thesis, there were many ups and downs. I want to thank my friends for their support, with a special mention to Shaad, Amey, Sharayu, Henrique, Alexandra, Bing, Kian, Henry, Garazi, Aniket, Max, Wessel, Anna, and most importantly, I wish to thank my parents for unwavering support during my whole of my education.

Finally, I want to acknowledge my graduation committee consisting of Dr. J. van Gemert, Prof. dr. M. Loog, and Dr. ir. R. Bidarra.

Niels de Bruin
Delft, Februari 2024

Contents

Part I

Thesis Paper

A Step Towards Understanding Normalizing Flows and their Likelihood Behavior

Niels de Bruin
Delft University of Technology

February 8, 2024

Abstract

Normalizing flows have demonstrated their ability to learn complex and high-dimensional distributions. However, the behavior of normalizing flow likelihoods are not yet fully understood, particularly when exposed to outlier data, where it has been observed that large likelihoods are often assigned to inputs that are substantially different from the training set. To better understand the likelihood behavior and outlier detection capabilities of normalizing flows, we analyze a more restricted version of the model using synthetic test data from parametric distributions, allowing access to the density of the underlying distribution.

1 Introduction

Normalizing flows are a class of generative models that provide both efficient inference, generation, and exact estimation of the log-likelihood [1], which should make them a promising candidate for density-based out-of-distribution (OOD) detection. However, it has been shown that normalizing flows often assign disproportionate log-likelihoods to inputs substantially different from the data they have been trained on, raising questions about the robustness of these models [2].

Various hypotheses have been proposed to explain the occurrence of this phenomenon, especially in the realm of image data [3]. However, we found that existing research primarily concentrates on empirically demonstrating the presence of this phenomenon in complex high-dimensional data or developing new components tailored to compel normalizing flows to identify outliers [4, 5], often at the expense of generalization rather than examining the underlying mechanics. The absence of such an in-depth analysis is not entirely unexpected, as normalizing flows, even for a deep-learning model, are particularly challenging to analyze for a number of reasons. We argue the two most important of these are: Firstly, model complexity, to solve increasingly challenging tasks on high-dimensional datasets, models keep growing in both size and architecture complexity. For normalizing flows, which, due to their bijective nature, are especially resource-intensive in high dimensional space, are often augmented with more complex architecture components, such as RealNVP's multi-scale method [6, 1], Flow++'s variational dequantization of the inputs [7], GLOW's invertible 1x1 convolutions and actnorm [8]. Hence, if unexpected and possibly undesirable behavior does arise, explaining and assigning it to a specific attribute or component of the model is extremely challenging.

Secondly, when training normalizing flows, which typically employs unsupervised learning, there's often no access to the ground truth density function. This lack of a concrete reference point complicates assessing how accurately these models capture the data distribution. Unlike supervised learning, where the ground truth is available, predictions can often be evaluated using bounded metrics such as accuracy, precision, and recall, which provide a more straightforward interpretation of model performance. The evaluation of normalizing flows often relies on differential cross-entropy, an unbounded probability measure highly dependent on the underlying distribution's descriptive statistics, which means that making quantitative statements about its magnitude is often challenging without a point of reference provided by the true density function. This introduces numerous problems. For example, learning curves can still be used to estimate whether the model has stopped converging. Yet, we cannot say if this results from model misspecification or if the model has reached a reasonable approximation. Consequently, rather than directly comparing their outputs to a known true density, evaluating their

effectiveness often relies on indirect evaluation methods, such as analyzing generated samples' quality or utility in unsupervised downstream tasks. This approach, however, becomes convoluted for data types that do not lend themselves to intuitive human interpretation.

When computational limitations make an in-depth model analysis challenging, a lot can still be learned through straightforward simplification. For instance, if the model behavior of interest can be reproduced in a simplified and controlled setting, it could indicate that this behavior might not be (fully) attributed to, e.g., a high-dimensional dataset or a complex architecture but could instead be an inductive bias of the model being tested.

To further bridge the gap in understanding between the theoretical promise of normalizing flows to provide exact densities and the less-understood empirical observations from complex real-world scenarios. This work explores the behavior of the core normalizing flow model using artificially constructed test distributions. These test distributions are specifically designed to evaluate the model's likelihood behavior in response to statistical variations from the training data. Our primary interest lies in observing and understanding the trends in model behavior across these different test sets rather than focusing on exact values. This approach aims to explore how these behavioral trends might contribute to less understood aspects of model likelihood behavior, as highlighted in previous studies [2, 3]. Our experimental setup, concentrating on the core elements of the normalizing flow model, utilizes Gaussian distributions with known parameterizations for testing. The well-defined nature of these distributions allows for the systematic construction of test scenarios with varying degrees of typicality, and provides access to their density functions, facilitating a more nuanced and precise analysis. This enables a broad, comparative analysis, focusing on general behavioral patterns rather than precise numerical outputs, using the distributions' density functions to examine the model's likelihood behavior in diverse testing environments.

Our experiments are organized around two principal themes: the generalization and robustness of the model. These are underpinned by our objective to replicate and thereby gain a better understanding of the atypical behaviors noted in prior research.

The generalization experiments are designed to answer how well the model can adapt to test distributions that maintain key statistical properties, especially correlations, or undergo only minor perturbations to these features. Furthermore, we generate a series of affine variants of these test distributions, varying in scale and location. This methodology facilitates a systematic evaluation by targeting specific regions of the data distribution. It also enables us to observe the general trends in the model's response to variations in factors like location or scale, thereby providing a deeper insight into its generalization capabilities.

The robustness experiments are designed to assess the model's likelihood behavior with test distributions that significantly diverge from the training set's properties, like correlations, location, and scale, to the extent that they are considered full outlier distributions. Similar to our generalization experiments, we create a set of affine variants of these test distributions. In this simplified and controlled setting, where we have access to a ground truth density, our objective is to closely examine the model's response patterns. Previous research has shown that normalizing flows can assign disproportionately high likelihoods to outlier distributions. Our goal is to determine the occurrence and the specific conditions and scenarios under which this happens, as well as its implications for the model's effectiveness in distinguishing outlier distributions, which is crucial for its capability in out-of-distribution (OOD) detection.

This work offers a detailed and systematic analysis of normalizing flows through the use of nearly 300 distinct test distributions, providing a clearer picture of these models' inherent properties and behaviors. We demonstrate that the abnormal likelihoods observed in previous work can be replicated in a simplified and controlled environment using only the core of the normalizing flow model. This suggests that such behaviors are not solely attributable to architectural or data complexity but likely originate from an inductive prior inherent to the normalizing flow model. Furthermore, our wide range of test distributions facilitated the identification of particular characteristics of a test distribution that significantly impair the performance of normalizing flows. Moreover, we illustrate an intriguing inverse relation: as the divergence between training and test distributions increases, the model increasingly tends to disregard the distinctive features of the test distribution, instead favoring likelihoods akin to those of the training distribution. In our final analysis, we illustrate the consequential impact on OOD detection performance in earlier identified scenarios. Moreover, by extrapolating on our combined results, we hypothesize why normalizing flows may have previously underperformed in OOD detection, particularly on image datasets. Overall, this

work sheds light on past challenges and may guide future research in normalizing flows.

The remainder of this work is organized as follows: section 2 provides the necessary background foundational material, such as an explanation of the core normalizing flow model and relevant concepts from information theory. section 3 will further formalize the definitions of generalization and robustness used in this work, lay out the experimental design, and elaborate on the evaluation methods used. section 4 provides a baseline evaluation of the model. section 5 covers the results of our experiments on the model's generalization performance. section 7 discusses the results of evaluating model robustness. section 8 briefly evaluates the outlier performance for our different experiments. Finally, section 9 offers a combined discussion and conclusion of the main results.

2 Preliminaries

This section aims to provide a brief overview of the theory essential for a basic understanding of the normalizing flow model.

2.1 Essence of Normalizing Flows

Arbitrarily complex distributions can be constructed by transforming a simper base distribution $p_Z(z)$ into a more complex target distribution $p_X(x)$ through an invertible mapping $f : X \to Z$. We can express the density function of X in terms of Z by applying the change of variable theorem. Given continuous random variables Z with density function $p_Z(z)$, and let $f : \mathbb{R} \to \mathbb{R}$ be a bijective monotonic function such that $x = f^{-1}(z)$ and its inverse is equal to $z = f(x)$. Then the random variable X that results from the mapping $f^{-1}(Z) \sim p_Z$ will have a density function given by:

$$p_X(x) = p_Z\left(f_\theta(x)\right) \left| \frac{d}{dx} f_\theta(x) \right| \ . \tag{1}$$

When used with vectors, the last term of Equation 1 is replaced with the determinant of the Jacobian matrix, resulting in:

$$p_X(x) = p_Z\left(f_\theta(x)\right) \left|\det\left(\nabla f_\theta(x)\right)\right| \ . \tag{2}$$

Normalizing flows are designed to learn the parameterization θ for mapping $f_\theta(x)$; it does so by decomposing $f_\theta(x)$ into a sequence of affine transformations $f = f_1 \dots f_k$ which are referred to as *coupling layers*. Densities *flow* through each coupling layer undergoing re-normalization to a valid density function following each application through the use of the change of variables Apart from being invertible, there are no restrictions on $f_\theta(x)$. However, the normalization of the densities after each subsequent application of $f^i_\theta(x)$ using Equation 2 requires computing the Jacobian determinant $|\det\left(\nabla f_\theta(x)\right)|$, often an intractable computation that becomes infeasible for high dimensional data. Normalizing flows employ affine coupling layers to solve this problem by constructing each transformation as a triangular map, which results in a triangular Jacobian matrix whose determinant can be computed in linear time, as the determinant of a triangular matrix is merely the product of its diagonal elements. By utilizing this property, the computation becomes feasible for high-dimensional data.

2.1.1 Construction of the affine coupling layers

To ensure a triangular map, each affine coupling layer produces a transformation conditioned on only a part of the input. More formally, given a D dimensional input and output vector x and y respectively, a dimension index $d \in \mathcal{Z}^+$ such that $d < D$. The affine coupling layer as per Equation 3, will *copy* the first d dimensions of the input without modification, thus $y_{1:d} = x_{1:d}$. The remaining $\{d-1 : D\}$ dimensions are scaled $\exp\left(s_\theta\left(x_{1:d}\right)\right)$ and translated $t_\theta(x_{1:d})$ by with the Hadamard product of the transformation conditioned $\{1 : d\}$. The functions $\exp\left(s_\theta\left(x_{1:d}\right)\right)$ and $t_\theta(x_{1:d})$ can be parameterized with any complex function, such as a deep neural network.

$$f^i_\theta(x) = \begin{cases} y_{1:d} & = x_{1:d} \\ y_{d+1:D} & = x_{d+1:D} \odot \exp\left(s_\theta\left(x_{1:d}\right)\right) + t_\theta\left(x_{1:d}\right) \end{cases} \tag{3}$$

$$f^i_\theta(y)^{-1} = \begin{cases} x_{1:d} & = y_{1:d} \\ x_{d+1:D} & = \left(y_{d+1:D} - t_\theta\left(y_{1:d}\right)\right) \odot \exp\left(-s_\theta\left(y_{1:d}\right)\right) \end{cases} \tag{4}$$

Since $x_{1:d}$ has not been changed but merely copied into $y_{1:d}$, it can trivially be retrieved. Recovering $x_{d+1:D}$ is analogous to recomputing the original transformation, using the unmodified part of the input and inverting the original linear operations on $y_{d+1:D}$, which can be done straightforwardly using Equation 4.

2.1.2 Computation of the Jacobian

$$\nabla f_\theta^i(x) = \frac{\partial y}{\partial x^T} = \begin{bmatrix} \frac{\partial y_{1:d}}{\partial x_{1:d}^T} & \frac{\partial y_{1:d}}{\partial x_{1+d:D}^T} \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \frac{\partial y_{d+1:D}}{\partial x_{d+1:D}^T} \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} \mathbb{I}^D & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left(s\left(x_{1:d}\right)\right)\right) \end{bmatrix} \tag{6}$$

The Jacobian of the transformation derived from Equation 3 forms a lower triangular matrix. Thanks to this structure, the calculation of the determinant becomes feasible, as it is directly given by the product of the diagonal of the matrix, demonstrated in Equation 6. This adjustment allows for a tractable computation that scales linearly, O(D), with the dimensionality, thereby mitigating the previous computational challenges associated with high dimensional data.

2.1.3 Choice of Prior

The last element of our model is the prior, which is subject to minimal constraints. Primarily, its support must comply with the condition $\text{supp}(p_X(x)) \subseteq \text{supp}(p_Z(f(x)))$. Non-adherence to this requirement could lead to instances where an input x with non-zero density is mapped to an output $z = f(x)$ for which the log likelihood is not defined. However, using a prior with either larger or infinite support is still possible to *approximate* a training distribution with finite support. For normalizing flows, the most common choice is the standard Gaussian distribution $\mathcal{N}(0, \mathbb{I}^D)$. Apart from infinite support, it's computationally efficient for large dimensionality as its density function can be computed in $O(D)$. Additionally, its smoothness and symmetry can be beneficial in stabilizing gradient descent-based optimization.

2.1.4 Generation

The generation process follows a straightforward procedure. Samples are drawn from the prior $z \sim p_Z$. Subsequently, these samples are transformed back into the data space through the application of the inverse mapping to $x = f_\theta(z)^{-1}$.

2.1.5 Core Characteristics

Normalizing flows distinguish them self from other prominent deep generative models like generative adversarial networks (GANs) [9], variational autoencoders (VAEs) [10], and transformers [11] by their unique ability to provide a valid probability density function. This density function, being directly optimizable through a maximum likelihood objective equips normalizing flows with enhanced resilience against overfitting and reduces common training instabilities in generative modeling, such as the mode collapse frequently seen in GANs. However, this strength also introduces a notable limitation: the necessity for f_θ to be bijective limits the model's capacity to compress its input into a lower-dimensional latent space, resulting in computational inefficiencies in high-dimensional spaces.

2.1.6 Normalizing Flows Architecture Extensions

As discussed, this work centers on the core components of the normalizing flow model, an architecture that becomes computationally infeasible as dataset complexity and dimensionality grows. This has led to many extensions and model variants not studied in this work. Hence, an understanding of these extensions is not required for this work. However, for the interested reader, we have compiled a more in-depth introduction to normalizing flows and the most prominent extensions of the model, which can be found in Appendix D.

2.2 Relevant concepts from Information Theory

In information theory, entropy, specifically Shannon's entropy, is a measure of the uncertainty or randomness associated with a random variable [12]. Given a discrete random variable with probability mass function $P(X)$ Shannon's entropy H(X) is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x) = -\mathbb{E}\left[\log P(x)\right] \geq 0 \tag{7}$$

It provides a way to quantify the information contained in a random variable. Higher entropy indicates greater uncertainty or randomness, while lower entropy implies more predictability. Differential entropy is the continuous counterpart of Shannon's discrete entropy, used to measure uncertainty or randomness in continuous probability distributions. It is defined similarly to discrete entropy but adapted for continuous random variables. Given a continuous random variable with probability density function p(x), the differential entropy H(X) is defined as:

$$H(X) = -\int_{x \in \mathcal{X}} p(x) \log p(x)\, dx = -\mathbb{E}\left[\log p(x)\right] \tag{8}$$

Just like with discrete entropy, a higher differential entropy value implies greater uncertainty or randomness in the continuous random variable, while a lower value suggests more predictability. However, it's important to note that, unlike discrete entropy, differential entropy can also be negative as a probability density function can, unlike a probability mass function, be greater than one and doesn't have the same straightforward interpretation as discrete entropy, making its use and interpretation more nuanced.

2.2.1 Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence, also called relative entropy, is a measure of the statistical difference between two distributions [13]. Given probability density functions $p(x)$ and $q(x)$, the KL-divergence between them is defined as

$$D_{KL}(p \mid\mid q) = \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \tag{9}$$

Additionally, the KL-divergence has the useful property of always being non-negative $0 \leq D_{KL}(p \mid\mid q)$ and is equal to 0 if and only if the two distributions p and q are identical. However, it noted that the KL-divergence is not symmetric: $D_{KL}(p \mid\mid q) \neq D_{KL}(q \mid\mid p)$ and therefore is not a metric as it does not conform the triangle inequality. Moreover, it can become infinite if p and q do not share the same support.

2.2.2 Cross Entropy

When training a normalizing flow model $p^\theta(x)$, we optimize the expected log-likelihood $\mathbb{E}_p[\log p^\theta(x)]$ over samples drawn from our target distribution. This optimization is equivalent to minimising the cross-entropy $H(p, p^\theta)$ between the target and model distribution. The cross-entropy can be expressed as the sum of the entropy of the data distribution $H(p)$ and the KL-divergence $D_{KL}(p \mid\mid p^\theta)$ between the data and target distribution.

$$H(p,\ p^\theta) = H(p) + D_{KL}(p \mid\mid p^\theta). \tag{10}$$

3 Methodology and Experiment Design

3.1 Generalization and Robustness

We will conduct experiments to examine the likelihood behaviour of the normalizing flows model in terms of generalization and robustness. As discussed, our simplified method will use synthetic data from Gaussian distributions with known parameterizations and study how the model $p^\theta(x)$ behaves when trained on data generated from $p(x)$. We will evaluate the model using a test set $\mathcal{Q}^N(q_\lambda)$ that is generated from a test distribution $q_\lambda(x)$ with parameterization λ_q.

$$\mathcal{Q}^N(q_\lambda) = \{x_1, x_2, ..., x_N \mid x_i \sim q_\lambda\} \tag{11}$$

Generalization and robustness can be challenging to quantify and share a certain degree of overlap. As their interpretations might depend on the type of data, task (e.g. supervised, unsupervised), or model (e.g. generative, discriminative). In this work, based on the context of our experiments using Gaussians, we will use the following definitions inspired by [14]:

Definition 3.1 (**Generalization**)**.** In this work, the ability to *generalize* is defined by the degree to which model $p^\theta(x)$ can correctly estimate $p(x)$ for test set $\mathcal{Q}^{\mathrm{N}}(q_\lambda)$ containing previously unseen samples that still largely preserve statistical properties such as correlation and mutual information but might be atypical (from low-mass regions or outside the typical set). The assumption is that if $p^\theta(x)$ does generalize well, then the quality of the density estimate should not be affected too much by changes in the parameterization of q_λ relative to p.

Definition 3.2 (**Robustness**)**.** Similar to generalization, the robustness of p^θ is determined by its ability to accurately estimate $\log p(x)$ for $\mathcal{Q}^{\mathrm{N}}(q_\lambda^{\mathrm{out}})$. However, $\mathcal{Q}^{\mathrm{N}}(q_\lambda^{\mathrm{out}})$ is constructed to consist of data that does not preserve the underlying properties of the training distribution, such as correlations and mutual information. Hence, q_λ^{out} can be considered a true outlier distribution.

We will now detail our experimental design based on Definition 3.1 and Definition 3.2. The experiments aim to test generalization by examining how normalizing flows trained on Gaussians with full covariance behave when evaluated on $Q(\lambda_q)$ specific from different regions of the training distribution. While these test sets maintain the underlying correlations of the training distribution, their location and scale may differ. Conversely, when evaluating robustness, we focus on data derived from distributions that lack the underlying properties of the training samples, in other words, an outlier distribution with both different covariance and location. A significant component of the experiments involves the use of affine transformations to construct a variety of test distributions. This allows a straightforward way to evaluate

model behavior in various regions of the training distribution through a combination of scaling and translation. More formally, in the case of evaluating model generalization, let the training distribution $p(x) = \mathcal{N}(\lambda_p)$ be a Gaussian with parameterization $\lambda_p = \{\mu_p, \Sigma_p\}$. Let the test distribution be a Gaussian, $q_\lambda(x) = \mathcal{N}(\lambda_q)$, with parameters λ_q constructed as a mapping of λ_p as follows:

$$\lambda_q(s,t) \rightarrow \{\mu_p + t, s^2\Sigma_p\}. \tag{12}$$

Alternatively, an equivalent definition q_λ is the application of the affine mapping $g(x; s, t)$ to $x \sim p(x)$, represented by:

$$g(x; s, t) \rightarrow (x - \mu_p)s + \mu_p + t \tag{13}$$

In the case of robustness evaluation, the definition $\lambda_q^{\text{out}}(s,t)$ is nearly identical but differs in the fact that it is no construct as a mapping based on λ_p but instead uses $\{\mu_q, \Sigma_q\}$. In both scenarios, a normalizing flow $p^\theta(x)$ is trained on data generated from $p(x)$ and is then evaluated using data generated from $q_\lambda(x)$. Finally, we will assess the model behavior and performance under different conditions, such as varying levels of scaling, translation, and dimensionalities, using various evaluation methods, which will be further elaborated upon in subsequent sections. Our methodology ensures that we always have access to and precise control of the density functions, which provides a straightforward approach to evaluating how the model reacts to different test sets. More importantly, it enables a direct comparison of these estimates with the densities designated by the true density function of the training distribution for different test sets. In short, this controlled approach, combined with a model architecture simplified to only the essence of the normalizing flow model, facilitates a more comprehensive analysis of normalizing flow capabilities and potential limitations in different scenarios.

3.2 Evaluation Methods

This section briefly elaborates on the evaluation method used in this work, building up the explanations in subsection 2.2. For the artificially constructed train and test distributions, the exact value for most evaluation methods can be computed analytically since the parameterization of these distributions is known. For the model, we require a numerical or combination of analytical and numerical estimates of both. Hence, we will elaborate further on the evaluation methods and how they are computed or approximated with sufficient accuracy.

3.2.1 Estimation of Cross-Entropy for different

To compute an estimate of the cross-entropy $H^*(q, p^\theta)$ over the test set $\mathcal{Q}(q_\lambda)$ for different parameterization λ_q we use the fact that the loss over our test distribution $\mathbb{E}_{q_\lambda}\left[\log p^\theta\right]$ and cross-entropy $H(q,p)$ are identical apart from their sign (Equation 15). Hence, we construct the cross entropy estimate for the a finite test set by computing:

$$H(q, p^\theta) = -\mathbb{E}_q\left[\log p^\theta\right] \approx \frac{1}{N}\sum_{x \in \mathcal{X}^N} \log p^\theta(x_i) = H^*(q, p^\theta) \tag{14}$$

Further detailed in section 4, we validated the reliability of $H^*(q, p^\theta)$, by comparing the estimates to the analytical solution for $H(q,p)$, as defined in Equation 10. Figure 2 shows the results for $H(q,p)$, $H^*(q,p)$, $H^*(q,p^\theta)$, and differential entropy $H(q)$ for different choices of s, t and dimensionalities. We can see that $H(q,p)$ overlaps nearly perfectly with its estimate $H^*(q,p)$; thus, we argue that this method is sufficiently accurate to approximate $H^*(q, p^\theta)$.

3.2.2 Estimation of the KL-Divergence

As already discussed in subsubsection 2.2.1, the Kullback–Leibler divergence is a measure of statistical distance between two distributions but not a *distance metric* as it's non-symmetric, does not satisfy the triangle inequality. Moreover, its absolute values can be hard to interpret without context. Additionally, the KL-divergence can be infinite when the support is either disjoint or non-overlapping. In our experiments, the latter does not occur, as only distributions with identical support are used. Regardless of shortcomings, when comparing $q(x)$ with other distributions, a lower KL-divergence means a better approximation of $q(x)$. We will compare the degree of difference in the KL-divergences $D_{KL}(q(x)||p(x))$ and $D_{KL}(q(x)||p^\theta(x))$. The former can be obtained easily through the analytical solution for the KL-divergence between two multivariate Gaussians.

Proposition 1. ([15]) Given two multivariate Gaussians q, p, the Kullback–Leibler divergence between them is given by:

$$D_{KL}(q \parallel p) = \frac{1}{2}[\log\frac{|\Sigma_p|}{|\Sigma_q|} - k + \left(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p\right)^T \Sigma_p^{-1}\left(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p\right) + \operatorname{tr}\left\{\Sigma_p^{-1}\Sigma_q\right\}]$$

While $D_{KL}(q(x) \parallel p(x))$ can be analytically computed using Proposition 1, the KL-divergence between the test-distribution $D_{KL}(q \parallel p^\theta)$ cannot be directly computed as this would require solving a non-tractable integral. Instead, we choose to use estimate $D^*_{KL}(q\|p^\theta)$ constructed with finite sample $\mathcal{X}^N$ of size N. The KL divergence can be written as the difference between the $H(q,p)$ and the $H(q)$ [16]. Since we control the parameterization of q, we can compute $H(q)$ analytically. The negative log-likelihood computed for an i.d.d. sample from q is used to approximate $D^*_{KL}(q \parallel p^\theta)$.

$$\begin{aligned} D_{KL}(q \parallel p^\theta) &= H(q,p^\theta) - H(q) \\ &= H(q,p^\theta) - \frac{1}{2}\log\det(2\pi \mathrm{e}\Sigma_q) \\ &\approx \left(\sum_{x\in\mathcal{X}^N} -\log p^\theta(x_i)\right) - \log\det(2\pi\mathrm{e}\Sigma_q) \\ &= D^*_{KL}(q \parallel p) \end{aligned}$$

To validate the reliability of our method for estimating $D^*_{KL}(q \parallel p^\theta)$, we computed an estimate for $D^*_{KL}(q \parallel p)$ using the same method and compared it with the ground truth obtained by calculating $D_{KL}(q \parallel p)$ analytically using Proposition 1. We found that for all parameterizations and dimensionalities tested, the estimated $D^*_{KL}(q \parallel p)$ was nearly identical to the ground truth $D_{KL}(q \parallel p)$ as shown by Figure 6. Hence, we concluded that this approach would be sufficiently precise for evaluation.

3.2.3 Evaluation of distribution using Kernel Density Estimation

Although KDEs may not be entirely precise, they offer a simple way to assess fit by comparing location, variance, shape, and potential skewness. While we recognize their limitations, we argue they can still yield useful information. For instance, they enable rapid assessment of potential overlap between two distributions (critical for identifying outliers using likelihood bounds). Additionally, we can easily determine if the model is over or underestimating by examining the location of the KDE.

3.2.4 Outlier detection and its relationship to distribution overlap

In the following sections, the concept of distribution *overlap* will be discussed often as it plays an important role in outlier detection. Consider a simple case where there are predefined lower (L) and upper (U) bounds, such that densities falling outside this range are considered outliers. The location and number of bounds are task-specific, but a basic strategy would be to estimate bounds such that approximately α of training distribution's mass is captured, hence $\alpha = \mathbb{P}[L \leq \log p^\theta(X) \leq U]$ = True Positive Rate (TPR) and False Positive Rate FPR $= \mathbb{P}[L \leq \log p^\theta(Q) \leq U]$. The effectiveness of this method thus depends on the overlap between two distributions, with less overlap being better as this makes them more separable and thus yields better performance. This highlights the importance of exploring possible inductive priors that cause outliers' likelihoods to be similar to those of the training distributions. The overlapping coefficient is a similarity between two distributions (OVL) that can be used as a measure of separability [17]. It is defined as the integral of the minimum between two density functions

$$\mathrm{OVL} = \int_{\mathbb{R}^\mathrm{d}} \min\left[p(x), q(x)\right] dx$$

The OVL is appealing due to its natural simplicity and straightforward visual interpretation. However, in our experimental setup, it is not possible to calculate the OVL directly through analytical means. Instead, we will use its aforementioned visual interpretation on the estimated KDEs to get an indication of the separability. Though this is a less quantitative approach, we argue that it is still valuable since we are not interested in the exact performance of our models; rather, we are interested in how they behave when applying perturbations, such as affine transformations, affect the model.

4 Evaluation Baseline Performance

This section provides an initial performance evaluation of the chosen model based on the simplified architecture discussed earlier. Our goal is to confirm the model's architecture capability to closely approximate its training distribution $p(x) = q_{\lambda(s=1,t=0)}(x)$ across the examined dimensionalities. This serves as a baseline and sets the stage for subsequent evaluations later in this work.

Figure 1 shows the KDEs of the estimated and true log-likelihoods for different dimensionalities. We argue that by judging by the shape and overlap p^θ has been able to provide a good estimate for each. In addition, Figure 2 shows that the estimated cross entropy for each D is nearly identical to the ground truth. Note that the expected value of the log-likelihood of the data under the model parameters is consistently less than the expected

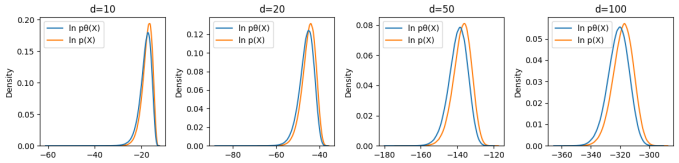


Figure 1: KDE of model $p^\theta(x)$ and training distribution $p(x)$ log-likelihood estimates for different dimensionalities.

value of the log-likelihood of the data under the true distribution $\mathbb{E}_p[\log p^\theta] \leq \mathbb{E}_p[\log p]$ also indicated by the minor left shift of the model's KDE. The fact that we observe this behaviour can be explained from an entropy point of view. The model's objective function is to maximize the likelihood function $\mathcal{L}(\theta)$, which in this case is equivalent to minimizing the cross-entropy $H(p, p^\theta)$ between the model and data distribution since

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim X}[\log p_x^\theta] = -H(p, p^\theta). \tag{15}$$

The cross-entropy can be written as the sum of $H(p)$ and KL divergence between the model and data distribution Equation 10. As the KL divergence is always non-negative, we can reorder the terms of Equation 10 to show that $\mathbb{E}_p[\log p^\theta] \leq \mathbb{E}_p[\log p]$ will always hold except when $D_{KL}(p_x||p_x^\theta) = 0$ in which case equality is achieved.

$$H(p, p^\theta) - D_{KL}(p||p^\theta) = H(p) \tag{16}$$
$$H(p, p^\theta) = -\mathbb{E}_p[\log p^\theta] \geq H(p) = -\mathbb{E}_p[\log p] \tag{17}$$
$$\mathbb{E}_p[\log p^\theta] \leq \mathbb{E}_p[\log p]. \tag{18}$$

In this scenario, we have access to the parameterization of $p(x)$, which allows for the analytical computation of $\mathbb{E}_p[\log p]$. However, for model p^θ the computation of $\mathbb{E}_p[\log p^\theta]$ which requires integration over $\mathbb{R}^D$ is not feasible. Instead, we rely on empirical estimates. Using a finite test set, we inherently lose the theoretical guarantee the aforementioned inequality holds. The likelihood of observing erratic behavior in our estimates is inversely proportional to the sample size. If the sample size is chosen to be large enough, the chances of erratic behavior are negligible. We experimented with different sample sizes and finally chose to use $N = 1000000$ for further experiments in this work. Repeated sampling of test sets $Q^{1000000}(p^\theta)$ yielded a consistent estimate of $\mathbb{E}_p[\log p^\theta]$ with $\sigma \leq 0.01$ for all tested dimensionalities, serving as a heuristic that our chosen sample size is large enough to yield a negligible chance of inconsistent estimates.

In summary, the simplified model architectures can correctly model the base case test distributions where $q_\lambda(x) = p(x)$ and performance is consistent across tested dimensionalities. Hence, we conclude the base case to be validated and will continue evaluating more complex scenarios.

5 Evaluation Generalization

5.1 Construction test-distributions

This section aims to investigate how the simplified model generalizes under different circumstances. To achieve this, we will evaluate various parameterizations of test distribution q_λ using the evaluation methods discussed in subsection 3.2. These parameterizations based on different values of s and t effectively consist of three types. They either vary s and t individually while keeping the other at their base value or modify both at the same time. Our analysis will focus on systematically exploring the effects of each type on the model's behavior. More formally, let S and T be the sets containing the chosen values of s and t respectively, then the set of test distributions denoted Λ is constructed from the mapping $\lambda(s, t)$ (Equation 12) for all (s,t) pairs in the cartesian product of sets S, T.

$$\Lambda = \{\lambda_q(s, t) \mid s \in S,\ t \in T\} \tag{19}$$

In this and subsequent sections, S will consist of values $\{0.25, 0.5, 0.75, 1, 1.25, 1.75\}$ and T of $\{0, 0.25, 0.5, 0.75\}$. Chosen to explore both compressing ($s < 1$) and expansive ($s > 1$) scaling transformations, with the baseline where s = 1 to examine various distribution scalings. T is selected to include transformations that vary between none (t = 0) and a substantial dissimilarity increase. We acknowledge that choosing specific values that represent a *substantial* increase in dissimilarity is subjective and depends on the characteristic, e.g., the variance of the baseline distribution. Based on analytical solutions to the KL divergence Figure 6, cross-entropy Figure 2, KDEs, and empirical observations, we heuristically chose S, T such that they ranged from a mild to substantial increase in dissimilarity. Additionally, we opted to include exclusively non-negative values due to the symmetric characteristics of the Gaussian distribution, yielding highly analogous results for positive and negative values.

The results for each member of Λ, over all tested dimensions, will lead to a large results set. Hence, in subsequent sections, we will use subsets of our results relevant to the research questions that will be discussed.

However, the full result set can be found in Appendix A and Appendix B.

5.2 Results and discussion

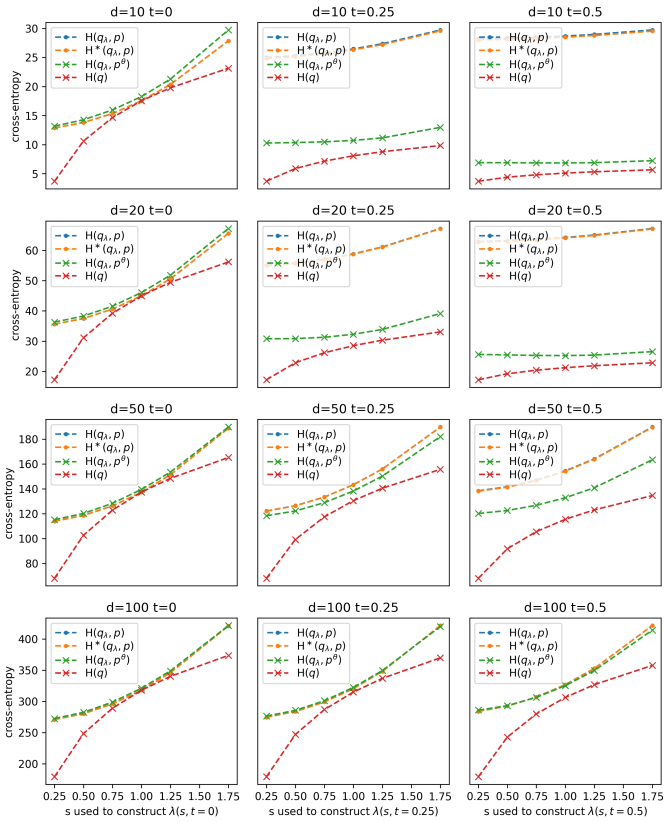


Figure 2: Cross entropy for different D where the plots have a fixed t in each column. The cross-entropy estimates are plotted as a function of s

5.2.1 Modifying s

For Gaussian in high dimensional space, the mass tends to concentrate in a shell at a fixed distance from the mean. Hence, even though the maximum density is still at the mean, the total probability mass in this high-density region is very small. Consider a contractive operation that reduces $s < 1$. The resulting test distribution q_λ will have more probability mass towards the mean relative to the training distribution. In other words, decreasing s will result in a test set $Q(q_\lambda)$ oversampled from high-density yet low-mass regions of $p(x)$. Figure 2 shows the evolution of the cross-entropy for the analytically computed $H(q, p)$ relative to $H^*(q, p)$ and $H(q, p^\theta)$ over all test sets $\{\mathcal{Q}(q_\lambda) \mid \lambda \in \Lambda\}$. First, note that the analytical $H(q, p)$ and estimated $H^*(q, p)$ cross-entropy are approximately equal, which, as discussed before, we take as a heuristic that the chosen sample size is sufficiently large. For a model that generalizes well, the gap between the two approximations $\Delta_{HC} = |H^*(q_\lambda, p) - H(q_\lambda, p^\theta)|$ should remain as small as possible. The first column in Figure 2 shows for each tested dimensionality the results when only s is modified. The gap Δ_{HC} remains fairly constant for both the expanding $s > 1$ and contracting $s < 1$ test distributions. Hence, we argue that the model, in these scenarios, generalizes well in terms of the cross-entropy's point estimate. Figure 3 show the

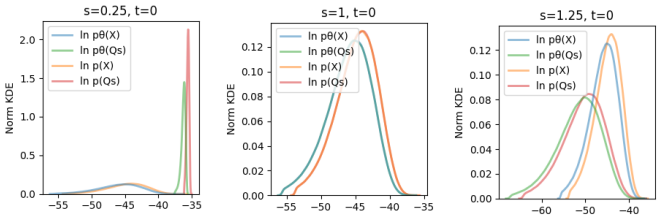


Figure 3: Estimated KDEs for model $\log p^\theta$ and ground truth $\log p(x)$ over $\mathcal{Q}(q_\lambda)$ for $D = 20$.

estimated KDEs for different a subset of the evaluated test distribution. The relatively constant gap Δ_{HC} in cross-entropy is also reflected by the KDEs where the difference in modes is approximately equal to Δ_{HC}. Comparing the variances of training $\mathrm{Var}(\{\log p(x) \in \mathcal{Q}(q_\lambda)\})$ and model $\mathrm{Var}(\{\log p^\theta(x) \in \mathcal{Q}(q_\lambda)\})$ distribution over the test sets show no significant difference. Effectively, the distribution of $\log p^\theta$ is just a slightly shifted version of $\log p(x)$, which is both desired and expected as discussed in section 4. Hence, we conclude that within our experimental setup, when varying only the volume of the test distribution between $0.25 \leq s \leq 1.75$ the model p^θ is able to generalize to a test distribution q_λ adequately.

5.2.2 Modifying t

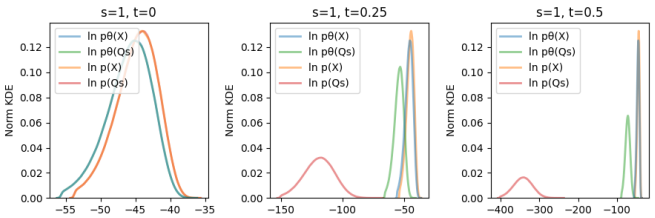


Figure 4: Example of increasing only t for D=20, the images shows estimates for both model $\log p^\theta$ and ground truth $\log p(x)$ for $\mathcal{Q}(q_\lambda)$. The estimates for the training distribution have also been included to provide context.

Using the same methodology as in the previous section, we explore how the model behaves when on test distributions where only t is modified. The second and third columns of Figure 2 show the effects on modified t with 0.25 and 0.5. As t is increased, the cross-entropy estimates $H(q, p^\theta)$ show substantial underestimation relative to both estimated $H^*(q, p)$ and true $H(q, p)$, which is equivalent to overestimation of $\mathbb{E}_{q_\lambda}\left[\log p^\theta\right]$. The magnitude Δ_{HC} consistently grows with increased t. The behaviour occurs across various dimensionalities, though it is relatively less noticeable in higher dimensions. The discrepancy between the estimates becomes especially clear when comparing the KDEs in Figure 4. After a minor shift of t=0.25 the estimated KDE for $p^\theta(x)$ and $p(x)$ over the test set $\mathcal{Q}(q_\lambda)$ are completely non-overlapping. This shows that the model is no longer able to generalize to these low-density regions, resulting and showing severe overestimation of $\mathbb{E}_{q_\lambda}\left[\log p^\theta\right]$. The latter is also reflected in the estimates of the KL-divergence shown in Figure 6. Note $D_{KL}(q||p^\theta)$ is substantially smaller than $D_{KL}(q||p)$, as shown in Figure 6. Note that an underestimated KL divergence implies that the model finds the test distribution more similar to the training distribution than it should be. As previously mentioned, when the estimated KDE for the model p^θ over the training and test distribution starts to overlap, it can significantly reduce the effectiveness of likelihood-based OOD detection.

5.2.3 Combined change of s,t

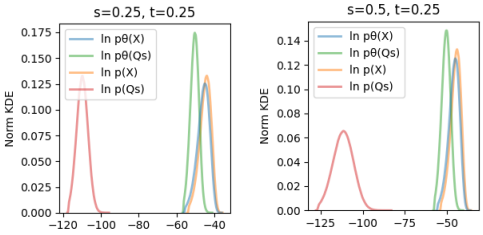


Figure 5: KDE of model $\log p^\theta$ and ground truth $\log p(x)$ over both training and test set for $D = 20$

In the previous sections, we explored the model's behavior through individual modification of either s or t and explored the effect on the cross-entropy, KL divergence, and estimated KDEs. Roughly summarizing, we found that the model was still able to generalize well to changes to changes of s, which did not substantially affect the magnitude of gap Δ_{HC}. Yet, modifying t led to an increasingly larger underestimation of $H(q, p^\theta)$. The simultaneous modification of both s and t for $t = \{0.25, 0.5\}$ shows the same impact on $H^*(q, p)$ and $H(q, p^\theta)$ as their individual modification. As before, the estimates of $H^*(q, p)$ and $H(q, p^\theta)$ in Figure 2 show the gap Δ_{HC} grows as t increases but is not substantially affected while simultaneously modifying s. For a very large shift, increasing s does further increase Δ_{HC} and is also reflected in increasingly underestimated KL-divergence $D_{KL}(q \mid\mid p^\theta)$ shown in Figure 6. This is not unexpected, as we apply scaling before translation Equation 12 thus particularly for a large t contracting the distribution before shifting results in test distribution that can be considered less challenging. Overall, the experimental results suggest that during simultaneous modification of s and t, the performance, as measured by the gap Δ_{HC}, is fairly robust. In section subsubsection 3.2.4,

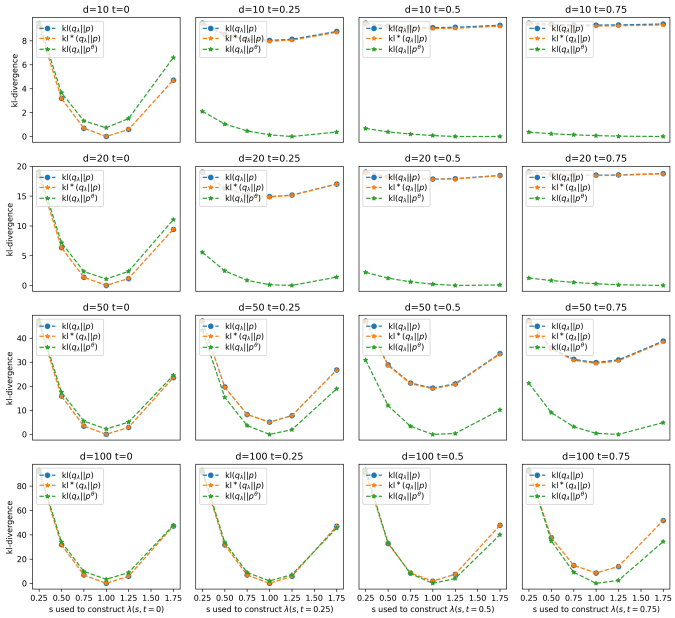


Figure 6: KL-divergence estimates for initial evaluation of generalization behavior using test-distribution $q_\lambda(x)$ constructed through affine transformations of the training distribution p.

we discussed the role of overlap and how it relates to the model's ability to perform well at likelihood-based OOD detection. Plainly put, the less overlap between the model estimates for the training distribution relative to the test distribution, the better the model will perform at the OOD detection task for this test distribution. For the distribution used in this work, any underestimation or overestimation tends to be harmful to OOD detection if it reduces the gap between the model's cross-entropy estimates over $\mathcal{Q}(p)$ and $\mathcal{Q}(q_\lambda)$ and, more

importantly, if this also leads to an increase in overlap. Once more, consider the findings from the previous two sections, where we observed that a reduction of $s < 1$ will cause $\text{Var}(\{\log p^\theta(x) \in \mathcal{Q}(q_\lambda)\})$ to decrease and increase t will cause underestimation of $H(q_\lambda, p^\theta)$. A worst-case scenario in terms of likelihood-based OOD detection would be a situation in which the underestimation of $H(q, p^\theta)$ caused by increasing t would shift $H(q, p^\theta)$ closer $H(p, p^\theta)$ to a degree where they might (partially) overlap as shown by the KDEs in Figure 4 if combined with contraction $s < 1$ that reduces $\text{Var}(\{\log p^\theta(x) \in \mathcal{Q}(q_\lambda)\})$ of the distribution that results from $s < 1$, which potentially could significantly increase the overlap. The KDEs estimated for combined modification shown Figure 5 illustrates this exact scenario. For a relatively small shift and scale modification $s = 0.75, t = 0.25$, the estimated KDEs for model p^θ over the $\mathcal{Q}(q_\lambda)$ and $\mathcal{Q}(p)$ fully overlap, making it impossible to distinguish the two based on bounds.

The findings in this section help answer a key part of our main question. By utilizing a simplified model, we explored how the model responds to different modifications to different test distributions q_λ, all of which still preserved the correlations underlying the training set. Moreover, we were able to reproduce scenarios where the likelihood is significantly overestimated, and the overlap between in-distribution and out-of-distribution overestimation becomes so large that bound-based OOD detection becomes infeasible. An immediate follow-up question we aim to explore in the subsequent section is how the model behaves when these underlying correlations are also slightly perturbed.

6 Evaluation Generalization Perturbed Distribution

6.1 Method

In continuation of the experiments in the previous section, we will again construct a set of test distributions largely derived from those of the preceding experiment. However, we will introduce two minor perturbations to these test distributions. The first is the addition of standard Gaussian noise, which causes a decrease in the strength of correlations. Formally, given a probability density function $p(x)$ its perturbation $p^\delta(x)$ is defined as

$$p^\delta(x) = \int p(t)\mathcal{N}(x \ -t \mid 0, \ \alpha\mathbb{I}^d) \ dt \tag{20}$$

The second perturbation is a slight modification of mean μ such that $\mu^\delta = \mu + \beta \text{u}$ where u is randomly sampled uniformly from $\mathcal{U}([-1, 1]^d)$. The mapping defined in Equation 12, which was utilized to construct the parameterizations of test distributions in subsection 5.2, can be amended to incorporate the aforementioned perturbations. The new mapping is given by

$$\lambda_q^\delta(s, t, \alpha, \beta) \rightarrow \{\mu_p + t + \beta u, \ s^2\Sigma_p + \alpha\mathbb{I}^d\} \tag{21}$$

The hyper-parameters α and β allow control over the degree of perturbation applied to the covariance matrix and mean, respectively. By substituting the previous mapping inside Equation 19 with the new perturbed version Equation 21, a new set of test distributions parameterizations Λ^δ are defined by

$$\Lambda^\delta = \{\lambda_q(s, t, \alpha, \beta) \mid s \in S, \ t \in T\} \tag{22}$$

In the remainder of this work, any mention of the *perturbed* distribution will refer to the modified (perturbed) version of the training distribution constructed through the process just discussed.

6.2 Results and Discussion

6.2.1 Coupling layer adaptability

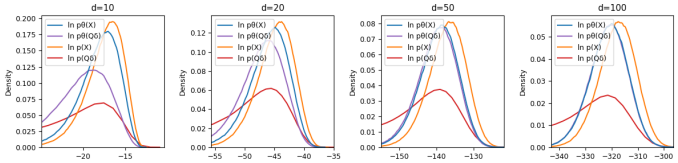


Figure 7: KDE of model and true distribution log-density estimates for the perturbed and training distribution.

Figure 7 presents the KDEs for true and model log-densities across both training set $\mathcal{Q}(p)$ and perturbed test-distribution $\mathcal{Q}(q_\lambda^\delta)$ ($s = 1$ and $t = 0$). The KDE estimated by $p(x)$ over the perturbed test set $\mathcal{Q}(q_\lambda^\delta)$ shows that q_λ^δ maintains a resemblance to the training distribution both in shape and location. Nevertheless, while being similar, it is evident that q_λ^δ is different from $p(x)$ as was the goal of our perturbation process. Yet, compared to the KDE constructed from the estimates of p^θ, a surprising observation can be made: the model appears to longer differentiate in its estimates for the perturbed and original density. This trend intensifies with increased dimensionality, so much so that for $D \geq 50$ the KDEs of

the model over both the training and test distribution become indistinguishable.

We hypothesize that the coupling layer might utilize the knowledge learned during training to offset the introduced perturbations. This hypothesis is consistent with the findings made by [3] using image data where, after removing half of the pixels in an image, they found that the coupling layers could still yield reasonable estimates for the missing pixels. They posited that each coupling layer incrementally introduces information learned during training, enabling subsequent coupling layers to leverage some of the additional information, a mechanism they termed *coupling layer co-adaptation*. However, their research, limited to visual inspections of the coupling layer's output for a single image in an extreme scenario, leaves open questions about the model's behavior and adaptability in various specific scenarios. The first is: Does the typicality of a sample impact the degree of this compensation phenomenon? Specifically, for a sample closely resembling those in the training distribution, would the compensation effect be more or less pronounced compared to samples that differ greatly from the training data?

Secondly, if atypical samples trigger stronger compensation, might this mean that the model, when faced with increasingly atypical inputs, tends to disregard the actual input in favor of aligning them with more recognizable, densely populated areas of the data distribution

The remaining part of this section will focus on exploring the first part of this question by analyzing the various test distribution parameterizations in Λ^δ following the same approach used in section 5. Allowing us to evaluate how the model will behave when the perturbed distribution through scaling or shifting alters its similarity to the training distribution. We will compare the results of different s,t to those on the non-perturbed test distributions discussed in section 5. The robustness evaluation in section 7 will explore further how the model will behave when presented with complete outlier examples.

6.2.2 Modification of s, t

The evaluation in section 5 included an in-depth analysis into the model's behavior when subjected to different modifications, be it combined or individual, of s and t. As previously noted, this work does not focus on the exact numerical values of our evaluation metrics but instead on the general patterns in our evaluation metric that can be observed when applying a modification. We

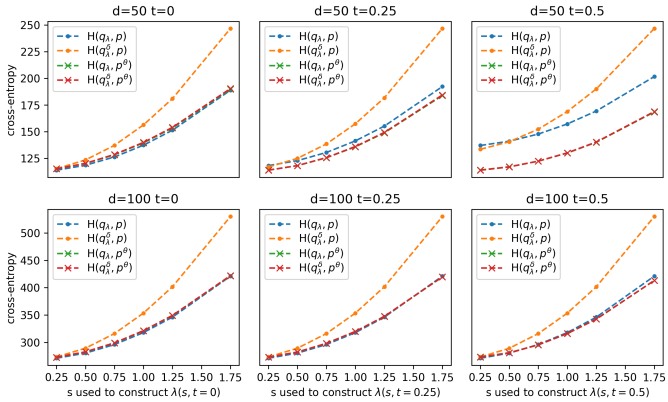


Figure 8: Cross-entropy estimates for different affine transformations of perturbed test distributions q_λ^δ and training distribution q_λ.

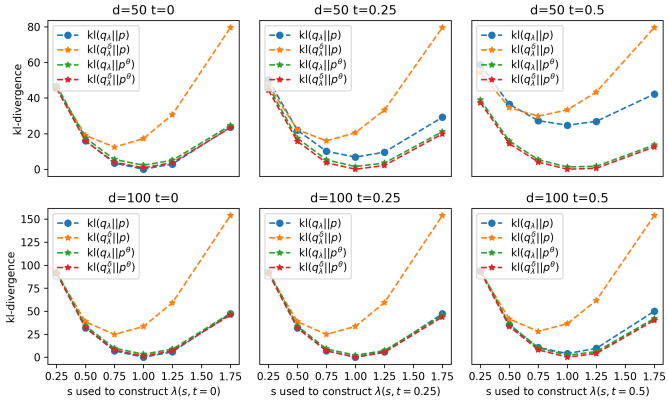


Figure 9: KL-divergence estimates for different affine transformations of perturbed test distributions q_λ^δ and training distribution q_λ.

found that the trends discussed in section 5 are very similar to those of the perturbed test distribution $q_\lambda^\delta \in \Lambda^\delta$ apart from the absolute magnitude. This is unsurprising since the perturbed distributions still closely resemble the training distribution from which they were derived. Hence, when analyzing different s,t will instead focus on highlighting the differences between their training and perturbed distribution, further analyzing the question raised earlier in this section regarding the effects of the test distribution typically relative to the training distribution.

Figure 8 presents the cross-entropy estimates for the analytically computed $H(q^\delta, p)$ relative to $H(q, p^\theta)$ over perturbed test sets $\{\mathcal{Q}(q_\lambda^\delta) \mid \lambda^\delta \in \Lambda^\delta\}$. From gap $|H(q_\lambda^\delta, p) - H(q_\lambda^\delta, p^\theta)|$ it is evident that the cross-entropy estimates based on ground truth $p(x)$ diverge notably

from those estimated using p^θ, especially when s is increased contrasting with our earlier findings in section 5 shown in Figure 2, where the disparity in the cross-entropy remained relatively consistent regardless of modification of s. It should be noted that an increment in s leads to a more significant increase in divergence between q_λ^δ and p as than it does between q_λ and p, posing a more pronounced challenge. This effect is illustrated by the variation in the analytically computed $D_{KL}(q_\lambda^\delta \mid\mid p)$ as relative to $D_{KL}(q_\lambda \mid\mid p)$, which are depicted in Figure 6 and Figure 9, respectively.

Interestingly, upon comparing the gap between the estimated cross-entropy estimates between the model p^θ estimates over $\{\mathcal{Q}(q_\lambda^\delta) \mid \lambda \in \Lambda\}$ with those over $\{\mathcal{Q}(q_\lambda^\delta) \mid \lambda^\delta \in \Lambda^\delta\}$ we find for a given combination of s,t the gap $|H(q_\lambda^\delta, p^\theta) - H(q_\lambda, p^\theta)|$ remains reasonably small. The same holds when we compare the gap in the KL divergence $|D_{KL}(q_\lambda^\delta \mid\mid p^\theta) - D_{KL}(q_\lambda \mid\mid p^\theta)|$. Moreover, for the more challenging scenarios e.g., $t = 1$ and $s > 1$, the KL-divergence estimate for the perturbed test distribution $D_{KL}(q_\lambda^\delta \mid\mid p^\theta)$ will even be smaller than $D_{KL}(q_\lambda \mid\mid p^\theta)$. Suggesting that the model may perceive the perturbed distribution to be more similar to the ground truth than its non-perturbed equivalent. These findings align with our baseline evaluation where KDEs of the model estimate for $\mathcal{Q}(q_\lambda)$ and $\mathcal{Q}(q_\lambda^\delta)$ were almost identical, further re-enforcing our earlier hypothesis that the model, to some degree, appears to ignore the added perturbations and assigns densities similar to the training distribution.

These findings allow us to address our earlier question regarding the impact of test distribution typicality on the model's adaptability. Firstly, our experiments using perturbed distributions show the typicality of the test distribution does impact the degree of the hypothesized coupling layer adaptability. Or at least indicate an inverse relation between the degree of test-distributions atypicality and the model's ability to distinguish it as atypical. Allowing a situation to occur where two test distributions exist, one of which is more atypical compared to the training distribution, yet paradoxically, the more atypical distribution is perceived by the model to be more similar.

7 Evaluation of Robustness

In this section, we will explore model robustness using a test distribution q_λ^{out} constructed as an outlier distribution. This means their correlations, covariance, and means are entirely different.

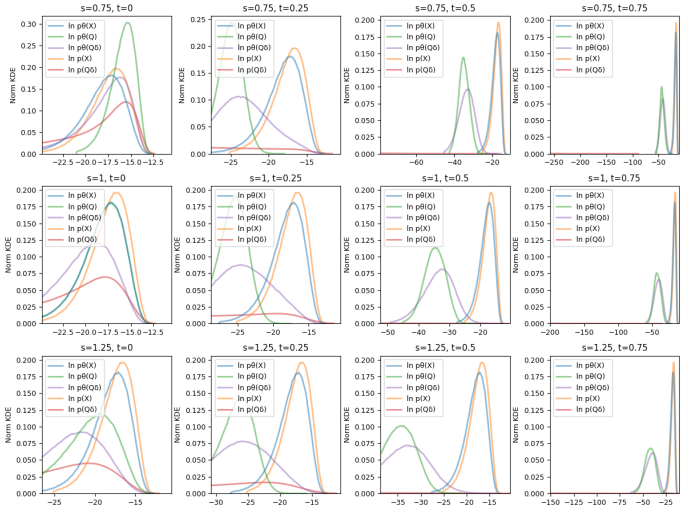


Figure 10: KDEs of $\log p^\theta$ and ground truth $\log p(x)$ for $\mathcal{Q}(q_\lambda^\delta)$ while s and t are adjusted either separately or simultaneously for D=50

7.1 Results and discussion

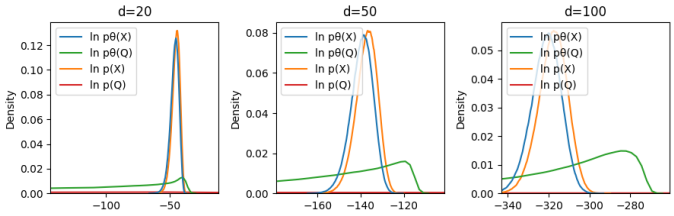


Figure 11: KDE of model and true distribution LLH estimate using outlier data for different dimensionalities shown next to those of the training distribution.

Figure 11 shows the estimated KDEs for the model p^θ and ground truth log-densities p over both the test $\mathcal{Q}(q_\lambda^{\text{out}})$ and training set $\mathcal{Q}(p)$ accross all tested dimensionalities. When comparing the shape and location of the KDE derived from the ground truth p, it is clear that the test set $\mathcal{Q}(q_\lambda^{\text{out}})$ has been assigned vastly smaller log-densities by p compared to those assigned to the training set $\mathcal{Q}(p)$, with minimal shared mass and overlap. This result is desirable as it confirms that the outlier distribution q_λ is, as intended, substantially different from the training distribution.

Considering the characteristics of the test distribution q_λ^{out}, it is likely that the model was exposed to few samples typical of q_λ^{out} during training. Hence, expecting p^θ to produce density estimates for $\mathcal{Q}(q_\lambda^{\text{out}})$ close to those of the data distribution $p(x)$ would be overly optimistic. Nevertheless, it would be reasonable to expect the p^θ

to assign densities to $\mathcal{Q}(q_\lambda^{\text{out}})$ different than those typical of the training set $\mathcal{Q}(p)$. Contrary to expectations, the KDEs illustrate that p^θ overestimates densities for $\mathcal{Q}(q_\lambda^{\text{out}})$ to such a degree that its KDE shows a large overlap with the $\mathcal{Q}(p)$, as depicted in Figure 11. This trend becomes more apparent as dimensionality increases, to the extent that for $D = 100$, q_λ^{out} is assigned even slightly larger densities than the training distribution.

7.1.1 Variations of the outlier distribution.

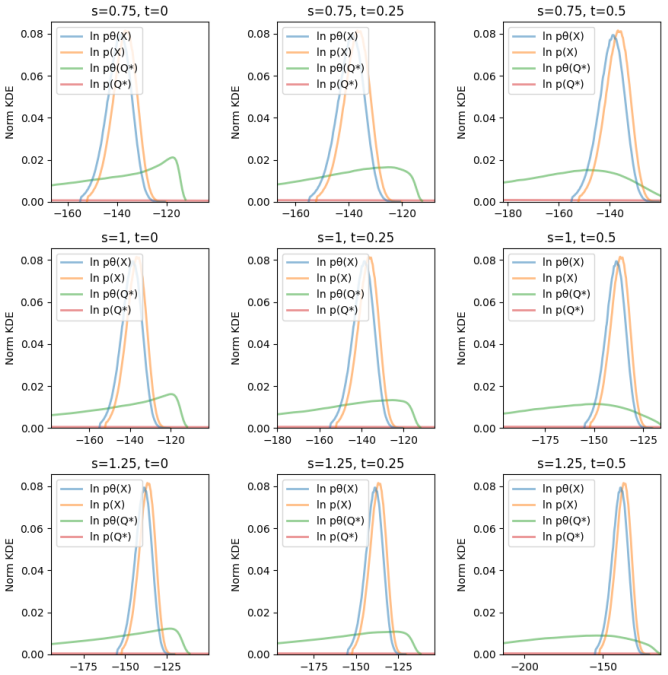


Figure 12: KDEs of $\log p^\theta$ and ground truth $\log p(x)$ for the outlier testset $\mathcal{Q}(q_\lambda^{\text{out}})$ while s and t are adjusted either separately or simultaneously for D=50

Using the same method as in the generalization experiments, we will modify the outlier distribution, using different variants of its parameterization in Λ^{out} and explore their effects. A selection of KDEs over $\{\mathcal{Q}(q_\lambda^{\text{out}}) \mid \lambda^{\text{out}} \in \Lambda^{\text{out}}\}$ where s and t are adjusted separately or simultaneously is illustrated by Figure 12. These show that for the tested s, t overestimation always occurs. Additionally, the modifications result in behavior consistent with the findings of the generalization experiments across the tested dimensionalities. That is, a decrease in s yields an increase in the log-likelihood and a decrease in variance. An increase in s leads to a decrease in the log-likelihood and an increase in variance, and an increase in t results in a decrease in the log-likelihood. Simultaneous modification of s,t to simply results in a combination of their individual effects.

To some degree, these findings align with those using perturbed test distribution q_λ^δ in section 6, where we used a perturbed test distribution q_λ^δ to evaluate our model's performance. We found that as the test distribution became more atypical, the cross-entropy was increasingly underestimated. However, for extreme values of s and t (such as s=1.25 and t=0.75), the perturbed test distribution still had a small overlap, even though the gap between $H(q_\lambda^\delta, p)$ and $H(q_\lambda^\delta, p^\theta)$ became larger. In such cases, the model could still accurately detect OOD data.

8 Outlier Detection Performance

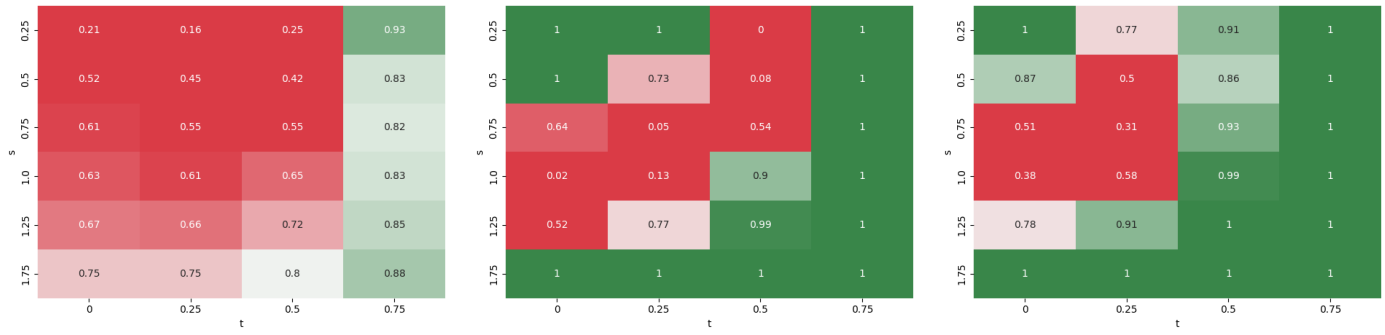
In this section, we will explore how different combinations of parameters s and t impact out-of-distribution (OOD) detection. Once more, note that our goal is not to hone in on exact numerical values but rather to identify which scenarios are most challenging and extrapolate upon these findings to provide a better understanding of the factors that influence the performance of these models in real-world scenarios.

8.1 Method

Our method, already briefly discussed in subsubsection 3.2.4, will use straightforward lower (L) and upper (U) bounds where any value falling outside these bounds is considered an outlier, such that the decision function $D_{\text{out}}(x)$ is given by

$$D_{\text{out}}(x) = \begin{cases} 1 & \text{if } \log p^\theta(x) < L \text{ or } \log p^\theta(x) > U \\ 0 & \text{otherwise} \end{cases}$$

The precise upper and lower bounds were determined by employing estimates of $\log p^\theta$ over the $\mathcal{Q}(p)$ to ensure that the true positive rate (TPR) equals a constant predetermined level. While the minimum required TPR will always be application-dependent, for the purpose of this discussion, we choose a TPR of 0.975, which we deemed reasonable for a general classifier. Subsequently, we estimated the true negative rate (TNR) for both the perturbed $\{\mathcal{Q}(q_\lambda^\delta) \mid \lambda^\delta \in \Lambda^\delta\}$ and outlier $\{\mathcal{Q}(q_\lambda^{\text{out}}) \mid \lambda^{\text{out}} \in \Lambda^{\text{out}}\}$ test sets. The results of each are depicted in Figure 13. The TNRs for the ground truth $\log p(x)$ over $\{\mathcal{Q}(q_\lambda^{\text{out}}) \mid \lambda^{\text{out}} \in \Lambda^{\text{out}}\}$ were not included in Figure 13 because they had an average score of 0.98 with a minimum of 0.96. This means that the selected bounds would be enough for accurate OOD detection on



(a) TNR based on $\log p^\theta(x)$ for $\mathcal{Q}(q_\lambda^{\text{out}})$ (b) TNR based on $\log p^\theta(x)$ for $\mathcal{Q}(q_\lambda^{\delta})$ (c) TNR based on $\log p(x)$ for $\mathcal{Q}(q_\lambda^{\delta})$

Figure 13: True Negative Rate (TNR) for different OOD detection tasks for D=50 with fixed TPR set to 0.975

the outlier distributions, given that someone can access $p(x)$.

8.2 Discussion and Results

8.2.1 OOD detection performance for $\mathcal{Q}(q_\lambda^{\text{out}})$

In section 7, outlier test-distributions $\{q_\lambda^{\text{out}} \in \Lambda^{\text{out}}\}$ were designed to be substantially different from the training distribution $p(x)$. For these atypical distributions, we anticipated that the estimates $p^\theta(x)$ might not closely match $p(x)$, but we did expect them to still differ sufficiently from those across the $\mathcal{Q}(p)$ to enable effective out-of-distribution detection. However, subsequent analysis in subsection 7.1 revealed severe overestimation combined with a significant overlap between the training and test-distributions KDEs as shown by Figure 12. Consequently, the subpar TNRs in Figure 13a align with our earlier results and further illustrate poor performance at the OOD task, particularly for q_λ^{out} with smaller variance than p, a trend earlier discussed in subsubsection 5.2.3.

8.2.2 OOD detection performance for $\mathcal{Q}(q_\lambda^{\delta})$

The perturbed test distributions constructed in section 6 were designed to maintain a large degree of resemblance to $p(x)$. As was illustrated in the KDE of the perturbed distribution in Figure 10, the perturbed tests distributions still overlap with the $p(x)$. Hence, apart from the most atypical variants, e.g., $t \geq 0.5$, a perfect separation between these distributions is impossible, even with access to a perfect density model. The TNR estimated using $p(x)$ over $\mathcal{Q}(q_\lambda^{\delta})$ shown in Figure 13c further illustrate this, showing that while the perturbed test distribution is similar to the training distribution, they still differ. However, when we compare the latter results with the TNR estimated using p^θ in Figure 13b, we find that the TNR is very small 0.02 vs 0.38 for the ground truth, indicating the inability of two to distinguish the test and training distribution. Moreover, for s=0.25 and t=0.5 the TNR for p^θ is close to very close while the p provides a TNR of 0.91. This result aligns with our previous findings discussed in subsubsection 6.2.2, further strengthening our hypothesis that p^θ, to some degree, *ignores* the perturbations in its estimates. As in previous experiments, the TNR of p^θ is particularly vulnerable to contractions $s < 1$ combined with a shift.

8.2.3 Combined implications

This analysis reveals a significant challenge in detecting low-variance, shifted data as outliers. This aligns with our previous results based on analysis of the cross-entropy, KL-divergence, and KDEs, where we found the model struggles in the same scenario.

9 Discussion and Conclusion

In this section, we will summarize our experiment's findings and discuss their collective implications.

In section 5, we conducted initial experiments to investigate the behavior of the model across different test distributions $\{q_{\lambda_{s,t}} \in \Lambda\}$, which represented scaled or shifted variants of the training distribution $p(x)$ yet maintained its underlying correlations. When varying s, effectively contracting or expanding the training distribution, the estimated KDEs of $p^\theta(x)$ and $p(x)$ mostly overlapped. Moreover, the cross-entropy gap between actual and estimated log-density $|H^*(q_\lambda, p) - H(q_\lambda, p^\theta)|$ remained largely unaffected. Contrarily, increasing t caused cross-entropy estimates $H(q_\lambda, p^\theta)$ to show substantial underestimation relative to both estimated $H^*(q_\lambda, p)$ and true $H(q, p)$ and implicitly overes-

timation of the log density. Analysis of their combined modification in subsubsection 5.2.3 demonstrated that a combination of a slight shift t=0.25 and minor contraction s=0.75 caused a nearly complete overlap of the test and training distributions KDEs effectively rendering the model useless for OOD detection and gave rise to the hypothesis that scenarios in which the variance of the test distribution is smaller than the training distribution combined with a shift could be particularly harmful to OOD detection.

These initial results addressed two critical parts of our research objectives: Firstly, reproduction showing both overestimation or underestimation can be reproduced in simplified normalizing flows and is therefore not limited to complex architectures or data with greater dimensionality or complexity indicating the presence of an inductive prior in the core model. Secondly, identifying and reproducing a scenario where the latter will also render the model ineffective at the OOD detection task and, moreover, providing an initial explanation for this behavior.

Experiments in section 6 addressed the natural follow-up question of whether the previous results would also hold for test distributions perturbed with Gaussian noise to decrease correlation strengths minor, combined with a minor perturbation of the mean using uniformly distributed noise. Interestingly, an initial evaluation of test-distribution q_λ^δ for s=1, t=0, showed no significant difference in the estimated KL-divergence, cross-entropy, and KDE of the model p^θ over $\mathcal{Q}(q_\lambda)$ compared to $\mathcal{Q}(q_\lambda^\delta)$. In other words, the model *ignores* the perturbations in its density estimates such that $\mathcal{Q}(p)$ and $\mathcal{Q}(q_\lambda^\delta)$ are effectively indistinguishable. The adaptability of the coupling layers to reproduce missing or altered by aligning them with more familiar patterns in the data distribution was hypothesized by [3]. In subsubsection 6.2.1, we discussed this phenomenon, raising the question of whether more atypical inputs could cause the model to ignore further the actual input values in favor of placing them into denser areas, potentially inflating the density estimates of atypical distributions and if so, to what degree?

In subsubsection 6.2.2, we addressed the first part of this question by showing that for increasing atypical variants of q_λ^δ constructed (through variations in s and t), the model's ability to provide accurate density estimates for $\mathcal{Q}(q_\lambda^\delta)$ diminished. Interestingly, given a specific s, t, the model estimates in terms of cross-entropy, KL-divergence, and KDE for q_λ^δ kept closely resembling those of their non-perturbed counterpart q_λ. This provides further evidence of an inverse relationship between the typicality of the test distribution and the model's ability to discriminate it from its training distribution.

Our evaluation of model robustness in section 7 further explored this relation by evaluating p^θ over outlier distributions $\{\mathcal{Q}(q_\lambda^{\text{out}}) \mid \lambda^{\text{out}} \in \Lambda^{\text{out}}\}$ specifically constructed to be completely atypical to the training set. We found underestimation of both the $H(q_\lambda^{\text{out}}, p^\theta)$ and $D_{KL}(q_\lambda^{\text{out}} \mid\mid p^\theta)$. Moreover, there is considerable overlap between the kernel density estimates for $\mathcal{Q}(p)$ and those of the $\{\mathcal{Q}(q_\lambda^{\text{out}}) \mid \lambda^{\text{out}} \in \Lambda^{\text{out}}\}$. Similar to our findings in section 5 and section 6, OOD performance degrades, especially in scenarios where the variance of q_λ^θ is smaller than p. Given the atypicality of the outlier distributions, it would be overly optimistic to expect p^θ to supply estimates that would allow separation between them. Hence, an intriguing aspect of our findings is not just the occurrence of this behavior but the degree to which it occurs.

One of the reasons behind this work was to better understand the behavior of normalizing flow on outlier data by using only the core model, paired with a well-defined training distribution, contrasting with earlier work that focused on complex architecture extensions with relatively high dimensional datasets. While we cannot state the influence of intricate architectures, our experiments have shown that the core model already exhibits behavior that makes it not robust against complete outliers and even minor perturbation of the training distribution. This strongly suggests that a lack of OOD detection performance is not caused exclusively by increased complexity either in the model or data but instead illustrates the presence of an inductive prior at the core of normalizing flows. Though we cannot make a statement about the exact nature of the inductive prior, we hypothesize that co-adaptability in the coupling layers, as discussed in subsubsection 6.2.1, plays an important role.

While this work's primary focus isn't to directly compare our results to the high likelihoods found in previous research on image data, it is worth considering the potential connections. Semantically, two sets of images might be considered outliers completely atypical. However, the inherent nature of image data is such that images retain strong correlations between adjacent pixels despite semantic differences. As pixel values are discrete and often bounded in the [0, 255] range, variances and deviations from the mean are generally inherently restricted. Deviations between means might not be sufficient for the model to consider something an outlier.

Moreover, in all of our experiments, we showed that the normalizing flows are particularly harmful to a decrease in variance $s < 1$ combined with a minor shift. These observations coincide with previous findings from earlier works [2, 3] demonstrated that a normalizing flow trained on a dataset with larger variance than the test set (ImageNet[18] and CIFAR[19]) was often not able to distinguish between them, which aligns with our findings.

Future Work and Limitations

Our pursuit of simplicity led to the removal of complex components from the architecture, leaving room for future studies to evaluate the impact of reintroducing complexity. Systematically adding complex components, evaluation, and comparison with the plain more could offer insights into their influence on the model's behavior.

We utilized data featuring a dimensionality between 10 and 100. This decision was a compromise to ensure computational feasibility amid many experiments. While we anticipate our results to also hold with higher dimensional data, it remains a hypothesis that allows for further research in this field.

References

[1] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, International Conference on Learning Representations, ICLR, 2017.

[2] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do Deep Generative Models Know What They Don't Know?," arXiv:1810.09136 [cs, stat], Feb. 2019. arXiv: 1810.09136.

[3] P. Kirichenko, P. Izmailov, and A. G. Wilson, "Why Normalizing Flows Fail to Detect Out-of-Distribution Data," arXiv:2006.08545 [cs, stat], June 2020. arXiv: 2006.08545.

[4] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, "Fully Convolutional Cross-Scale-Flows for Image-based Defect Detection," arXiv:2110.02855 [cs], Oct. 2021. arXiv: 2110.02855.

[5] N. Kumar, P. Hanfeld, M. Hecht, M. Bussmann, S. Gumhold, and N. Hoffmann, "InFlow: Robust outlier detection utilizing Normalizing Flows," arXiv:2106.12894 [cs], Nov. 2021. arXiv: 2106.12894.

[6] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Nonlinear independent components estimation," in 3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings, International Conference on Learning Representations, ICLR, 2015.

[7] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design," arXiv:1902.00275 [cs, stat], May 2019. arXiv: 1902.00275.

[8] D. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in Advances in Neural Information Processing Systems, vol. 2018-December, pp. 10215–10224, Neural information processing systems foundation, 2018. ISSN: 10495258.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems (C. Cortes, N. Lawrence, Z. Ghahramani, M. Welling, and K. Weinberger, eds.), vol. 3, pp. 2672–2680, Neural information processing systems foundation, 2014. ISSN: 10495258 Issue: January.

[10] D. Kingma and M. Welling, "Auto-encoding variational bayes," in 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, International Conference on Learning Representations, ICLR, 2014.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 2017-December, pp. 5999 – 6009, 2017. Type: Conference paper.

[12] C. E. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, pp. 379–423, July 1948. Conference Name: The Bell System Technical Journal.

[13] S. Kullback and R. A. Leibler, "On Information and Sufficiency," The Annals of Mathematical Statistics, vol. 22, pp. 79–86, Mar. 1951. Publisher: Institute of Mathematical Statistics.

[14] D. Tran, J. Liu, M. W. Dusenberry, D. Phan, M. Collier, J. Ren, K. Han, Z. Wang, Z. Mariet, H. Hu, N. Band, T. G. J. Rudner, K. Singhal, Z. Nado, J. van Amersfoort, A. Kirsch, R. Jenatton, N. Thain, H. Yuan, K. Buchanan, K. Murphy, D. Sculley, Y. Gal, Z. Ghahramani, J. Snoek, and B. Lakshminarayanan, "Plex: Towards Reliability using Pretrained Large Model Extensions," July 2022. arXiv:2207.07411 [cs, stat].

[15] J. Soch, "Kullback-Leibler divergence for the multivariate normal distribution," May 2020.

[16] T. M. Cover and J. A. Thomas, Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). USA: Wiley-Interscience, 2006.

[17] H. F. Inman and E. L. Bradley, "The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities," Communications in Statistics - Theory and Methods, vol. 18, pp. 3851–3874, Jan. 1989. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/03610928908830127.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," Jan. 2015. arXiv:1409.0575 [cs].

[19] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," University of Toronto, May 2012.

Part II

Appendices

A Log-likelihoods for basic s,t experiments

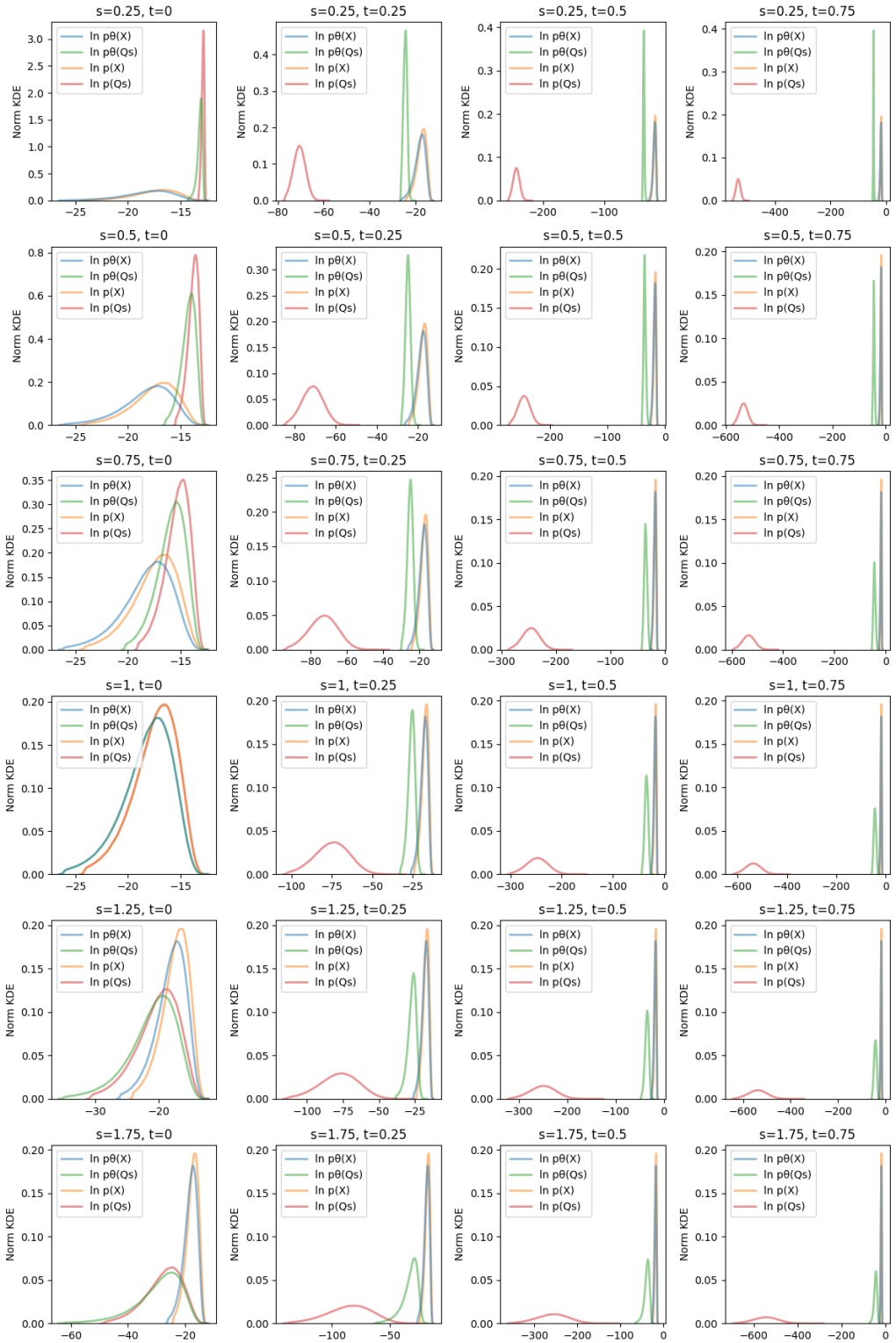


Figure 14: D=10 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different test-distribution combined with training distribution estimates.

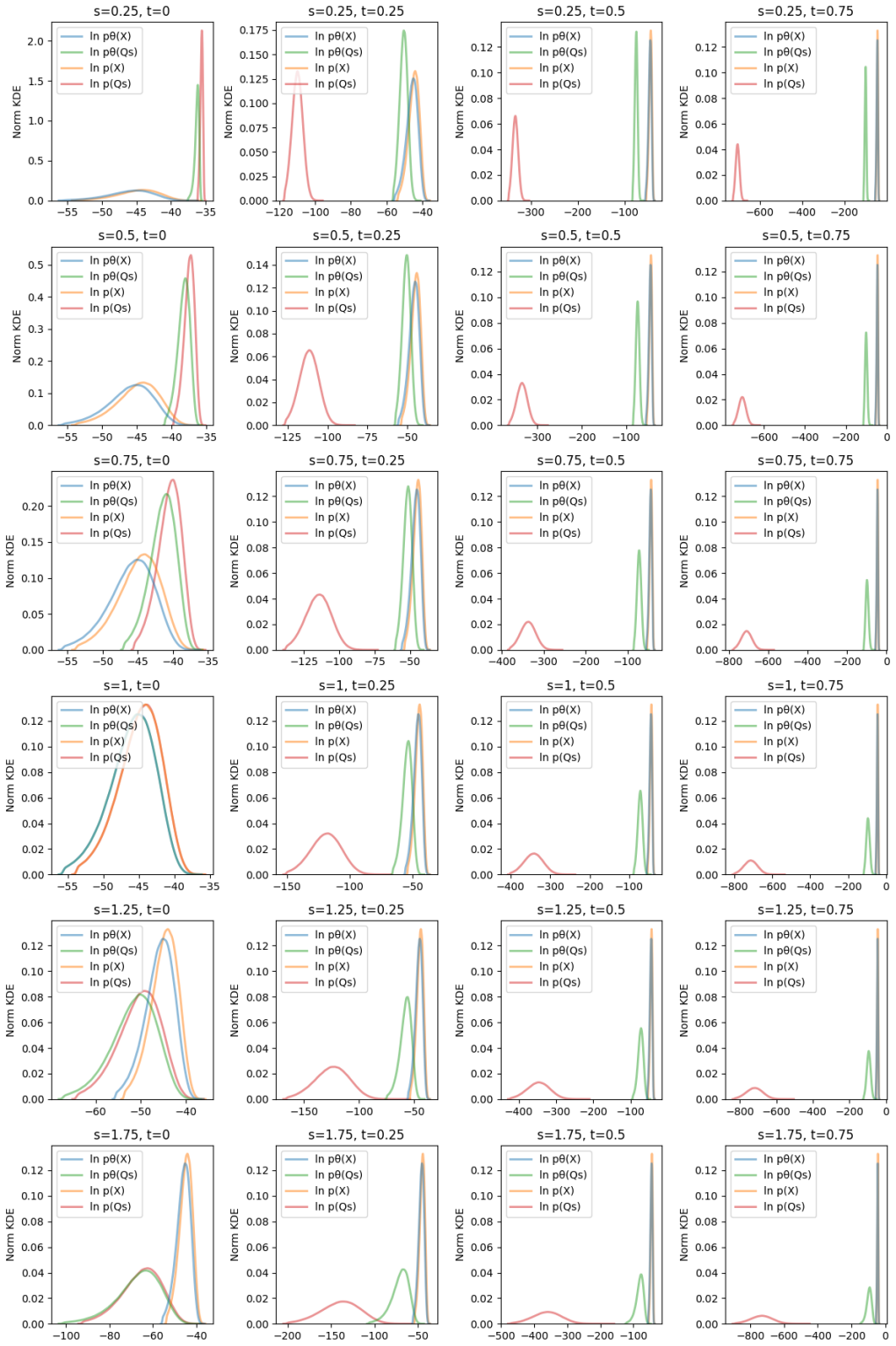


Figure 15: D=20 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different test-distribution combined with training distribution estimates.

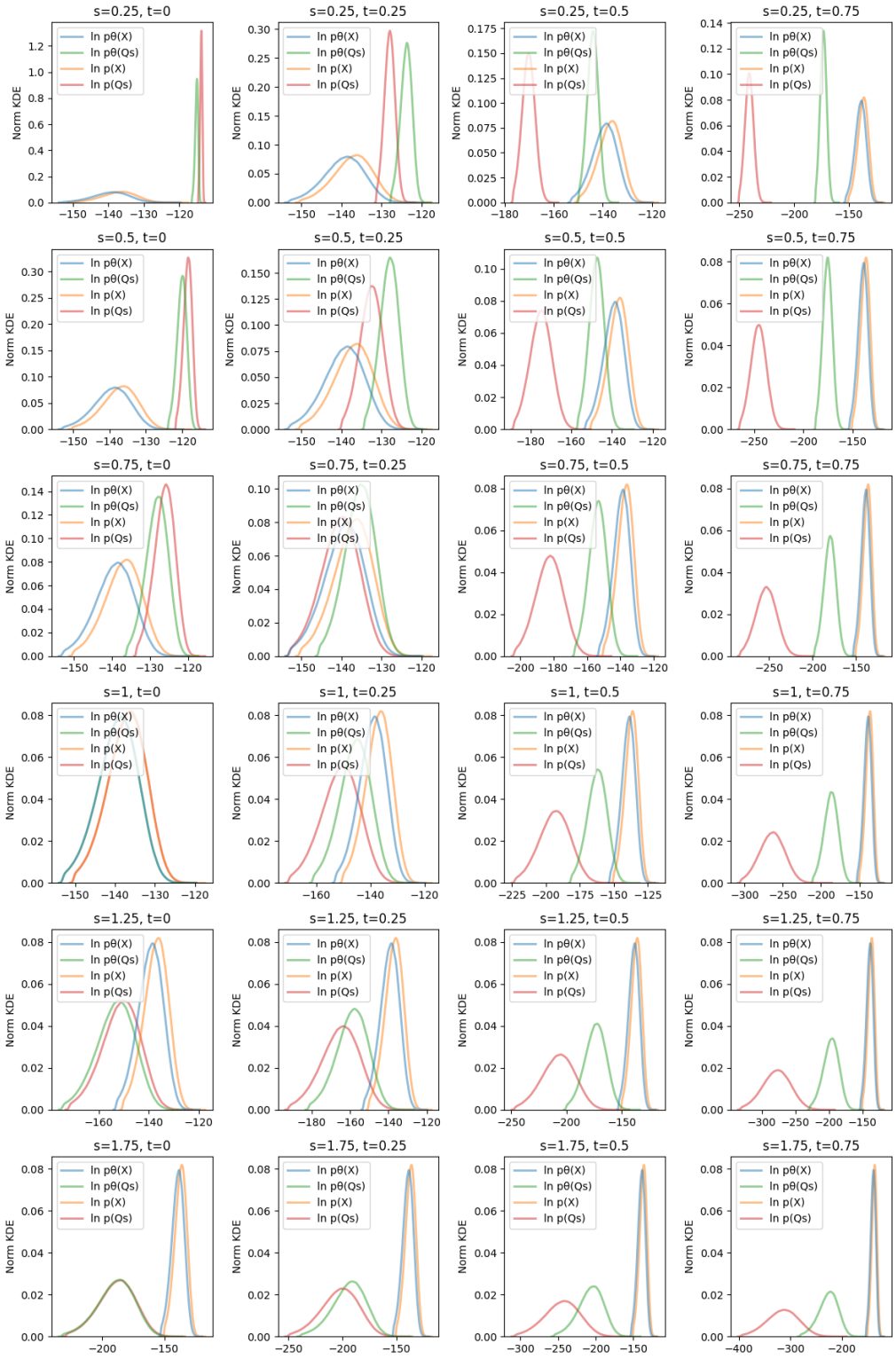


Figure 16: D=50 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different test-distribution combined with training distribution estimates.

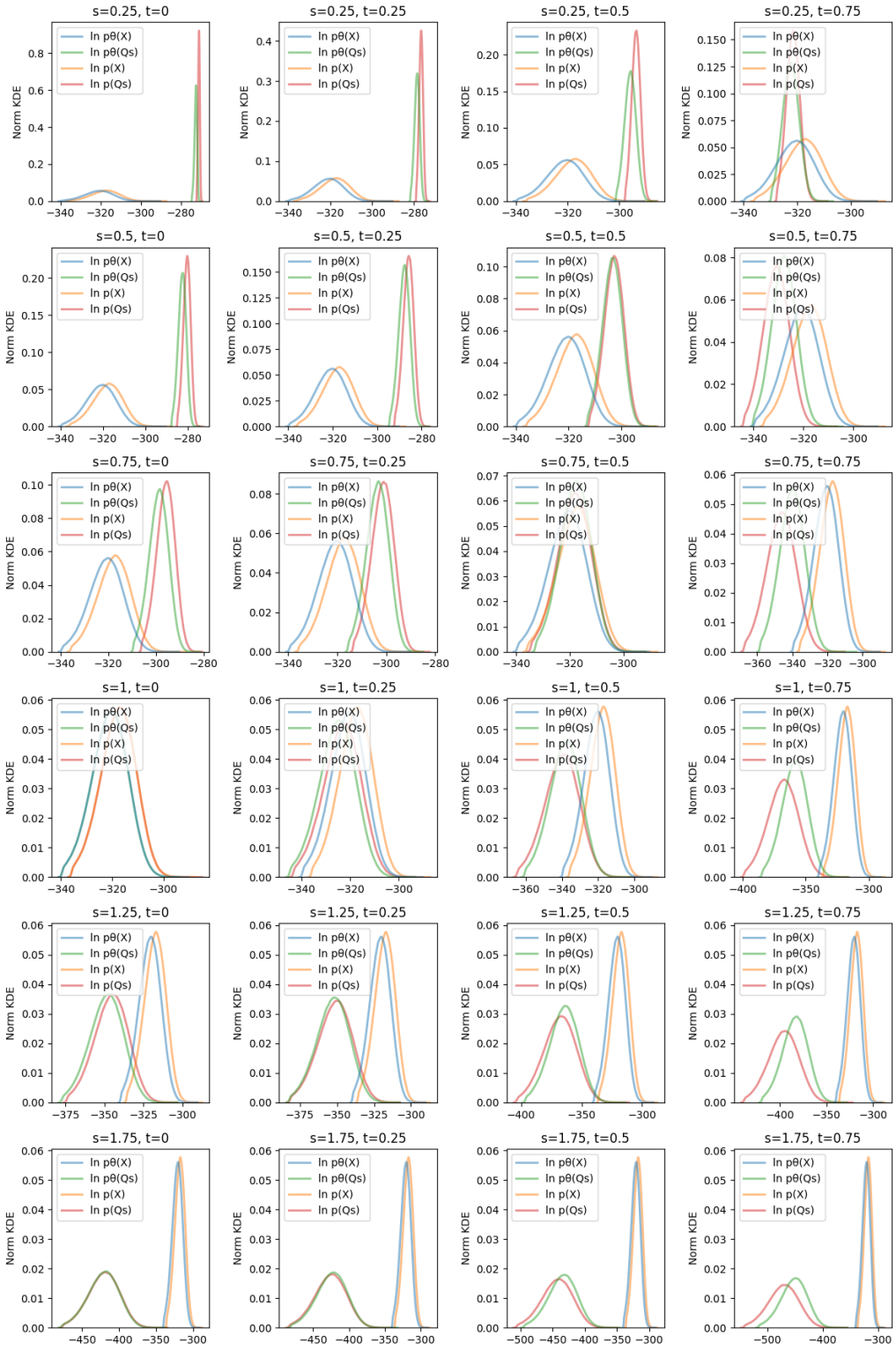


Figure 17: D=100 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different test-distribution combined with training distribution estimates.

B Log-likelihoods for perturbed s,t experiments

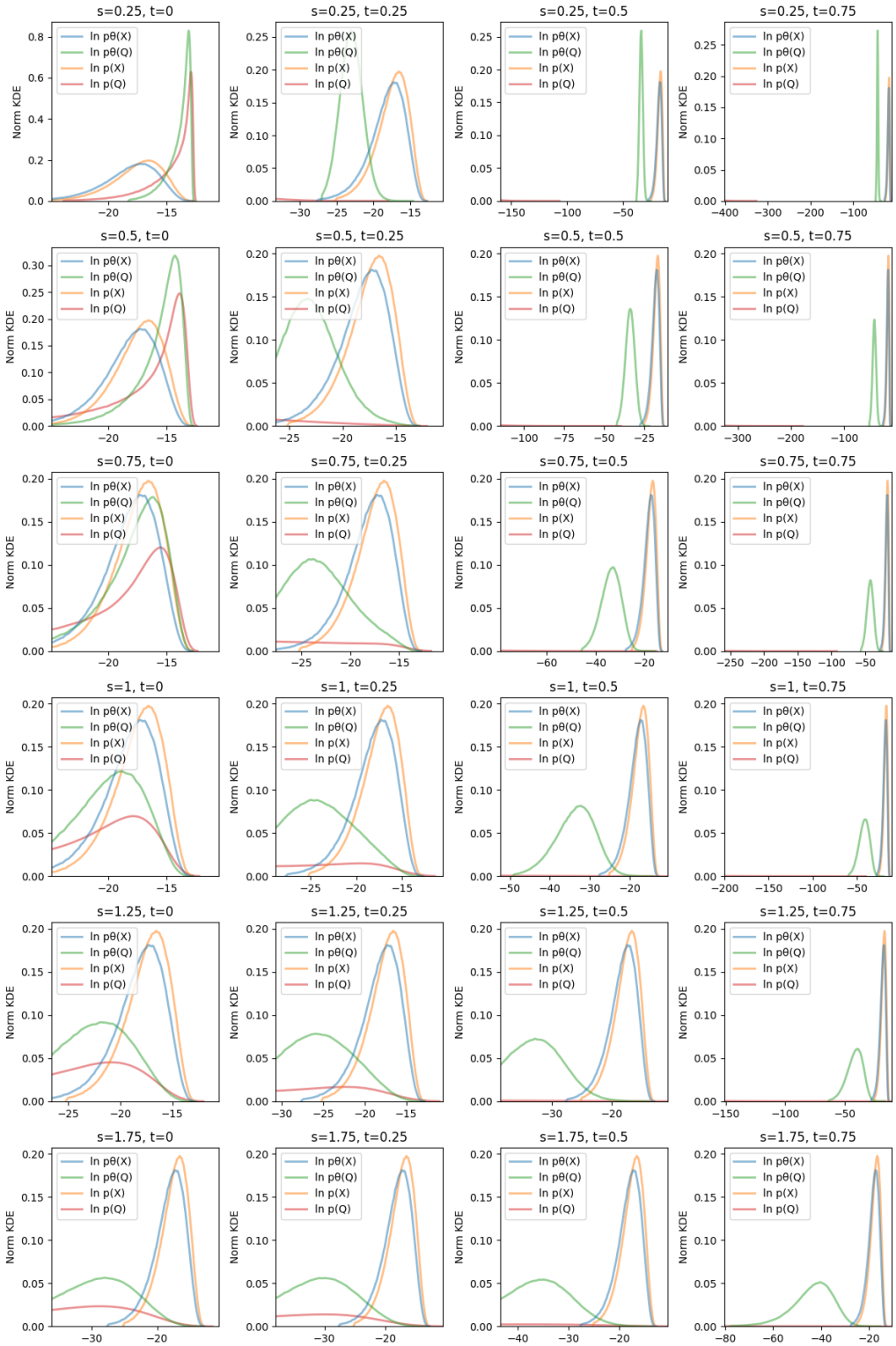


Figure 18: D=10 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different affine variants of the perturbed test-distribution q_{λ}^{δ} combined with training distribution estimates.

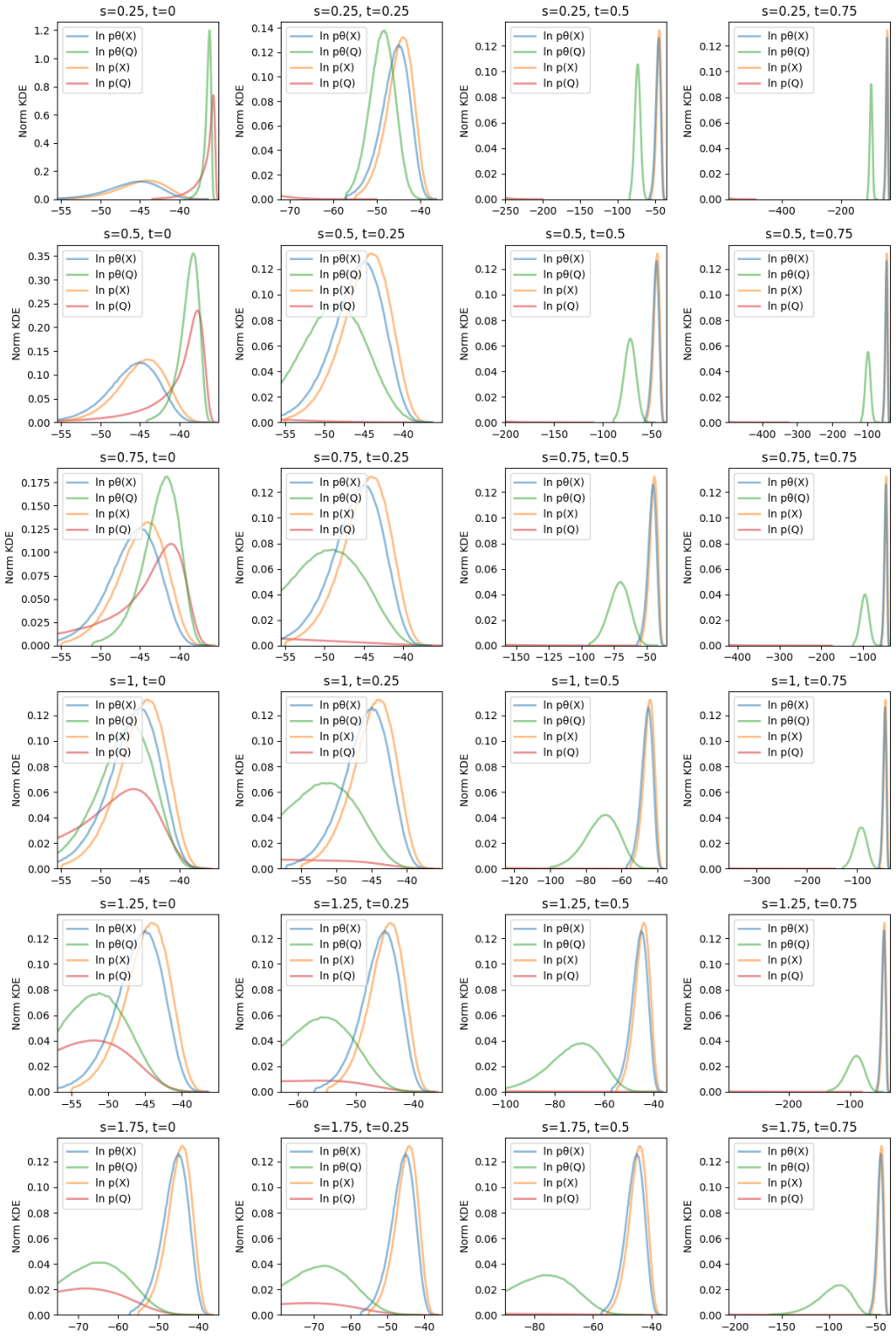


Figure 19: D=20 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different affine variants of the perturbed test-distribution q_{λ}^{δ} combined with training distribution estimates.

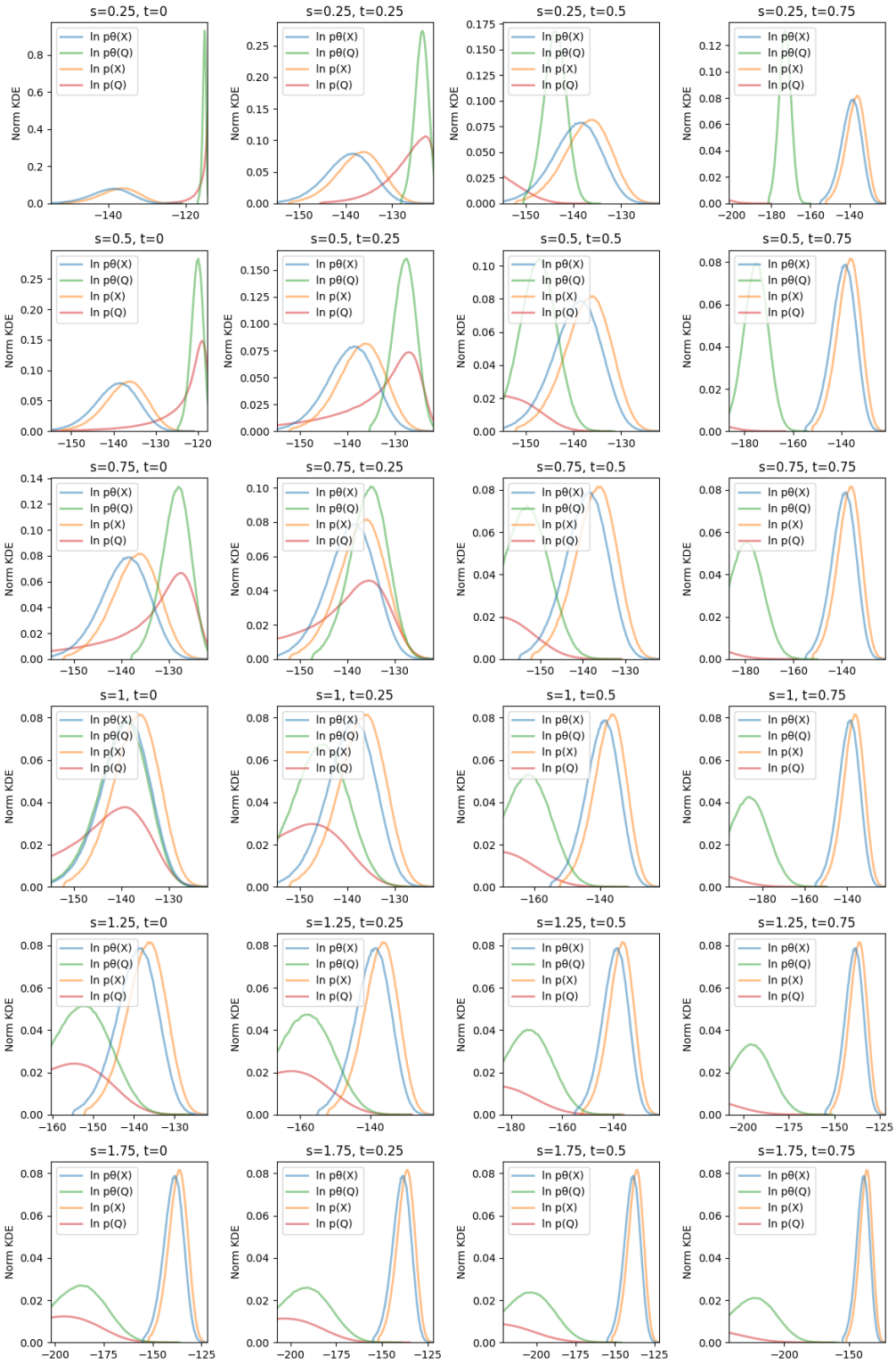


Figure 20: D=50 KDE of log-likelihoods $p(x)$ and $p^\theta(x)$ for different affine variants of the perturbed test-distribution q_λ^δ combined with training distribution estimates.

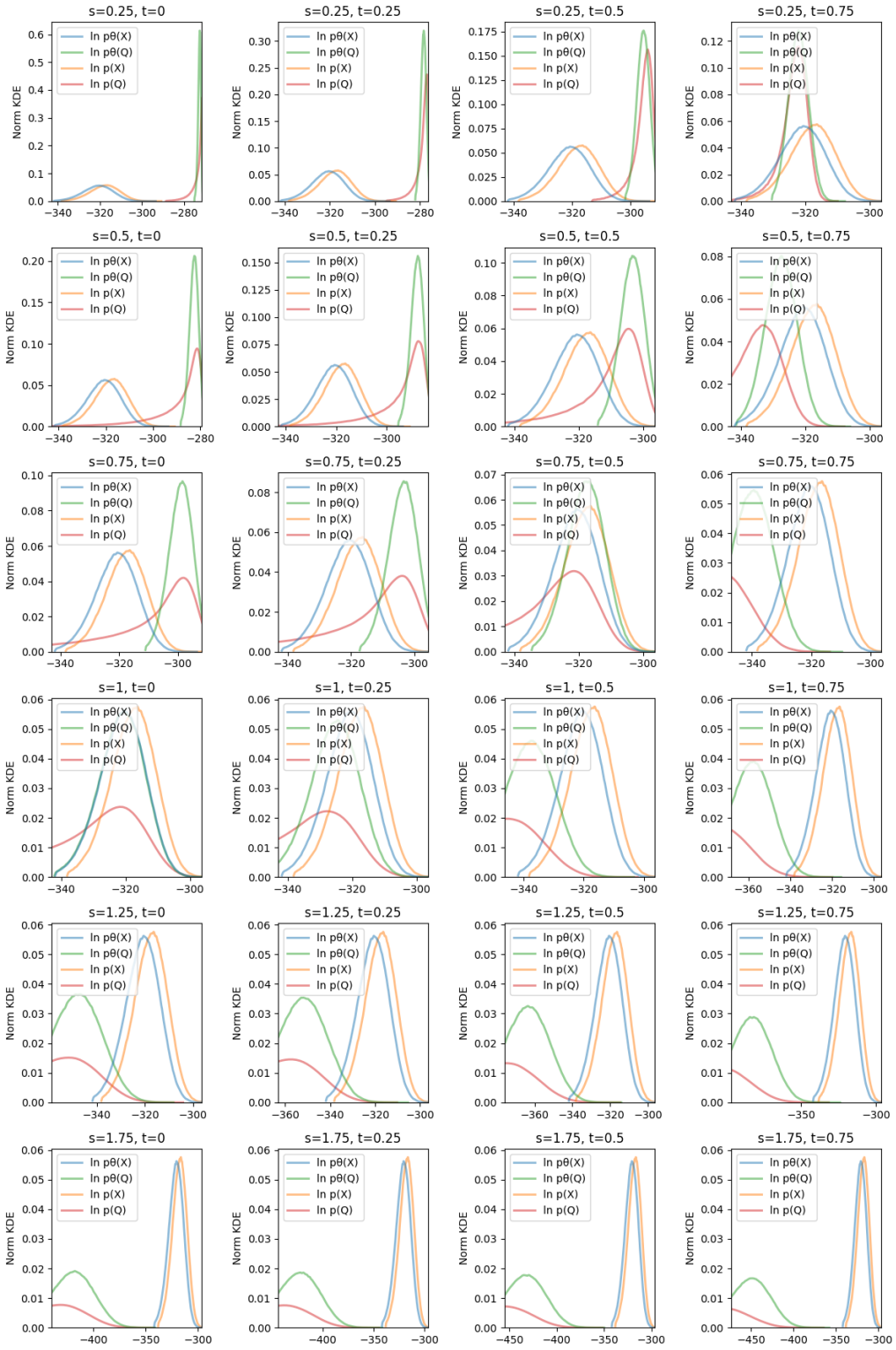


Figure 21: D=100 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different affine variants of the perturbed test-distribution q_{λ}^{δ} combined with training distribution estimates.

C Log-likelihoods for robustness s,t experiments

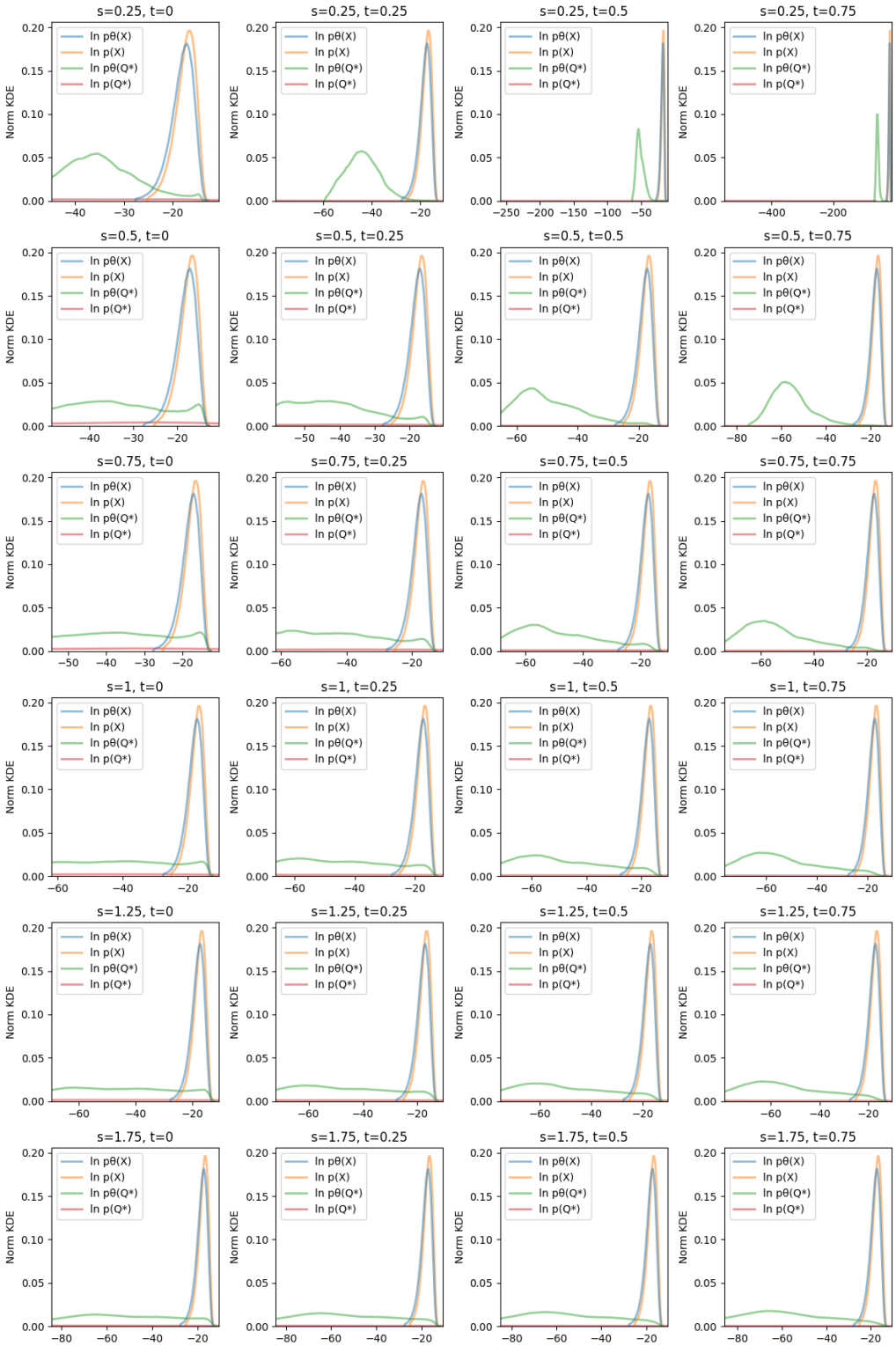


Figure 22: D=10 KDE of log-likelihoods $p(x)$ and $p^\theta(x)$ for different affine variants of the outlier test-distribution q_λ^{out} contrasted with log-likelihoods of over the training distribution estimates.

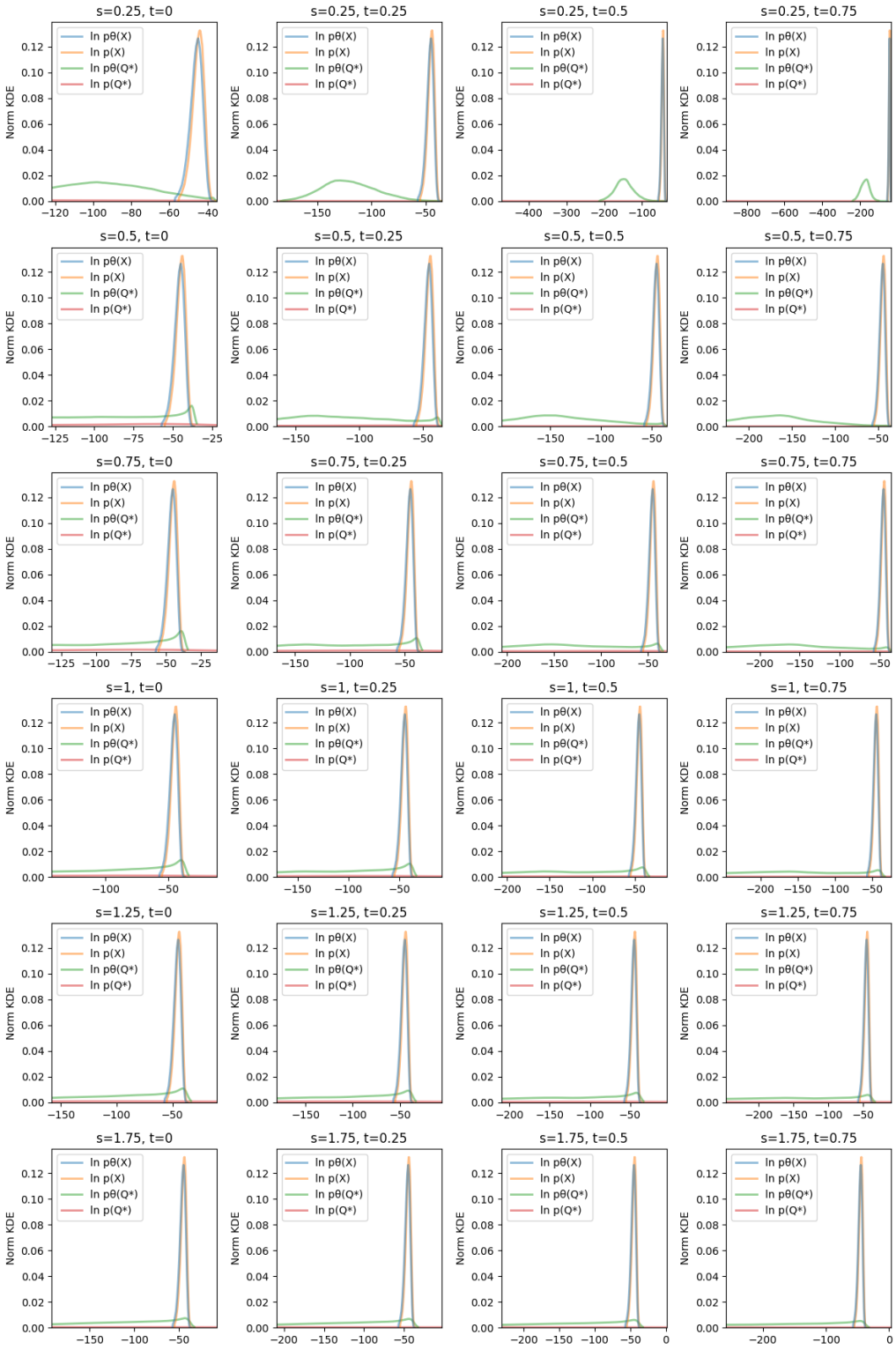


Figure 23: D=20 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different affine variants of the outlier test-distribution q_{λ}^{out} contrasted with log-likelihoods of over the training distribution estimates.

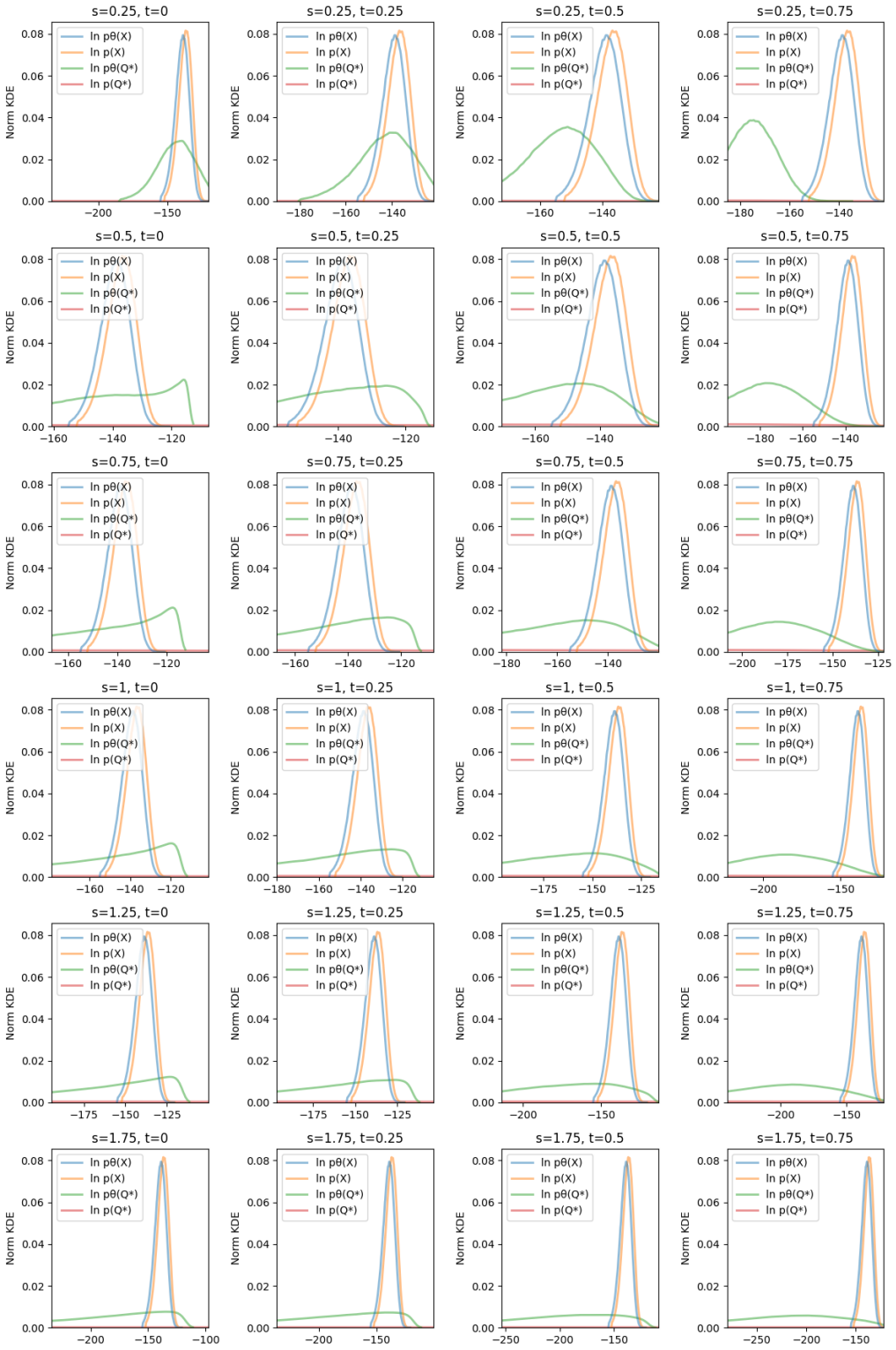


Figure 24: D=50 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different affine variants of the outlier test-distribution q_{λ}^{out} contrasted with log-likelihoods of over the training distribution estimates.

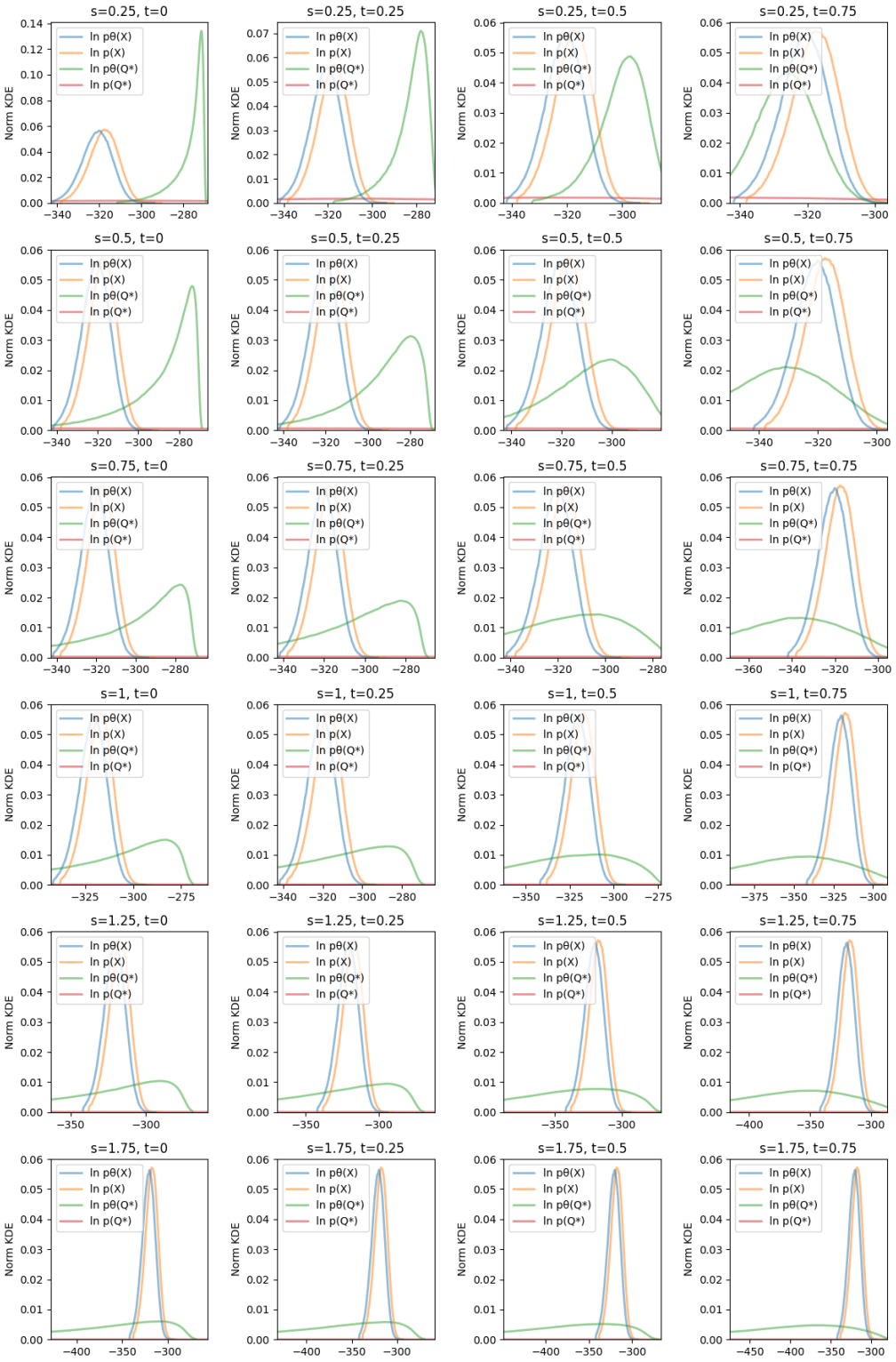


Figure 25: D=100 KDE of log-likelihoods $p(x)$ and $p^{\theta}(x)$ for different affine variants of the outlier test-distribution q_{λ}^{out} contrasted with log-likelihoods of over the training distribution estimates.

D Brief review of normalizing Flows

A Brief Overview of Normalizing Flows

Niels de Bruin
TU Delft
d.j.m.debruin@student.tudelft.nl

1. Introduction

Generative models have the ability to create new data-instances, a task which requires the ability to learn a realistic world-model approximating all dependencies in high-dimensional data; a generative model does so by approximating the joint distribution of all random variables in the dataset. The ability to create realistic world-models is a promise towards models with better generalizability. Though no overlapping theorem is present, the hypothesis is that generative models can learn efficiently from limited examples provided that the examples roughly represent the data distribution; which is crucial for a *data-efficiency*: the capability to learn from limited examples. Both generalizability and data-efficiency are two of the notable open challenges within the field of machine learning [1].

Though substantial progress has been made in generative models for tasks such as super-resolution over the last few years, a core problem remains. The best performing method such as Generative Adversarial Networks (GANs) [2] and Variational Autoencoders (VAEs) [3] tend to have a decoder component generally leading to intractable log-likelihood estimation. During training, GANs use a trick to forego optimizing log-likelihood directly. VAEs maximize the evidence lower bound (ELBO) but have an intractable marginal log-likelihood. Normalizing flows (NF) are a type of fully invertible generative models with tractable log-likelihood, allowing for direct optimization and evaluation of the log-likelihood. This makes it simpler to quantitatively compare the quality of models and is more robust against a problem common to GANs: model collapse [2].

The main goal of this document is provide an introduction into normalizing flow models particularly RealNVP [4] and it successor GLOW [1].

2. Background: Flow models

The goal of this section is to provide a brief overview of the core theory essential for understanding the flow models and their implementations in subsequent sections.

2.1. Continuous density function under change of variable

The *change of variable formula* from calculus [5] is at the core of normalizing flow models. Though the formula itself has more general application when integrating it can be interpreted in the context of rewriting continuous density functions as follows:

Definition 2.1. Given continuous random variables Z with density function $f_Z(z)$, and let $g : \mathbb{R} \to \mathbb{R}$ be a invertible monotonic function such that $x = g(z)$ and its inverse is equal to $z = g^{-1}(x)$. Then the random variable X that results from the mapping $x = g(z)$ will have a density function given by:

$$f_X(x) = f_Z\left(g^{-1}(x)\right)\left|\frac{d}{dx}\left(g^{-1}(x)\right)\right| . \tag{1}$$

In Equation 1 g is defined only for scalars, but the formula can also be applied to vectors $g : \mathbb{R}^n \to \mathbb{R}^n$. The last term of Equation 1 is replaced with the determinant of the Jacobian matrix of g^{-1} resulting in:

$$f_X(x) = f_Z\left(g^{-1}(x)\right)\left|\det\left(\frac{\partial g^{-1}(x)}{\partial x}\right)\right| . \tag{2}$$

When the last term $\left|\det\left(\frac{\partial g^{-1}(x)}{\partial x}\right)\right|$ is equal to 1, g is said to be *volume-preserving*.

3. Normalizing Flows

Approximations of complex distributions can be constructed by transforming a simple source distribution $p_Z(z)$ into a more complex target distribution $q_X(x)$ through a sequence of invertible transformations [6].

The source to target transformation can be described as a function composition of simple transformations $f_1 \dots f_k$. The density function with *flow* though the composed transformation and will be re-normalized to a valid density function after each application using the change of variable formula. For source $z \sim p_Z(z)$ and target $x \sim q_X(x)$ sample

the relation is as follows:

$$\mathbf{x} \overset{\mathbf{f}_1}{\longleftrightarrow} \mathbf{h}_1 \overset{\mathbf{f}_2}{\longleftrightarrow} \mathbf{h}_2 \cdots \overset{\mathbf{f}_K}{\longleftrightarrow} \mathbf{z}. \tag{3}$$

In this way, normalizing flows provide an explicit representation of the likelihood, unlike GANs or VAEs, which provided an implicit approximation of the likelihood that cannot be directly evaluated for individual samples. Note that the source-to-target transformation is also present in GANs and VAEs. For GANs the transformation is performed by the generator $G(z)$ and for VAEs by the decoder $p_\theta(x|z)$ but only flow models allow for direct likelihood evaluation.

3.1. Coupling

Assuming standard algorithms, computing the matrix determinant scales O(n^3) and quickly becomes a bottleneck. The feasibility of a flow model is tied to the computation the change of variable formula Equation 2 in each transformation; which includes computation of the Jacobian matrix determinant. In [7] it was shown that constructing the transformation as a triangular map leads to a lower triangular Jacobian with a tractable determinant. These mappings form the core of so-called coupling layers in normalizing flow models. The properties of a selection of these implementations will be discussed in the remainder of this section.

3.1.1 Real-valued Non-Volume Preserving (RealNVP)

RealNVP was introduced in 2017, it idiomatically stands for *real-valued non-volume preserving* (transformations), and introduces the concept of an *additive coupling layer* [4]. Additive coupling layers allow for the creation of complex yet cheaply invertible bijective functions through a design that can be compared with an auto-regressive model; they can be formalized as follows:

Definition 3.1. Given a D dimensional input and output vector x and y respectively, a dimension index $d \in \mathcal{Z}^+$ such that $d < D$. The affine coupling layer as per Equation 4, will *copy* the first d dimensions of the input without modification, thus $y_{1:d} = x_{1:d}$. The same copy operation is performed for the remaining $\{d-1 : D\}$ dimensions as per Figure 1, but these inputs are also scaled and translated with the Hadamard product of the first $\{1 : d\}$ dimensions (hence the affine).

$$\begin{cases} y_{1:d} & = x_{1:d} \\ y_{d+1:D} & = x_{d+1:D} \odot \exp\left(s\left(x_{1:d}\right)\right) + t\left(x_{1:d}\right) \end{cases} \tag{4}$$

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \frac{\partial y_{1:d}}{\partial x_{1:d}^T} & \frac{\partial y_{1:d}}{\partial x_{1+d:D}^T} \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \frac{\partial y_{d+1:D}}{\partial x_{d+1:D}^T} \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{bmatrix} \tag{6}$$

Figure 1-A illustrates the dependencies that now exists as a result of the forward propagation in the additive coupling layer. From now on x_1, x_2 will be used to refer to the first and second part of the vector $x_{1:d}, x_{d+1:D}$; likewise for vector y. The Jacobian can now be constructed in an elegant way that results in a lower triangular matrix Equation 6 with by tractable determinant. It is also worth pointing out that we do **not** need to compute the derivative of $\frac{\partial y_{d+1:D}}{\partial x_{1:d}^T}$ which is a great advantage as it allows for the choice of an arbitrarily complex function.

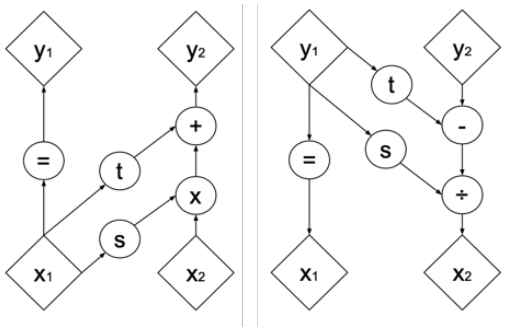


Figure 1: Shows the dependencies of forward (left) and inverse propagation (right) through an additive couple layer. Source: [4]

Invertibility is a condition for the transformations to feasible. Since x_1 has not been changed but merely copied into y_1 it can trivially be retrieved. Recovering x_2 is analogous to inverting the linear scale and translate operations on y_2 which can be done straightforward and cheaply using:

$$\begin{cases} x_{1:d} & = y_{1:d} \\ x_{d+1:D} & = \left(y_{d+1:D} - t\left(y_{1:d}\right)\right) \odot \exp\left(-s\left(y_{1:d}\right)\right) \end{cases}. \tag{7}$$

An observant reader might now have noticed that by composing transformations in this fashion only half of the values being modified. By permuting the order of the input after each transformation the issue is resolved in an efficient and invertible manner.

3.1.2 Non-Linear Independent Components Estimation (NICE)

Non-Linear Independent Components Estimation (NICE) [7] was introduced in 2015, and is an important paper as

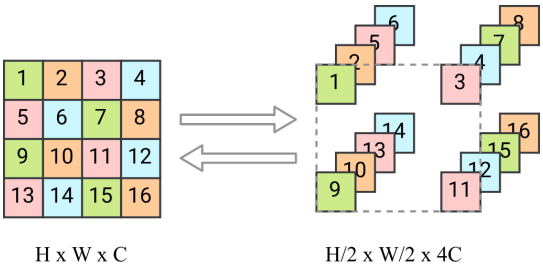


Figure 2: Example of the squeeze operations [8]

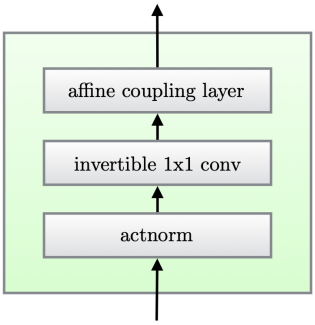


Figure 3: One step of the Glow model ([1])

it lays much of the groundwork for RealNVP. NICE introduces an additive coupling layer that is volume-preserving. Unlike RealNVP where scaling is applied directly in the affine coupling layer, NICE uses a separate scaling matrix.

3.2. Masking

The affine coupling layers discussed in section subsubsection 3.1.1 require the input to be split in parts. Naively one might choose to simply *cut* the image data in half. However, it is arguably more likely that two pixels directly next to each other are more correlated than two pixels in two different halves of an image. Therefore to create the mask for an image, RealNVP uses either a checkerboard pattern or channels wise splitting.

3.3. Multi-Scale Architecture

[9] shows that multi-scale architectures can allow for deeper networks. RealNVP [4] provides an implementation of a multi-scale architecture for normalizing flows using squeeze and split operations. The squeeze operation is simply a permutation of the input. Given an image with `C` channels, a subdivision of `2x2xC` subsquares is squeezed into `1x1x4C` tensors. Figure 2 shows an example of this operation on a `4x4` image.

The split operations will factor out half of the variables by directly mapping them to Gaussian latent space. Therefore, repeated application of the split operation will result in an exponential decrease in dimensionality, a significant computational advantage. The reduction in dimensionality and the increased number of parameter dimensions allow the model to learn more fine-grained features at the higher-level layers.

3.4. Glow: Generative Flow with Invertible 1x1 Convolutions

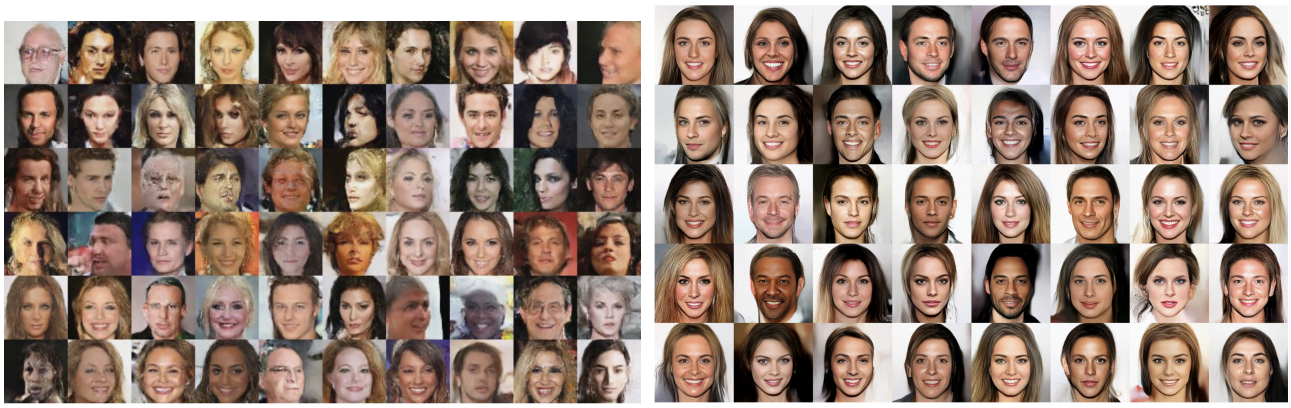
The Glow model builds on the methods introduced in [7, 4], primarily RealNVP's affine-coupling and multi-scale architecture. Glow's most notable architecture change is the addition of *invertible* 1x1 *convolution* in each step. These steps consist of three sequentially executed components as depicted in Figure 3; their roles can be summarized as follows:

- **actnorm** is a standard scale and bias layer which will initialize the input data to zero mean and unit-variance. After *data dependent* initialization the parameters in this layer become trainable and thus *data-independent*. RealNVP found that batch normalization [10] could lead to improved performance. But when using small mini-batches, batch normalization can lead to instability since the variance of its added noise is inversely proportional to the size of the mini-batch. Actnorm is an alternative to batch-normalization for use with mini-batches size of 1.
- **invertible 1x1 convolution** are convolutions with a filter of 1x1 for which the input and output dimension are identical and thus allowing for inversion. Convolutions with these characteristics can be seen as a generalization of permutation matrices with learnable parameters; making them an alternative to other permutations methods such as the checkerboard approach proposed by RealNVP.
- **affine coupling:** identical to affine coupling in RealNVP as described in Definition 3.1.

The three steps or Glow block can be stacked to create model complex models. Though the concept is simple, impressive results generating realistic new samples have been achieved Figure 4b. Currently, the state-of-the-art flows based models are less powerful as similar methods using GANs or VAE [2, 3]. Yet the latter statement is hard to quantify as comparing generative models is particularly challenge even between different instances of the same model.

4. Conclusion

With this document we provided an brief overview into normalizing flow models particularly those based on a Re-



(a) Sample from RealNVP ([4])

(b) Sample from Glow ([1])

Figure 4: Faces generated from two different flow models.

alNVP architecture [4]. We provided a review of the essential mathematical concepts, followed by an outline and review of prominent normalizing flow models: NICE [7], RealNVP [4] and Glow [1]. Illustrating that generative models based on normalizing flow are an alternative to state-of-the-art methods using VAE [3] or GAN based methods [2].

RealNVP [4] demonstrated that using the triangular bijections introduced in [7] can be extended with an additive coupling and permutation layers allowing the incorporation of arbitrary complexity function such as complex neural networks while maintaining a tractable Jacobian determinant. Moreover, since no sequential dependencies are present during sampling, efficient parallel sampling can be performed.

With the introduction of Glow [1] flow models can be used to generate realistic images. The quality of generated content in Figure 4 illustrates flows based models can be viable alternative to state-of-the-art methods using VAE [3] or GAN based methods [2].

Normalizing flows perform direct optimization of the likelihood. Though the observation might be trivial, it is worth considering that in practice models optimizing likelihood alternatives might be faster and obtain better results.

References

[1] D. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Advances in Neural Information Processing Systems*, vol. 2018-December. Neural information processing systems foundation, 2018, pp. 10 215–10 224, iSSN: 10495258. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85064849206&partnerID=40&md5=f172ad0bb5f1a59cbe2fcda18e1ff84d

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, Z. Ghahramani, M. Welling, and K. Weinberger, Eds., vol. 3. Neural information processing systems foundation, 2014, pp. 2672–2680, iSSN: 10495258 Issue: January. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84937849144&partnerID=40&md5=6441b2a288c5fdded2adbcc8b21e092c

[3] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2014. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083952489&partnerID=40&md5=3d871b7aba84e04275e4a97ba3b95934

[4] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2017. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088228644&partnerID=40&md5=d3e915598896065d028a3c9a9412b348

[5] J. Stewart, *Calculus : early transcendentals*. Belmont, Cal.: Brooks/Cole, Cengage Learning, 2012.

[6] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *32nd International Conference on Machine Learning, ICML 2015*, B. F. Blei D., Ed., vol. 2. International Machine Learning Society (IMLS), 2015, pp. 1530–1538. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84969776493&partnerID=40&md5=d24c3ecdb9bf0c6aecb7fcc25d0f62ae

[7] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," in *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*. International

Conference on Learning Representations, ICLR, 2015. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083952147&partnerID=40&md5=965a2a24262a3ebe78f53a3dd3f5cc55

[8] P. Lippe, "Tutorial 11: Normalizing Flows for image modeling — UvA DL Notebooks v1.0," Dec. 2020. [Online]. Available: https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial11/NF_image_modeling.html

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, arXiv: 1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning, ICML 2015*, B. D. Bach F., Ed., vol. 1. International Machine Learning Society (IMLS), 2015, pp. 448–456. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84969584486&partnerID=40&md5=bf7e146fe4da2c48a38050108068d697