Robust anomaly diagnosis with calibrated normalizing flows: What can we learn from a single failure?

Anonymous Authors[1]

Abstract

Applications like anomaly diagnosis (a.k.a post-mortem analysis) require learning from a very limited amount of data: only tens of data points from the anomaly, compared to hundreds or thousands of non-anomaly examples. Existing methods for solving inverse problems struggle in these data-constrained settings, often overfitting to noise in the limited data or underfitting due to an overly strong prior. We address this challenge with CALNF, a novel framework for posterior learning from limited data. We develop a training strategy inspired by random sample consensus, learning a family of densities on subsets of the training data before fine-tuning on the full dataset. CALNF achieves state-of-the-art performance on data-limited inference problems and enables a first-of-a-kind case study into the root causes of the 2022 Southwest Airlines scheduling crisis, providing new insights on the structural cause of this failure.

1. Introduction

Despite our best efforts, accidents happen. Autonomous vehicles crash, networks become congested, and predictions go awry. When things go wrong, we must be able to understand why in order to prevent future failures. This *anomaly diagnosis* task, sometimes referred to as *post-mortem analysis*, is challenging because we typically have only a handful of data points from the anomaly itself. While substantial effort has gone towards predicting failures with preemptive testing (often in simulation; Corso et al., 2022), relatively little work has been done on retrospective failure analysis, where only a limited amount of data is available.

Anomaly diagnosis can be framed as type of Bayesian inverse problem (IP), where we aim to infer the distribution of latent variables z from noisy observations x of a stochastic process $x \sim p(x|z;y)$, where y are known context variables (Stuart, 2010; Molinaro et al., 2023; Liu et al., 2023; Asim et al., 2020). In a traditional Bayesian IP setting, we are given one or more i.i.d. samples $\{y_i, x_i\}$, but in the anomaly diagnosis setting there is a *data imbalance* between a large number of samples $\mathcal{D}_n = \{y_i, x_i\}_{i=1,\dots,N_n}$ from nominal operations and a much smaller number of examples observed during the anomaly $\mathcal{D}_a = \{y_j, x_j\}_{j=1,\dots,N_a}$, where $N_a \ll N_n$. This setting is related to, but distinct from, out-of-distribution detection (where $\mathcal{D}_a$ is not known; Liang et al., 2018; Hendrycks et al., 2018; Kirichenko et al., 2020) and few-shot learning (where a pre-trained model adapted to a small number of samples; Wang et al., 2020).

Given these data, anomaly diagnosis aims to infer the *nominal distribution* $p(z|\mathcal{D}_n)$ and *anomaly distribution* $p(z|\mathcal{D}_a)$, conditioned on the nominal and anomaly datasets, respectively. Sampling from each of these distributions helps us understand what changes in the latent variables were associated with the observed anomaly (helping us ask "what went wrong"), while comparing the likelihoods of these distributions allows us to test for the presence of anomalies in future data. Unfortunately, imbalanced data in anomaly diagnosis problems makes it challenging to apply existing methods for solving inverse problems, which risk either overfitting to noise in the limited anomaly data or underfitting the anomaly in favor of the large nominal dataset.

Existing works on generative modeling from limited data attempt to strike this balance by regularizing the learned distributions (Abdollahzadeh et al., 2023; Asim et al., 2020; Higgins et al., 2016); unfortunately, it is difficult to specify an appropriate amount of regularization *a priori*, leading to tedious hyperparameter tuning.

In this paper, we address this gap by introducing CALNF, or calibrated normalizing flows. To make full use of available data, CALNF amortizes inference over both the nominal and anomaly data, learning a shared representation for both posteriors, but it prevents overfitting using a novel subsample-then-calibrate approach to learn an optimal representation for the anomaly posterior. In contrast to existing methods for regularized distribution learning, our method

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

does not require manual hyperparameter tuning, and it exceeds the performance of hand-tuned baselines on a range of challenging data-constrained inference problems.

To demonstrate the real-world applicability of our approach, we apply our method to a post-mortem analysis of the 2022 Southwest Airlines scheduling crisis, which stranded more than 2 million passengers during a winter storm and led to more than $750 million in financial losses (Rose, 2023). Our analysis provides new insights into the dynamics of the Southwest network and suggests that an imbalanced distribution of aircraft at key airports (other than those affected by the storm) may have contributed to the failure.

The paper is organized as follows. Section 2 provides relevant background on inverse problems and normalizing flows. Section 3 introduces CALNF, and Section 4 compares our approach to existing regularized inference methods on a range of benchmarks. Section 5 presents our main case study: a data-driven post-mortem analysis of the 2022 Southwest Airlines scheduling crisis. Section 6 concludes the paper and identifies directions for future work.

2. Background

2.1. Variational inference for Bayesian inverse problems

There is a large body of work dealing with IPs from a Bayesian perspective. Historically, Markov chain Monte Carlo (MCMC) methods have been the gold standard for posterior sampling, but the computational expense of MCMC in high dimensions has motivated the development of approximate algorithms like variational inference (VI; Stuart, 2010). These methods learn a variational guide that approximates the true posterior $q_\phi(z) \approx p(z|x;y)$ by maximizing the evidence lower bound (ELBO) on the dataset $\mathcal{D}$ (Kingma & Welling, 2013),

$$\mathcal{L}(\phi, \mathcal{D}) = \mathop{\mathbb{E}}_{(x,y)\in\mathcal{D}} \mathop{\mathbb{E}}_{z\sim q_\phi(z)} \left[\log \frac{p(x,z;y)}{q_\phi(z)}\right]. \quad (1)$$

2.2. Normalizing flows

$\mathcal{L}$ is maximized when the variational guide matches the true posterior $q_\phi(z) = p(z|x;y)$. Classical VI methods use simple representations for q_ϕ, such as independent Gaussians, which are often not capable of matching the true posterior, motivating the use of more flexible guides like normalizing flows (NFs). NFs represent q_ϕ as the transformation of a simple base distribution q_0 by an invertible mapping; e.g., $z = f_\phi(z_0)$, with $z_0 \sim \mathcal{N}(0, I)$ and a smooth bijection f_ϕ with inverse f_ϕ^{-1} (Tabak & Vanden-Eijnden, 2010; Rezende & Mohamed, 2015). We can sample from this distribution by passing samples from the base distribution through f, with exact likelihood given in terms of the Jacobian of f:

$$\log q_\phi(z) = \log q_0(f^{-1}(z)) - \log \left|\det J_f\left(f^{-1}(z)\right)\right|. \quad (2)$$

Normalizing flows have seen substantial success for image generation, density estimation, and inverse problems (Asim et al., 2020). Substantial effort has been devoted to developing flows based on different choices for f (Papamakarios et al., 2021; Grathwohl et al., 2018; Onken et al., 2021; Huang et al., 2018; Durkan et al., 2019). Our focus in this paper is not on proposing new architectures for f but rather on addressing the key challenge of training normalizing flows in data-constrained environments.

2.3. Anomaly detection

There is an important distinction between the inverse problems studied in this paper and prior works on anomaly detection. Anomaly detection aims to learn a binary classifier to *detect* anomalies, whereas our method aims to learn the posterior distribution of hidden states to *explain* an observed anomaly. Unsupervised anomaly detection methods, which do not require a system model or anomaly labels, are of independent interest, but they do not provide any insights into the distribution of latent parameters (Gudovskiy et al., 2022; Kumar et al., 2021; Najari et al., 2022). In Section 4, we discuss the downstream benefits of improved posterior inference on supervised anomaly detection.

3. Method: Calibrated Normalizing Flows

The key challenge in applying existing VI methods, including those using normalizing flows, to our setting is the imbalance in the size of the nominal and failure datasets. Relying solely on anomaly data risks overfitting to noise in those data, but using both datasets risks underfitting the anomaly in favor of the much larger nominal dataset.

Existing methods attempt to resolve this issue by first learning the nominal posterior, then using it as a prior to regularize the anomaly posterior. This is commonly done by training q_{ϕ_n} on nominal data alone, then learning q_{ϕ_a} subject to a penalty on divergence from the nominal distribution (Asim et al., 2020; Higgins et al., 2016); for example, $\phi_n = \arg\max_\phi \mathcal{L}(\phi, \mathcal{D}_n)$ and $\phi_a = \arg\max_\phi \mathcal{L}(\phi, \mathcal{D}_a) - \beta D(q_\phi, q_{\phi_n})$, where β is a hyperparameter that controls how close the anomaly posterior is to the nominal distribution and D is a function that measures the divergence between two distributions (common choices include the Kullback-Leibler divergence $D = D_{KL}$ and the Wasserstein metric $D = W_2$). The main challenge with this approach is that β can be difficult to tune. As we show in Fig. 1, too little regularization results in overfitting to noise in the sparse data, while too much makes it difficult to distinguish between the nominal and anomalous cases. There is no clear choice for how much regularization is appropriate, and so it must be tuned manually, leaving substantial room for error.

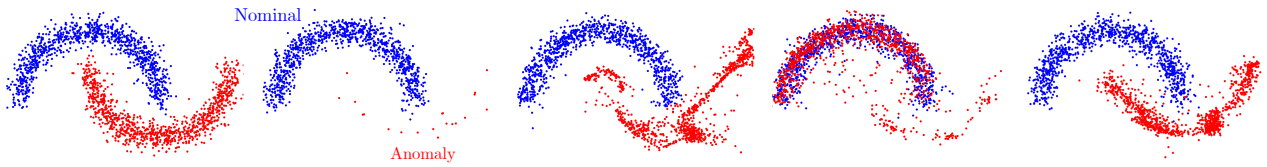

Figure 1. **Illustrating the effect of data imbalance.** (a) The ground truth distribution. (b) An imbalanced dataset. (c) When the regularization strength β is too small, existing methods overfit to noise in the anomaly dataset. (d) When β is too large, the learned distribution underfits the anomaly and struggles to distinguish between nominal and anomalous data. (e) Our method yields a more accurate reconstruction of the anomaly distribution by constraining the divergence between the nominal and anomaly distributions.

To address this challenge, we propose the calibrated normalizing flow, or CALNF, framework. Instead of learning the anomaly distribution directly, we train a normalizing flow to represent a family of possible distributions, then tune the model to select the optimal anomaly distribution from this family. This approach allows us to efficiently learn the posterior without overfitting.

3.1. Overview

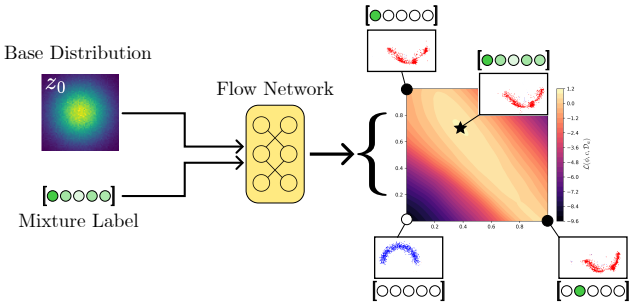

Figure 2. **CalNF architecture**: A normalizing flow is trained on random subsets of the anomaly data and the full nominal dataset, using one-hot labels to identify different subsets (●) and the zero vector to identify the nominal data (○). The model is calibrated by optimizing the label to find a posterior distribution that best explains the entire anomaly training dataset (★).

CALNF takes inspiration from robust regression algorithms like RANSAC (Fischler & Bolles, 1981), randomly sampling K subsets of the anomaly data $\mathcal{D}_a^1, \ldots, \mathcal{D}_a^K$ and using a conditional flow $q_\phi(z; c)$ to learn a posterior for each, identifying the different subsets with one-hot labels $c_i = \mathbf{1}_i$:

$$q_\phi(z; \mathbf{1}_i) \approx p(z|\mathcal{D}_a^i), \; i = 1, \ldots, K$$
$$q_\phi(z; \mathbf{0}_K) \approx p(z|\mathcal{D}_n),$$

where the zero label $c = \mathbf{0}_K$ is used to identify the nominal dataset. Once posteriors have been learned for each of these subsets, we calibrate the model by finding an optimal mixture of these posteriors to explain the full anomaly dataset; i.e. holding the model weights ϕ constant and finding the optimal label c^* such that $q_\phi(z; c^*) \approx p(z|\mathcal{D}_a)$.

This two-step process is illustrated in Fig. 2. On an intuitive level, our approach learns a family of anomaly posteriors parameterized by the low-dimensional label c, then optimizes in the lower-dimensional label space to find a good estimate of the overall anomaly posterior, as shown in Fig. 2. Examples of the individual posteriors are shown in Fig. 3.

It is important to note that CALNF is agnostic to the specific architecture chosen for normalizing flow (e.g. the form of f_ϕ); our main contribution is the higher-level framework for training the model in the context of sparse anomaly data, which we discuss in more detail in the following sections.

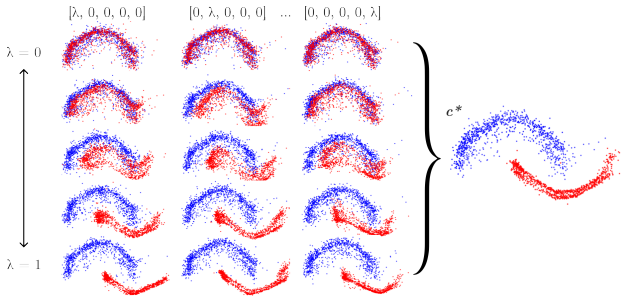

Figure 3. **Uncalibrated vs. calibrated posteriors.** (Left) The family of distributions learned on subsamples. Red and blue points are samples from the nominal $q_\phi(z; \mathbf{0})$ and anomaly posteriors $q_\phi(z; \lambda \mathbf{1}_i)$ for $\lambda \in [0, 1]$, respectively. Individual posteriors are overfit to their respective subsets, but the calibrated posterior (right) fits well across the full anomaly dataset.

3.2. Theoretical motivation

The main idea of CALNF is to learn a single model with shared parameters for the nominal and anomaly posteriors. In this section, we provide theoretical motivation for this decision, showing that learning a single model to encode different posteriors via the low-dimensional label c *implicitly* regularizes the learned posteriors. Moreover, our analysis shows by optimizing the anomaly label c^*, CALNF is effectively finding an optimal amount of implicit regularization for the anomaly posterior.

In particular, consider the Wasserstein metric $W_2(p_1, p_2) = \inf_\gamma \left[\mathbb{E}_{z_1, z_2 \sim \gamma} ||z_1 - z_2||^2\right]^{1/2}$ where γ is a coupling of probability distributions p_1 and p_2; i.e., γ is a joint dis-

tribution over z_1 and z_2 with p_1 and p_2 as its corresponding marginals. The following remark and theorem show that CALNF provides implicit regularization of the W_2 metric between the learned nominal and anomaly posteriors.

Remark 1. The map $f_\phi(z;c)$ is L-Lipschitz in many flow architectures; i.e. there exists L such that $|f_\phi(z;c_1) - f_\phi(z;c_2)| \leq L||c_1 - c_2|| \; \forall \; z$ (Verine et al., 2023).

Theorem 2. *The Wasserstein distance between the nominal and anomaly posteriors, $q_\phi(z, \mathbf{0}_K)$ and $q_\phi(z;c^*)$ respectively, is bounded by $W_2\left(q_\phi(z,\mathbf{0}_K), q_\phi(z,c^*)\right) \leq L||c^*||$ so long as the underlying flow map $f_\phi(z,c)$ is L-Lipschitz in the second argument.*

A proof is included in the appendix, along with L for common flows. This result suggests that CALNF first learns a family of possible anomaly posteriors $q_\phi(z;\mathbf{1}_i)$ without divergence constraints (since L is often large), then the calibration process finds the optimized c^* (and corresponding divergence bound) that leads to the best explanation of the full anomaly dataset.

3.3. Training

The CALNF model, together with the optimized label, can be trained using Algorithm 1. This algorithm modifies the standard variational inference training process in two ways: by training on multiple random subsets of the anomaly data, and by interleaving model updates and label calibration.

First, we split the anomaly training data into K random subsets with one-hot labels and train the model to learn the posterior for each subset. Each subset $\mathcal{D}_a^i$ is created by independently drawing $\lfloor N_a/2 \rfloor$ samples from $\mathcal{D}_a$ without replacement. We denote the ELBO on a given dataset $\mathcal{D}$ as

$$\mathcal{L}\left(\phi, c, \mathcal{D}\right) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \mathop{\mathbb{E}}_{z\sim q_\phi(z;c)} \left[\log \frac{p(x,z;y)}{q_\phi(z;c)}\right]. \quad (3)$$

The model parameters are updated to maximize the sum of several ELBOs: for each anomaly subset (with one-hot labels), for the nominal dataset (with a zero label), and for the full anomaly dataset (with the calibrated label c):

$$L_a(\phi) = -\frac{1}{K}\sum_{i=1}^{K} \mathcal{L}\left(\phi, \mathbf{1}_i, \mathcal{D}_a^i\right), \quad (4)$$

$$L_n(\phi) = -\mathcal{L}\left(\phi, \mathbf{0}_K, \mathcal{D}_n\right), \quad (5)$$

$$L_{cal}(\phi, c) = -\mathcal{L}(\phi, c, \mathcal{D}_a) \quad (6)$$

This leads to the overall loss,

$$L(\phi, c) = L_a(\phi) + L_n(\phi) + L_{cal}(\phi, c). \quad (7)$$

The mixture label c is initialized at $[1/K, \ldots, 1/K]$ and updated to minimize $L_{cal}(\phi, c)$. In practice, we find that we can interleave optimization for ϕ and c.

Algorithm 1 Calibrated Normalizing Flows

Input: Nominal data $\mathcal{D}_n$, anomaly data $\mathcal{D}_a$, step size γ, number of anomaly subsamples K
Output: Model parameters ϕ and calibrated label c^*
for $k = 1, \ldots, K$ **do**
  $\mathcal{D}_a^k \leftarrow \lfloor N_a/2 \rfloor$-element random subset of $\mathcal{D}_a$
end for
Initialize ϕ, c
while ϕ not converged **do**
  Compute $L = L_a(\phi) + L_n(\phi) + L_{\text{cal}}(\phi, c)$
  Update model $\phi \leftarrow \phi + \gamma \nabla_\phi L$
  Update calibration $c \leftarrow c + \gamma \nabla_c L_{\text{cal}}(\phi, c)$
end while

4. Experiments

4.1. Benchmark problems

This section introduces the data-constrained anomaly diagnosis problems used in our experiments. More details are provided in the appendix, and we will open-source code and data for each upon publication. The first benchmark is newly developed for our case study, but the second and third are previously-published benchmark problems (Keipour et al., 2021; Deng et al., 2022). We also include the toy 2D problem in Fig. 1 ($N_n = 1000$, $N_f = 20$).

Air traffic disruptions We develop a stochastic queuing model of the Southwest Airlines network using actual flight arrival and departure data published by the US Bureau of Transportation Statistics (Bureau of Transportation Statistics)[1]. This model accounts for the movement of aircraft between airports, uncertain travel times and air traffic control (ATC) delays, runway congestion, and varying aircraft reserves at each airport. We base our model on that in Pyrgiotis et al. (2013), with extensions for aircraft reserves. The latent variables represent travel times between airports, runway delays at each airport, and the number of aircraft stationed at each airport at the start of the day. The context includes the scheduled departures and arrivals for the day, and the observations include the actual departure and arrival time for each flight. The nominal and anomaly datasets are taken from Dec. 1 through Dec. 20 and Dec. 21 through Dec. 30, respectively. For benchmarking, we consider only the four busiest airports, but we consider larger sub-networks in our case study in Section 5. The four-airport sub-network has 24 latent variables. We train on $N_n = 9$ and $N_f = 4$ data points and evaluate on 4 anomalous data points (each data point is a single day with between 88–102 flights).

Geophysical imaging Seismic waveform inversion (SWI) is a well-known geophysics problem used as a benchmark

[1] https://www.transtats.bts.gov/

for inference and physics-informed learning (Gouveia & Scales, 1998; Deng et al., 2022; Zhang et al., 2016). SWI seeks to infer the properties of the Earth's subsurface using seismic measurements, which are simulated by solving the elastic wave partial differential equation (PDE) numerically. This model uses latent variables z for subsurface density, context y for the source signal, and observations x for the seismic measurements (Richardson, 2023). The latent space has 100 dimensions (a 10×10 grid). We train using $N_n = 100$ and $N_a = 4$ and evaluate on 500 synthetic samples.

Aerial vehicle control We consider a failure detection benchmark for unmanned aerial vehicles (UAVs) using the ALFA dataset (Keipour et al., 2021). This dataset includes real-world data from a UAV during normal flight and during failures where various control surfaces are deactivated. z parameterizes the nonlinear attitude dynamics, y includes the current state and desired orientation, and x is the next state. The latent space has 21 dimensions; we train on 10 nominal trajectories with $N_n = 2235$ and 1 anomalous trajectory with $N_a = 58$, and we evaluate on a second anomalous trajectory with 69 data points.

4.2. Baselines and Metrics

Our main claim is that our CALNF framework is an effective way to learn the posterior when a small number of anomaly data points are available. As a result, the most relevant comparisons are to methods for posterior learning with dataset bias, which typically involve regularizing the learned posterior. In particular, we compare against two baselines: a "state-of-the-practice" method regularizing the KL divergence (Asim et al., 2020; Higgins et al., 2016) and a state-of-the-art method specific to normalizing flows that regularizes the Wasserstein distance W_2. This second method follows RNODE and related works by penalizing the squared norm of the vector field of a continuous normalizing flow (Finlay et al., 2020; Onken et al., 2021). We implement the KL-regularized method using neural spline flows (Durkan et al., 2019) and label this method β-NSF. Since each of these baselines relies on a hyperparameter to determine the strength of the regularization (β for KL regularization and λ_K for RNODE), we provide results for a range of hyperparameters.

Since the relatively large amount of nominal data makes it easy to fit the nominal distribution, we compare primarily on the basis of the evidence lower bound $\mathcal{L}$ computed on held-out anomaly data. It is important to note that while our method requires less hyperparameter tuning than the other methods, it requires additional likelihood evaluations to fit the subsampled anomaly data. To quantify this trade-off, we report the training time for all methods. All metrics report the mean and standard deviation over four random seeds. When useful, we also provide visual comparisons of the posterior distributions learned using different methods.

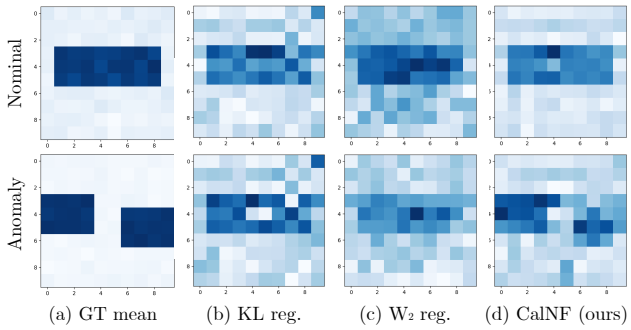

Figure 4. **Seismic waveform inversion.** (a) The ground truth nominal and anomalous density profiles. (b-d) The posteriors fit using KL and W_2 regularization and CALNF. CALNF is the only method to correctly infer the anomaly density profile.

4.3. Results & Discussion

Our main empirical results are shown in Table 1. Our method achieves better performance on held-out anomaly data than baselines on all problems; moreover, our method does not require manual hyperparameter tuning ($K = 5$ was sufficient for all problems). CALNF's improved performance comes at the cost of increased training time, requiring K additional likelihood evaluations per step; this difference is most significant on the SWI and ATC problems, where evaluating the likelihood is particularly expensive. On problems where the likelihood is easy to evaluate, the RNODE-derived methods are slowest to train due to their use of neural ODEs.

To understand the difference in performance, Fig. 4 compares the learned anomaly posteriors on the SWI example with the ground truth in 4a. We see that the KL- and W_2-regularized methods (4b and 4c, respectively) do not infer the correct density profile in the anomaly case, while our method (4d) is able to infer the correct shape. This suggests that our method is able to appropriately balance the information gained from the nominal distribution with the limited number of anomaly data points.

We also provide the results of an ablation study in Table 2, comparing the ELBO achieved when we omit the calibration step (using a constant c), omit the nominal data, and remove the subsampling step. These results indicate that most of the performance improvement from CALNF is due to training on random subsamples of the anomaly data. We observe that in cases with plentiful nominal data (like the UAV problem), including the L_n term also substantially boosts performance. We find that the benefit of optimizing c is relatively minor, but this step can be included for little additional computational cost. Results on the sensitivity of CALNF to varying K are included in appendix.

Table 1. ELBO (nats/dim) on held-out anomaly data and training times (in minutes) on benchmark problems. 2D and SWI use synthetic data, so additional anomaly data were generated for the test set; in all other cases, half of the anomaly data was withheld for testing. Mean and standard deviation across four seeds are reported. †scaled by $\times 10^{-3}$/

	2D NATS/DIM ↑	SWI NATS/DIM† ↑	UAV NATS/DIM ↑	ATC NATS/DIM† ↑
β-NSF ($\beta = 0.01$)	$-3.22_{\pm 0.13}$	$43.8_{\pm 0.61}$	$3.30_{\pm 0.83}$	$-2.33_{\pm 0.05}$
β-NSF ($\beta = 0.1$)	$-2.03_{\pm 0.04}$	$43.9_{\pm 0.79}$	$3.64_{\pm 1.27}$	$-2.30_{\pm 0.05}$
β-NSF ($\beta = 1.0$)	$-1.04_{\pm 0.06}$	$44.1_{\pm 0.84}$	$2.78_{\pm 1.71}$	$-2.12_{\pm 0.09}$
RNODE ($\lambda_K = 0.01$)	$-4.58_{\pm 0.18}$	$36.0_{\pm 3.14}$	$0.76_{\pm 2.31}$	$-4.36_{\pm 1.02}$
RNODE ($\lambda_K = 0.1$)	$-2.95_{\pm 0.14}$	$36.0_{\pm 3.13}$	$0.76_{\pm 2.28}$	$-4.39_{\pm 1.08}$
RNODE ($\lambda_K = 1.0$)	$-1.67_{\pm 0.05}$	$36.0_{\pm 3.06}$	$1.14_{\pm 2.50}$	$-4.35_{\pm 1.04}$
CALNF (OURS)	$\mathbf{-0.90}_{\pm 0.10}$	$\mathbf{46.3}_{\pm 0.18}$	$\mathbf{6.95}_{\pm 1.24}$	$\mathbf{-2.01}_{\pm 0.10}$
	TIME ↓	TIME ↓	TIME ↓	TIME ↓
β-NSF ($\beta = 0.01$)	$0.43_{\pm 0.02}$	$33.5_{\pm 0.2}$	16.9 ± 0.09	$\mathbf{81.6}_{\pm 9.2}$
β-NSF ($\beta = 0.1$)	$0.45_{\pm 0.03}$	$33.6_{\pm 0.2}$	17.0 ± 0.08	$\mathbf{81.7}_{\pm 8.5}$
β-NSF ($\beta = 1.0$)	$0.45_{\pm 0.03}$	$33.6_{\pm 0.1}$	16.9 ± 0.28	$\mathbf{81.4}_{\pm 8.7}$
RNODE ($\lambda_K = 0.01$)	$5.37_{\pm 0.17}$	$\mathbf{25.1}_{\pm 0.5}$	68.0 ± 2.98	$\mathbf{82.0}_{\pm 8.4}$
RNODE ($\lambda_K = 0.1$)	$5.38_{\pm 0.19}$	$\mathbf{25.1}_{\pm 0.7}$	67.5 ± 3.60	$\mathbf{82.2}_{\pm 7.6}$
RNODE ($\lambda_K = 1.0$)	$5.23_{\pm 0.06}$	$\mathbf{24.9}_{\pm 0.7}$	69.7 ± 12.6	$\mathbf{81.8}_{\pm 8.7}$
CALNF (OURS)	$0.53_{\pm 0.02}$	$80.1_{\pm 0.5}$	45.9 ± 0.32	$148.8_{\pm 16.5}$

Table 2. ELBO (nats/dim) on held-out anomaly data for ablations of CALNF. The first is our proposed method, the second fixes c, the third excludes the nominal data during training, and the fourth does not subsample the anomaly data. †scaled by $\times 10^{-3}$

	2D	SWI†	UAV	ATC†
CALNF	$\mathbf{-0.90}_{\pm 0.1}$	$\mathbf{46.3}_{\pm 0.2}$	$6.95_{\pm 1.2}$	$\mathbf{-2.01}_{\pm 0.1}$
W/O c^*	$-0.96_{\pm 0.2}$	$46.2_{\pm 0.4}$	$\mathbf{7.86}_{\pm 1.0}$	$-2.02_{\pm 0.1}$
W/O L_n	$-1.12_{\pm 0.2}$	$46.1_{\pm 0.4}$	$-9.22_{\pm 10}$	$-2.03_{\pm 0.2}$
W/O $\mathcal{D}_a^i$	$-1.03_{\pm 0.2}$	$43.9_{\pm 2.8}$	$-3.65_{\pm 11}$	$-2.05_{\pm 0.1}$

Table 3. Performance of CALNF and baselines for anomaly detection, reporting area under the receiver operating characteristic curve (AUROC); mean and standard deviation across four seeds are reported, higher is better. Additional metrics are given in Table 8.

	2D	SWI	UAV
NF-AD	$\mathbf{1.00}_{\pm 0.001}$	$0.74_{\pm 0.030}$	$\mathbf{0.60}_{\pm 0.16}$
NF-AD$_{\text{KL}}$	$\mathbf{1.00}_{\pm 0.001}$	$0.74_{\pm 0.030}$	$\mathbf{0.71}_{\pm 0.15}$
NF-AD$_{W_2}$	$\mathbf{1.00}_{\pm 0.002}$	$0.65_{\pm 0.033}$	$0.54_{\pm 0.077}$
CALNF	$\mathbf{1.00}_{\pm 0.005}$	$\mathbf{0.79}_{\pm 0.022}$	$\mathbf{0.70}_{\pm 0.16}$

4.4. Using CALNF for anomaly detection

Table 3 includes results from adapting CALNF to anomaly detection by using the learned posterior to classify previously-unseen observations. We compare with supervised anomaly detectors based on normalizing flows, both as proposed in (Gudovskiy et al., 2022; Kang et al., 2022; Rudolph et al., 2021) and using KL and W_2 regularization with hand-tuned penalties. We were not able to test anomaly detection on the ATC example due to the small number of evaluation points. We find that the improved posterior learned using CALNF leads to anomaly detection performance that meets or exceeds all existing methods.

5. Case Study: 2022 Southwest Airlines Scheduling Crisis

In this section, we apply our method to a post-mortem analysis of the 2022 Southwest Airlines scheduling crisis. In the period between December 21st and December 30th, 2022, a series of cascading delays and cancellations severely disrupted the Southwest network, starting in Denver and spreading across the United States. The disruption occurred in roughly two stages, as shown in Fig. 11 in the appendix. In the first stage, from 12/21 to 12/24, weather and operational difficulties caused cancellations to increase from a $< 5\%$ baseline to over 50% of scheduled flights. In the second phase, after trying and failing to recover normal operations, Southwest flight dispatchers started preemptively canceling flights and ferrying crew between airports to reset the network, canceling up to 77% of scheduled flights between 12/25 and 12/29 before returning to near-normal operations on 12/30. Southwest ultimately canceled more than 16,000 flights, affecting more than 2 million passengers, and the airline later paid a $140 million penalty imposed by the US Department of Transportation (28% of its 2023 net income; Rose, 2023) in addition to lost revenue.

This incident has been the subject of extensive investigation, with a report from Southwest Airlines (Southwest Airlines,

2023), testimony before the US Senate from the Southwest Airlines Pilots Association (SWAPA; Murray, 2023), and press coverage (Rose, 2023; Cramer & Levenson, 2022). These sources propose a number of hypotheses on the root cause of the 2022 incident. While there is broad agreement that winter weather was a major factor, sources differ on the role of other factors; e.g. the SWAPA report emphasizes poor crew management, while press coverage emphasizes the point-to-point nature of the Southwest network.

Given this context, we have two goals for our case study. First, we are interested in identifying changes in the network state that coincided with the disruption, and how those disrupted parameters compare to the nominal state of the network. Second, we aim to produce a generative model of the nominal and disrupted network conditions to act as a tool for network design and analysis, so that future operational, scheduling, and recovery policies might be proactively stress-tested.

5.1. Implementation

Due to the difficulty of modeling the decision-making process of the Southwest flight dispatchers during the second half of the disruption, we focus on the first four days of the scheduling crisis, prior to the wave of cancellations aimed at resetting the network. We conduct our analysis at multiple levels of spatial resolution, looking at both the top-4 and top-10 subnetworks that include only flights between the 4 and 10 busiest airports in the Southwest network, respectively. More details on the network model are included in the appendix, along with a key for relevant three-letter airport codes in Table 6.

5.2. Results

Localized delays due to winter weather. Our first observation confirms a common explanation for the disruption: that localized delays at airports across the US coincided with winter weather. For example, Fig. 5 shows CALNF's posterior estimates of nominal and disrupted service times, which include taxiing, deicing, and ATC delays, at the four busiest airports. Of these four, DEN, MDW, and DAL, which saw severe cold temperatures, experienced a 50% increase in average service time, while there was no corresponding increase at LAS, which did not experience severe weather. This result agrees with press and official accounts that identify winter weather and a lack of deicing equipment at critical airports like DEN as a contributing factor (Southwest Airlines, 2023; Cramer & Levenson, 2022). However, the more important question is how these localized service delays cascaded into the nationwide disruption.

Cascading failures due to aircraft flow interruption. Our main finding comes from modeling the movement of aircraft within the Southwest network. The number of aircraft that start the day at each airport provides an important measure of robustness, since if there are insufficient aircraft to meet demand, then departing flights must be delayed or canceled.[2] A lack of aircraft can also cause cancellations to cascade through the network if down-stream airports are deprived of the aircraft needed to serve scheduled departures. Despite its importance, aircraft distribution is not directly observable from public data, and so it must be inferred.

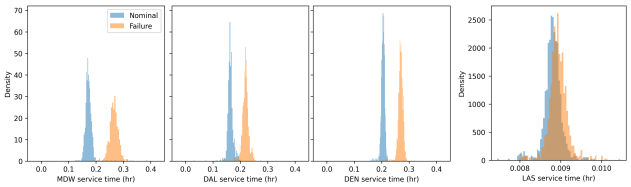

Figure 5. The posterior distribution indicates that service times (including taxiing, de-icing, and ATC delays) increased at DEN, MDW, and DAL, which were hit by a winter storm, but were unchanged at LAS, which did not see severe weather.

Fig. 6 shows our results from using CALNF to infer the distribution of aircraft reserves in the top-10 network over each of the first four days of the disruption. CALNF finds that there was no detectable deviation from the nominal aircraft distribution on the first day of the disruption, but CALNF detects a steadily increasing deficit at LAS, DAL, and PHX over the following three days. The fact that the aircraft deficit at these airports continued to worsen may have been a factor in Southwest's decision to "hard reset" the network by ferrying empty planes between airports.

Fig. 7 provides an analysis of the Southwest network structure suggesting a mechanism by which the aircraft deficits at LAS, DAL, and PHX propagated to the rest of the network. During normally scheduled operations, LAS, PHX, and DEN receive nearly 50% of their last-leg flights (i.e. aircraft that park overnight) from either DEN or MDW (orange segments in Fig. 7a). Together, LAS, DEN, MDW, DAL, and PHX host the five largest overnight reserves in the network, so a large majority of flights in the Southwest network involve routes passing through one of these airports (red segments in Fig. 7). Together with our finding with CALNF in Fig. 6 that overnight reserves at LAS/DAL/PHX were depleted, this coupling suggests a possible mechanism for the 2022 disruption where winter weather caused cancellations at DEN/MDW, leading first to depleted overnight reserves at LAS/DAL/PHX and then to an imbalanced aircraft distribution throughout the network.

Our analysis suggests that LAS, PHX, and DAL may have played a key role in allowing the disruption to spread

[2]The same logic holds for the crew distribution. Our model assumes that crews and aircraft move together, but a separate crew model with duty time limits would be an important extension.

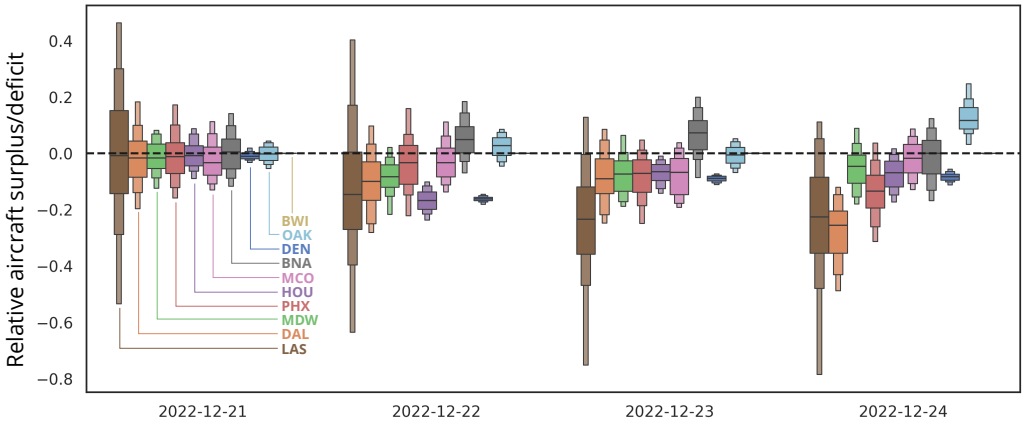

Figure 6. CALNF's inferred posterior estimates of the distribution of Southwest aircraft at the start of the first four days of the disruption, normalized by the number of scheduled departures at each airport; positive/negative indicates more/fewer aircraft than in the nominal case, respectively. CALNF suggests that LAS, DAL, and PHX accumulated a large aircraft deficit over the course of the disruption.

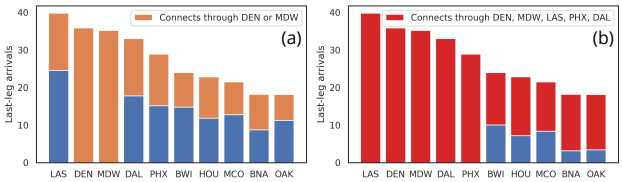

Figure 7. (a) Data from normal operations show that LAS, DEN, MDW, DAL, and PHX have the largest number of "last leg" arrivals, indicating that a large number of aircraft park at these airports overnight. Nearly half of aircraft parked overnight at LAS, DAL, and PHX travel through DEN or MDW (orange segments), suggesting a mechanism where winter weather at DEN/MDW leads to the depleted reserves that CALNF predicts in Fig. 6 at LAS/DAL/PHX. (b) The large majority of normally-scheduled flights connect through either DEN/MDW or LAS/DAL/PHX (red segments), providing a mechanism for the depleted aircraft reserves at LAS/DAL/PHX to propagate to the rest of the network

throughout the Southwest network, even though LAS and PHX did not experience winter weather. Our results indicate that trends in aircraft reserves at these airports may be a valuable warning sign for detecting future disruptions.

Generative modeling Once we have learned the nominal and disruption posteriors for the Southwest network, we can use these as generative models for stress-testing proposed modifications to the Southwest network or scheduling system. In future work, we hope to explore how these generative models can be used to design more resilient schedule recovery algorithms.

6. Conclusion

In this paper, we propose a novel algorithm for data-constrained posterior inference, which uses a subsampling and calibration strategy to avoid overfitting to sparse data. We apply our algorithm to anomaly diagnosis problems, achieving competitive performance on challenging inverse problem benchmarks with both simulated and real data. We also apply our algorithm to a real-world anomaly diagnosis problem, providing new insight into the factors behind the 2022 Southwest Airlines scheduling crisis.

Limitations & future work There are two major limitations of our work, which indicate directions for future research. First, although we include preliminary results showing the benefit of our posterior inference method for downstream tasks, more work is needed to understand how best to apply the posterior distributions learned using our method for online anomaly detection. In particular, it has been reported that normalizing flows are prone to assigning high likelihoods to out-of-distribution samples, especially in higher dimensions (Kirichenko et al., 2020). As a result, more work is needed before our method can be used to detect previously-unseen anomalies, building on existing out-of-distribution detection techniques.

Second, our method does not provide an estimate of the risk of an anomaly. Estimating the probability of failure is challenging due to the size of the dataset, but we hope that future work will close this gap; e.g. through the application of large deviation theory (Dembo & Zeitouni, 2010).

Finally, we hope to further explore how the nominal and anomaly posterior distributions can be used as generative models for developing improved control algorithms; for example, using the learned model of UAV failures to optimize a flight controller, or using the learned model of the Southwest Airlines disruption to design improved scheduling and recovery algorithms.

Impact statement

This paper deals with the problem of anomaly diagnosis with the goal allowing system designers to understand the root causes of past failures and to prevent future incidents. If successful, we hope that our work will help enable a more comprehensive data-driven approach to safety analysis for complex systems, including cyberphysical systems and complex infrastructural networks.

While we do not explicitly deal with dual-use applications like adversarial testing in this paper, we acknowledge the potential for a generative model trained using our approach to be used in an attempt to induce failure in the system under test. In such cases, we note that the system designer can use this capability to develop more robust designs, reducing the possibility for harm.

References

Zuko (open-source library). The Probabilists, January 2024.

Abdollahzadeh, M., Malekzadeh, T., Teo, C. T. H., Chandrasegaran, K., Liu, G., and Cheung, N.-M. A Survey on Generative Modeling with Limited Data, Few Shots, and Zero Shot, July 2023.

Asim, M., Daniels, M., Leong, O., Ahmed, A., and Hand, P. Invertible generative models for inverse problems: Mitigating representation error and dataset bias. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 399–409. PMLR, November 2020.

Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J.-H. Invertible Residual Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 573–582. PMLR, May 2019.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, January 2019. ISSN 1532-4435.

Bureau of Transportation Statistics. TranStats. U.S. Department of Transportation.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Chen, R. T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. Residual Flows for Invertible Generative Modeling. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Corso, A., Moss, R., Koren, M., Lee, R., and Kochenderfer, M. A Survey of Algorithms for Black-Box Safety Validation of Cyber-Physical Systems. *Journal of Artificial Intelligence Research*, 72:377–428, January 2022. ISSN 1076-9757. doi: 10.1613/jair.1.12716.

Cramer, M. and Levenson, M. What Caused the Chaos at Southwest. *The New York Times*, December 2022. ISSN 0362-4331.

Dembo, A. and Zeitouni, O. *Large Deviations Techniques and Applications*, volume 38 of *Stochastic Modelling and Applied Probability*. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-03310-0 978-3-642-03311-7. doi: 10.1007/978-3-642-03311-7.

Deng, C., Feng, S., Wang, H., Zhang, X., Jin, P., Feng, Y., Zeng, Q., Chen, Y., and Lin, Y. OpenFWI: Large-scale Multi-structural Benchmark Datasets for Full Waveform Inversion. In *Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, June 2022.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 675, pp. 7511–7522. Curran Associates Inc., Red Hook, NY, USA, December 2019.

Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. How to train your neural ODE: The world of Jacobian and Kinetic regularization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pp. 3154–3164. JMLR.org, July 2020.

Fischler, M. A. and Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692.

Gouveia, W. P. and Scales, J. A. Bayesian seismic waveform inversion: Parameter estimation and uncertainty analysis. *Journal of Geophysical Research: Solid Earth*, 103(B2):2759–2779, 1998. ISSN 2156-2202. doi: 10.1029/97JB02933.

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models. In *International Conference on Learning Representations*, September 2018.

Gudovskiy, D., Ishizaka, S., and Kozuka, K. CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows. In *2022*

IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1819–1828. IEEE Computer Society, January 2022. ISBN 978-1-66540-915-5. doi: 10.1109/WACV51458.2022.00188.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep Anomaly Detection with Outlier Exposure. In *International Conference on Learning Representations*, September 2018.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*, November 2016.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural Autoregressive Flows. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2078–2087. PMLR, July 2018.

Kang, Z., Mukhopadhyay, A., Gokhale, A., Wen, S., and Dubey, A. Traffic Anomaly Detection Via Conditional Normalizing Flow. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2563–2570, Macau, China, October 2022. IEEE Press. doi: 10.1109/ITSC55140.2022.9922061.

Keipour, A., Mousaei, M., and Scherer, S. ALFA: A dataset for UAV fault and anomaly detection. *The International Journal of Robotics Research*, 40(2-3):515–520, February 2021. ISSN 0278-3649. doi: 10.1177/0278364920966642.

Kingma, D. P. and Dhariwal, P. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations*, December 2013.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Why Normalizing Flows Fail to Detect Out-of-Distribution Data. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20578–20589. Curran Associates, Inc., 2020.

Kumar, N., Hanfeld, P., Hecht, M., Bussmann, M., Gumhold, S., and Hoffmann, N. InFlow: Robust outlier detection utilizing Normalizing Flows, November 2021.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Liu, T., Yang, T., Zhang, Q., and Lei, Q. Optimization for Amortized Inverse Problems. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 22289–22319. PMLR, July 2023.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*, February 2018.

Molinaro, R., Yang, Y., Engquist, B., and Mishra, S. Neural Inverse Operators for Solving PDE Inverse Problems. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 25105–25139. PMLR, July 2023.

Murray, C. Strengthening airline operations and consumer protections, February 2023.

Najari, N., Berlemont, S., Lefebvre, G., Duffner, S., and Garcia, C. Robust Variational Autoencoders and Normalizing Flows for Unsupervised Network Anomaly Detection. In Barolli, L., Hussain, F., and Enokido, T. (eds.), *Advanced Information Networking and Applications*, pp. 281–292, Cham, 2022. Springer International Publishing. ISBN 978-3-030-99587-4. doi: 10.1007/978-3-030-99587-4_24.

Onken, D., Fung, S. W., Li, X., and Ruthotto, L. OT-Flow: Fast and Accurate Continuous Normalizing Flows via Optimal Transport. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9223–9232, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i10.17113.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):57:2617–57:2680, January 2021. ISSN 1532-4435.

Pyrgiotis, N., Malone, K. M., and Odoni, A. Modelling delay propagation within an airport network. *Transportation Research Part C: Emerging Technologies*, 27:60–75, February 2013. ISSN 0968-090X. doi: 10.1016/j.trc.2011.05.017.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1530–1538, Lille, France, July 2015. JMLR.org.

Richardson, A. Deepwave. Zenodo, September 2023.

Rose, J. Southwest will pay a $140 million fine for its meltdown during the 2022 holidays. *NPR*, December 2023.

Rudolph, M., Wandt, B., and Rosenhahn, B. Same same but DifferNet: Semi-supervised defect detection with normalizing flows. In *Winter Conference on Applications of Computer Vision (WACV)*, January 2021.

Southwest Airlines. Final Summary and Action Plan, 2023.

Stuart, A. M. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, May 2010. ISSN 1474-0508, 0962-4929. doi: 10.1017/S0962492910000061.

Tabak, E. G. and Vanden-Eijnden, E. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, March 2010. ISSN 1539-6746, 1945-0796.

Verine, A., Negrevergne, B., Chevaleyre, Y., and Rossi, F. On the expressivity of bi-Lipschitz normalizing flows. In *Proceedings of The 14th Asian Conference on Machine Learning*, pp. 1054–1069. PMLR, April 2023.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys*, 53(3):63:1–63:34, June 2020. ISSN 0360-0300. doi: 10.1145/3386252.

Zhang, R., Czado, C., and Sigloch, K. Bayesian Spatial Modelling for High Dimensional Seismic Inverse Problems. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 65(2):187–213, February 2016. ISSN 0035-9254. doi: 10.1111/rssc.12118.

A. Lipschitz constants for conditional normalizing flows

In this section, we provide the Lipschitz constants for various conditional normalizing flow architectures; i.e. L such that $|f(z, c_1) - f(z, c_2)| \leq L \|c_1 - c_2\|$ for all z, c_1, c_2.

Remark 3. The conditional inverse autoregressive flow (IAF; Kingma et al. (2016)) has Lipschitz constant $L \leq \prod_i \prod_t (L_{s_t} + L_{m_t})$, where the outer product is over autoregressive blocks and the inner product is over steps within each autoregressive block. L_{s_t} and L_{s_t} are the Lipschitz constants of the neural networks yielding the m_t and s_t values for each autoregressive step (these can be easily bounded for most neural networks; e.g. by the product of the L_2 matrix norms of the weight matrices; Miyato et al. (2018)).

Remark 4. A neural spline flow (Durkan et al., 2019) has Lipschitz constant $L \leq 2\bar{s}$, where $\bar{s}$ is an upper bound on the slope $s = (y_{k+1} - y)/(x_{k+1} - x_k)$ between adjacent knot points of the spline (this can be constrained by construction by ensuring a minimum spline bin width).

Remark 5. Continuous normalizing flows (Chen et al., 2018) have Lipschitz constant $L \leq e^{L_g \Delta t}$ where Δt is the duration of integration and L_g is the Lipschitz constant of the neural network defining the vector field of the flow.

Remark 6 (from (Verine et al., 2023)). Normalizing flows based on invertible residual networks, such as i-ResNet (Behrmann et al., 2019) and Residual Flow (Chen et al., 2019), have Lipschitz constant $L \leq (1 + L_g)^m$, where m is the number of residual blocks and $L_g < 1$ is the Lipschitz constant of the residual block $g(x)$.

Remark 7 (from (Verine et al., 2023)). Normalizing flows based on Glow (Kingma & Dhariwal, 2018) have Lipschitz constant $L \leq \prod \|W_i\|_2$, where the product is over the weight matrices W_i of the convolution blocks.

B. Proof of Theorem 2

Proof. The W_2 metric is defined as an infemum over couplings γ, so in order to provide an upper bound it suffices to propose a coupling between the nominal and anomaly posteriors, $q_\phi(z, \mathbf{0}_K)$ and $q_\phi(z, c^*)$. Recall that the normalizing flow q_ϕ has base distribution q_0 and flow map f_ϕ, where $f_\phi(z, c)$ is assumed to be L-Lipschitz in the second argument. Consider the joint distribution $\gamma(z_1, z_2)$ defined by $z_0 \sim q_0(z)$, $z_1 = f_\phi(z_0, \mathbf{0}_K)$, and $z_2 = f_\phi(z_0, c^*)$. By construction, the marginals of γ in each argument are $q_\phi(z, \mathbf{0}_K)$ and $q_\phi(z, c^*)$, respectively, and so γ is a valid coupling.

This provides the bound

$$\begin{aligned} W_2(q_\phi(\cdot, \mathbf{0}_K), q_\phi(\cdot, c^*)) &\leq \left[\mathop{\mathbb{E}}_{z_1, z_2 \sim \gamma} ||z_1 - z_2||^2 \right]^{1/2} \\ &\leq \left[L^2 ||c^* - \mathbf{0}_K||^2 \right]^{1/2} \\ &\leq L ||c^*|| \end{aligned}$$

□

C. Details on benchmark problems

This section provides additional details for the three types of inverse problem studied in our paper. All problems are implemented using the Pyro probabilistic programming framework (Bingham et al., 2019).

C.1. Seismic waveform inversion

An illustration of the SWI problem is given in Fig. 8. We implement the SWI problem using the Deepwave library (Richardson, 2023). We use latent parameters $z \in \mathbb{R}^{n_x \times n_y}$ representing the subsurface density profile (with spatial resolution $n_x = 10$ and $n_y = 10$), context $y \in \mathbb{R}^{n_T}$ representing the source signal, and observations $x \in \mathbb{R}^{n_s \times n_r \times n_T}$ representing the signal measured at each receiver, where $n_s = 1, n_r = 9, n_T = 100$ are the number of sources, receivers, and timesteps, respectively. Before solving the elastic wave PDE, the density profile is upsampled to 100×30. The observations are corrupted with additive isotropic Gaussian noise. The parameters of this problem are summarized in Table 4.

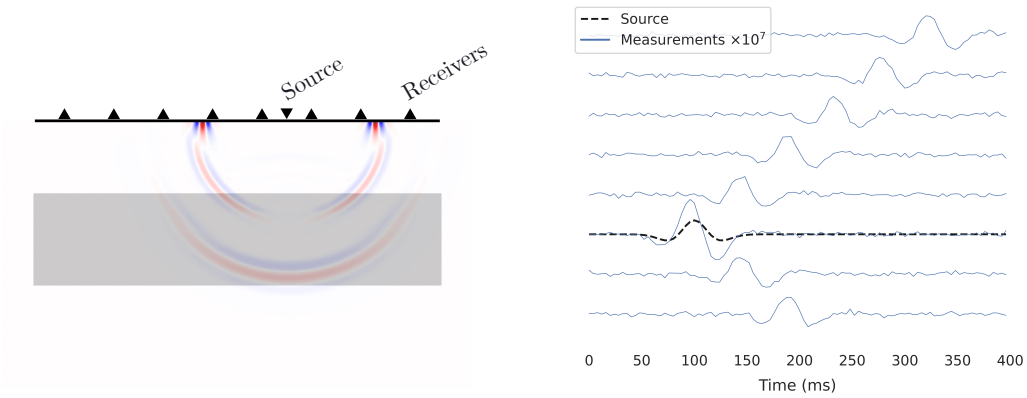

Figure 8. (Left) An illustration of the SWI problem and (right) the receiver measurements (blue) given a source signal (black).

Table 4. Summary of parameters for the SWI problem.

	Dimension
Latent parameters z	
Density profile (10×10)	100
Context y	
—	
Observation x	
Seismic waveform (100 timesteps at 9 receivers)	900

C.2. UAV control

We model the nonlinear attitude dynamics of the UAV as a combination of an unknown linear mapping from the current and desired states to angular rates, then a nonlinear mapping from angular rates to updated UAV orientation. The state $q = [\phi, \theta, \psi]$ includes the roll, pitch, and yaw angles of the UAV, and $\hat{q}$ denotes the commanded orientation. We model the angular rates of the UAV as

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = Aq + K(\hat{q} - q) + d + \eta \tag{8}$$

where A, K, and d are unknown feedforward, feedback, and bias dynamics, and η is Gaussian process noise. The state derivative is related to ω by

$$\frac{d}{dt}q = J^{-1}(q)\omega \tag{9}$$

$$J^{-1}(q) = \begin{bmatrix} 1 & \tan(\theta)\sin(\phi) & \tan(\phi)\cos(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \tag{10}$$

We apply a first-order time discretization to yield the one-step stochastic dynamics

$$q_{t+1} = q_t + \delta_t J^{-1}(q)\left(Aq + K(\hat{q} - q) + d + \eta\right)$$

and observed states are additionally corrupted by Gaussian noise. A summary of the parameters for this problem are given in Table **??**.

An example trajectory, including both nominal and anomalous segments, for the UAV dataset are shown in Fig. 9. In this case, the anomaly is relatively easy to detect; the challenge is understanding how the aircraft's flight dynamics change during the failure so that a recovery controller can be designed to handle this case.

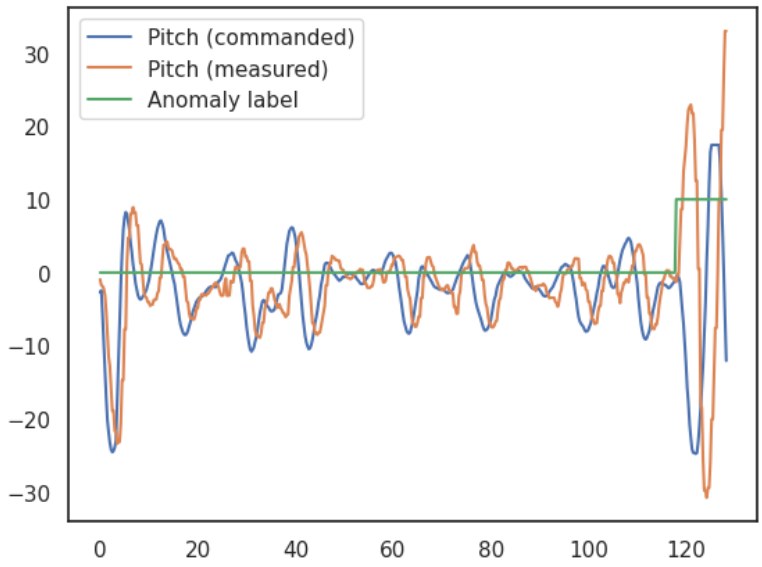

Figure 9. Example trajectory that includes an elevator failure, including both nominal and anomalous segments.

Table 5. Summary of parameters for the UAV problem.

	Dimension
Latent parameters z	
Feedforward matrix A (3×3)	9
Feedback matrix K (3×3)	9
Bias term d	3
Context y	
Current state	3
Desired orientation	3
Observation x	
Next state	3

C.3. Air traffic network

The input to our air traffic model is a list of scheduled flights, each specifying an origin and destination airport and a scheduled departure and arrival time. The latent state z includes the mean travel time between each origin/destination pair, the mean service time at each airport (which affects both arriving and departing aircraft and models taxi, deicing, and ATC delays), the mean turnaround time at each airport (the minimum time that must elapse before an arriving aircraft may depart), the baseline cancellation rate at each airport, and the initial number of aircraft at each airport. A summary of these parameters are given in Table 7. So that the benchmarks in Section 4 can be run in a reasonable time, we restrict the ATC problem used for benchmarking to the four busiest airports and do not model cancellations, but we use the ten busiest airports and do include cancellations in our case study in Section 5.

The model steps through the scheduled flights in 15 minute increments. In each increment, it checks for the flights that are scheduled to depart from each airport. Each of these flights receives a certain probability of cancellation given by

$$P(\text{canceled}) = 1 - (1 - p_c)\sigma\left(10\frac{\text{\# available aircraft}}{\text{\# departing flights in this block}}\right) \tag{11}$$

where p_c is the baseline cancellation rate for the origin airport and σ is the sigmoid function, so the probability of cancellation is p_c when there are more available aircraft than scheduled departures and approaches 1 as the number of available aircraft decreases. Cancellations are sampled from a relaxed Bernoulli distribution with this cancellation probability and a straight-through gradient estimator. If a flight is canceled, it is marked as such and the observation for that flight will just be

`canceled` and will not include actual departure and arrival times. If the flight is not canceled, then it is moved to the runway queue if there are enough aircraft available; otherwise, it is delayed until the next time block.

Both departing and arriving flights are served using a single M/M/1 queue for each airport, with service times drawn from an exponential distribution with the mean specified according to each airport's mean service time. Once airborne, departing flights are assigned a random flight time from a Gaussian with mean given by the mean travel time for each route and fixed variance. Once this travel time has elapsed, they enter the runway queue at the destination airport. Once an aircraft has landed, it does not become available to serve new flights until the minimum turnaround time has elapsed (which is sampled from a Gaussian with mean given by the mean turnaround time for each airport). Observations for non-canceled flights include the simulated arrival and departure times, plus some fixed-variance Gaussian noise.

Table 6. International Air Transport Association (IATA) codes and full names of the ten busiest airports in the Southwest network.

DEN	Denver International Airport
DAL	Dallas Love Field Airport
MDW	Chicago Midway International Airport
PHX	Phoenix Sky Harbor International Airport
HOU	William P. Hobby Airport
LAS	McCarran International Airport
MCO	Orlando International Airport
BNA	Nashville International Airport
BWI	Baltimore/Washington International Thurgood Marshall Airport
OAK	Oakland International Airport

Table 7. Summary of parameters for the ATC problem. n_{airport} indicates the number of airports in the model. n_{flights} indicates the total number of scheduled flights. † indicates parameters that are only included in the case study.

	Dimension	Top-4 (Section 4)	Top-10 (Section 5)
Latent parameters z			
Logarithm of turnaround time at each airport (mean minimum delay between arrival and departure)	n_{airport}	4	10
Logarithm of service time at each airport (mean delay between pushback and takeoff)	n_{airport}	4	10
Logarithm of mean travel times between each airport	n_{airport^2}	16	100
Logarithm of initial aircraft reserves at each airport	$n_{\text{airport}}{}^\dagger$	–	10
Logarithm of baseline cancellation probability at each airport	$n_{\text{airport}}{}^\dagger$	–	10
Context y			
Scheduled arrival time of each flight	n_{flights}	44–102	405–497
Scheduled departure time of each flight	n_{flights}	44–102	405–497
Observation x			
Actual arrival time of each flight	n_{flights}	44–102	405–497
Actual departure time of each flight	n_{flights}	44–102	405–497
Whether each flight was cancelled	n_{flights}	44–102	405–497

C.4. Toy 2D problem

The data for the 2D toy problem is generated by uniformly sampling nominal data:

$$\begin{aligned}\theta &\sim \mathcal{U}(0, \pi)\\ x &\sim \mathcal{N}(\cos\theta - 0.5, 0.1)\\ y &\sim \mathcal{N}(\sin\theta - 0.25, 0.1)\end{aligned}$$

and anomaly data

$$\theta \sim \mathcal{U}(\pi, 2\pi)$$
$$x \sim \mathcal{N}(\cos\theta + 0.5, 0.1)$$
$$y \sim \mathcal{N}(\sin\theta + 0.75, 0.1)$$

Since this problem is meant as an easy-to-visualize test for whether a method can learn a posterior distribution with a complex shape, we set $[x, y]$ as the latent parameters and assume they are observed directly (with the addition of Gaussian noise), rather than treating θ as the latent parameter (which would lead to a very easy-to-fit posterior).

D. Implementation details

We implement CALNF using neural spline flows (NSF) as the underlying normalizing flow (Durkan et al., 2019). We note that CALNF is agnostic to the underlying flow architecture; we also tried using masked autoregressive flows (Huang et al., 2018), which trained faster but had slightly worse performance, and continuous normalizing flows (Chen et al., 2018), which trained much more slowly.

We implement β-NSF using neural spline flows with a KL regularization penalty between the learned anomaly and nominal posteriors. We implement an RNODE-derived method that includes only the W_2 regularization term, not the Froebenius norm regularization term (which is used only to speed training and inference, not to regularize the learned posterior; Finlay et al., 2020).

We extend our method to anomaly detection by defining a score function as the ELBO of a given observation, approximated using 10 samples from the learned posterior.

All methods were implemented in Pytorch using the Zuko library for normalizing flows (Zuk, 2024). The neural spline flows used 3 stacked transforms, and all flows used two hidden layers of 64 units each with ReLU activation (except for the continuous flows on the 2D problem, which use two hidden layers of 128 units each). All flows were trained using the Adam optimizer with the learning rate 10^{-3} (except on the UAV problem, which used a learning rate of 10^{-2}) and gradient clipping. CALNF used $K = 5$ on all problems. All methods were trained on a single NVIDIA GeForce RTX 2080 Ti GPU, with 200, 500, 1000, and 300 epochs for the 2D, SWI, UAV, and ATC problems, respectively.

Code examples, including scripts for reproducing the results in Tables 1 and 2 and notebooks containing our data analysis for Section 5, are included in the attached supplementary material.

E. Sensitivity analysis

Fig. 10 shows the sensitivity of our method to different values of K on the SWI benchmark. We find that there is a slight trend towards better performance as K increases, and that including the calibration step (rather than using a fixed c^*) improves performance at all levels of K.

F. Additional anomaly detection metrics

Table 8 includes additional metrics on the downstream use of our method for anomaly detection. Precision and recall are reported for the threshold that optimizes the difference between the true positive and false positive rates.

G. Additional results on Southwest Airlines case study

A timeline of the 2022 Southwest Airlines scheduling crisis is shown in Fig. 11.

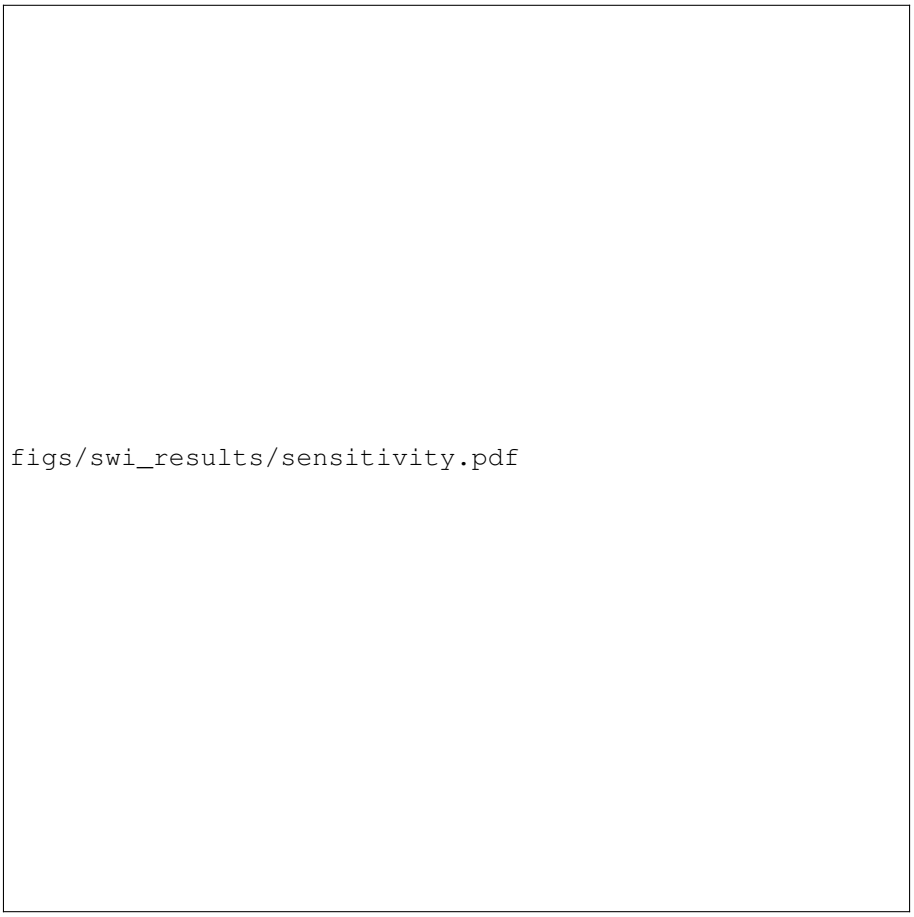

Figure 10. The ELBO on a held-out test set for the anomaly posterior learned using CALNF on the SWI example using a varying number of subsamples.

Table 8. Additional anomaly detection metrics.

		AUROC	AUCPR	PRECISION	RECALL
2D					
	NF-AD	0.999 ± 0.001	1.00 ± 0.001	0.989 ± 0.008	0.984 ± 0.010
	NF-AD$_{\text{KL}}$	0.999 ± 0.001	1.00 ± 0.001	0.989 ± 0.008	0.984 ± 0.010
	NF-AD$_{W_2}$	0.995 ± 0.002	1.00 ± 0.002	0.988 ± 0.003	0.961 ± 0.009
	CALNF	0.996 ± 0.005	1.00 ± 0.004	0.984 ± 0.010	0.974 ± 0.020
SWI					
	NF-AD	0.740 ± 0.030	0.768 ± 0.032	0.736 ± 0.050	0.578 ± 0.024
	NF-AD$_{\text{KL}}$	0.740 ± 0.030	0.768 ± 0.032	0.736 ± 0.050	0.578 ± 0.024
	NF-AD$_{W_2}$	0.647 ± 0.033	0.616 ± 0.032	0.602 ± 0.025	0.668 ± 0.045
	CALNF	0.787 ± 0.022	0.815 ± 0.028	0.790 ± 0.028	0.599 ± 0.070
UAV					
	NF-AD	0.600 ± 0.157	0.492 ± 0.207	0.426 ± 0.493	0.213 ± 0.344
	NF-AD$_{\text{KL}}$	0.707 ± 0.153	0.620 ± 0.203	0.633 ± 0.424	0.438 ± 0.316
	NF-AD$_{W_2}$	0.543 ± 0.077	0.406 ± 0.080	0.463 ± 0.538	0.088 ± 0.167
	CALNF	0.703 ± 0.165	0.621 ± 0.219	0.623 ± 0.435	0.449 ± 0.321

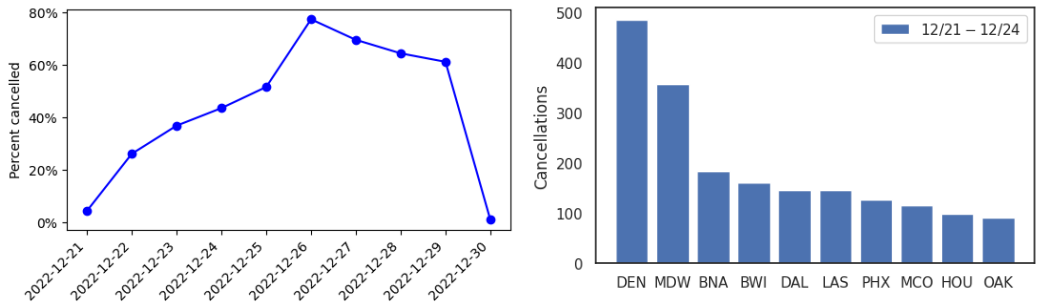

Figure 11. (Left) Timeline of cancellations during the 2022 Southwest Airlines scheduling crisis. (Right) Cancellations at the 10 busiest airports during the first four days of the disruption.