

Loss of Control Detection for Commercial Transports Using Conditional Variational Autoencoders

Extended Abstract

Newton H. Campbell Jr.*

NASA Goddard Space Flight Center, Greenbelt, MD, 20771

Jared Grauer[†] and Irene Gregory[‡]

NASA Langley Research Center, Hampton, Virginia, 23681

This work describes a detector for the loss of control of a commercial transport in flight. The detector has a belief state defined by the latent variable stochastic modeling of a conditional variational autoencoder (CVAE) constructed with bidirectional recurrent layers. In 2000, the Boeing Company and the NASA Langley Research Center jointly developed a quantitative set of metrics for defining loss-of-control (LOC) for a commercial transport. We use the thresholds for these quantitative metrics to define a condition vector for training the CVAE. We demonstrate through experimentation that reconstruction probability is an accurate indicator that the vehicle has shifted to an LOC state. Second, we introduce a technique for inferring that the vehicle is experiencing a flight state change is approaching by measuring a shift in the sampling Gaussian distributions of the latent space. We provide an analysis of its applicability to flight data from a NASA generic commercial transport-type aircraft.

I. Nomenclature

AUROC	=	Area Under the Receiver Operating Characteristics
C2	=	Command-and-Control
CVAE	=	Conditional Variational Autoencoder
GRU	=	Gated Recurrent Unit
LOC	=	Loss of Control
LSTM	=	Long-Short Term Memory
RNN	=	Recurrent Neural Network
ROC	=	Receiver Operating Characteristic
RPH	=	Roll, Pitch, Heading
RPY	=	Roll, Pitch, Yaw
VAE	=	Variational Autoencoder
UAM	=	Urban Air Mobility
.	=	time derivative

*Senior Principal Solutions Architect, Science Applications International Corporation, NASA Enterprise IT

[†]Research Engineer, Dynamic Systems and Control Branch, MS 308, Associate Fellow AIAA.

[‡]NASA Senior Technologist for Advanced Control Theory and Application, Associate Fellow AIAA.

II. Introduction

A crucial component of the autonomous urban air mobility (UAM) mission [1] is the ability to quickly and accurately approximate the belief state of a vehicle. For contingency planning, an approximation of reachable vehicle states must be quickly computed during flight, in real time. The vehicle should be able to quantify the likelihood that it is in a particular state and track the likelihood that an action (or set of actions) will result in a new state. Furthermore, an intelligent artificial agent must be able to evaluate many branching conditions using a succinct belief state definition during and after reachability analysis. The use of an intelligent agent for contingency planning necessitates the adoption of the concept of mission-command, allowing the system to draw from long-vetted concepts from command-and-control (C2)[2]. It will be up to this C2 agent to understand if and why it is in a belief state where the branching condition's values are unknown.

In this work, we describe and experimentally vet a viable belief state framework for such a C2 agent. All constructs of the agent's architecture are guided by a common description of belief states. To do so, the C2 must be able to identify the point at which the aircraft changes qualitative states through measurement. During a flight mission, the CA takes in physical information (health, sensor data, kinematic measurements), termed *percepts*, about the aircraft, finds the lower dimensional embeddings of this data (with probability $P(x)$, for percepts x), and interprets dramatic shifts in these embeddings as **anomalies**. Anomalies are defined as observations that deviate so much from the other observations as to arise suspicions that they were generated by different mechanisms. In the case of control systems, this is the equivalent of a disturbance signal affecting the system's output, and result in an increase of the system tracking error. Our hypothesis is that the most significant of these anomalies indicate a loss-of-control disturbance signal.

We examine the envelope conditions defined in a previous joint study by Boeing and NASA [3]. We use these envelope conditions to describe a set of qualitative vehicle states and investigate the mapping of these states to a latent variable model. This model is defined by the latent space of a conditional variational autoencoder (CVAE)[4, 5], trained on the percepts of a commercial transport, with a conditioning vector defined by the vehicle's envelope status. The goal of this paper is to explore the utility of the CVAE in describing when the vehicle is in a loss-of-control state and when it is shifting in or out of specific envelopes. Our main contributions are as follows.

- We provide a forensic analysis on loss-of-control for a series of T-2 aircraft flights by training and measuring the reconstruction probability of a CVAE, confirming the loss-of-control definition of the original study using the anomaly detection capabilities of the CVAE.
- We design a new anomaly metric for the CVAE architecture, called Gaussian Shift, that directly measures the gradient of "surprise" in the latent space over time. We provide an analysis of its applicability to interpret anomalous behavior whether inside or outside of an envelope.

The remainder of this paper is organized as follows. Section III provides the previous loss-of-control study by NASA and Boeing, as well as the theoretical background for anomaly detection for the CVAE model. In Section IV, we describe the theoretical basis of our anomaly detection techniques. In Section V, we describe the results of our forensic analysis of the loss-of-control study, using T-2 flight data, and apply our proposed anomaly detection technique to monitoring anomalous shifts with respect to a given flight envelope status. And finally, conclusions are given in Section VI.

III. Related Work

A. Loss-of-Control Research

The NASA Aviation Safety Program [6] was established in response to a national goal to reduce the fatal accident rate for aviation by 80% within 10 years, as recommended by the Presidential Commission on Aviation Safety and Security. The program involved all four NASA aeronautics centers and coordinated with the Federal Aviation Administration, other government agencies, industry, and academia.

One large element of the program entailed research into "loss of control" (LOC) accidents. In Ref. [7], data from 126 LOC-related accidents between 1979 and 2009 were analyzed to identify worst-case combinations of attributes and sequencing of precursors to identify intervention strategies. Much of the work in the Aviation Safety Program was published in a special issue of the Journal of Guidance, Control, and Dynamics [8].

One other research avenue into LOC was a joint work by Boeing and NASA [3] in which 24 flights, classified by the Commercial Aviation Safety Team (CAST) as LOC containing events, were analyzed. Five envelopes, collectively known as the Quantitative Loss-of-Control Criteria (QLC), were compiled that are defined by different variables typically used in airplane flight dynamics, aerodynamics, structural integrity, and flight control work. When applied to

the accident data, violation of three or more of these envelopes was able to predict onset of LOC in 95% of the cases studied. In Ref. [9], these LOC envelopes formed the basis for a flight envelope protection system. More details about these LOC envelopes, particularly in relation to their use in the current work, is discussed later in Section V.

B. Anomaly Detection using Deep Variational Autoencoders

Outside of deep learning, a myriad of other machine learning structures have been proposed for detecting anomalies during flight, including decision trees [10, 11], support-vector machines [12–14], and Naive Bayesian models [15]. These are supervised learning methods, many of which are limited to poor sensitivity, a need for dimensionality reduction, and an inability to interpret anomalies in the latent feature space.

An *autoencoder* is a neural network that learns and captures the latent features of its input data into a single hidden layer of the network [16]. The *variational autoencoder* (VAE) is an expressive model that explicitly captures and encodes these latent features as joint probability distributions [17, 18]. VAEs are an unsupervised learning method that do not fall prey to the limitations of the aforementioned anomaly detection methods. The straightforward approach to anomaly detection using the VAE considers a Monte-Carlo estimate of the reconstruction error, known as the reconstruction probability, to identify uncharacteristic changes in groups of features [19]. Recent studies that have built on this approach argue that implementations run the risk of not being applicably discriminative, as the model is expected to only be trained on inliers [20, 21]. Other recent work points out that the model cannot sufficiently learn complex multimodal data [22].

The conditional variational autoencoder is a semi-supervised learning method that reconstructs input data, x , for a particular label c [5]. The model has been demonstrated to naturally estimates how anomalous the input data x is in the context of c based on reconstruction error [23]. Recent experimentation has shown the utility of using the Kullback-Leibler divergence, denoted D_{KL} , of the CVAE latent space from the standard Gaussian distribution in anomaly detection. It has been shown to yield less false positives than the VAE when a proper x is provided for a rare c [24, 25]. D_{KL} measures the amount of information loss when the prior over the latent variable is used to approximate the posterior distribution. Therefore, a low value for $|D_{KL}|$ indicates that x is an inlier. The thresholds for anomalies in this study is identified through empirical testing.

IV. Methodology

In this study, we predict loss-of-control of a commercial transport by assessing two measurements of a trained conditional variational autoencoder: **Reconstruction Probability** and **Gaussian Shift**.

A. Preliminaries

Let z be the set of latent variables for a given input percept x . As we can only observe x , a VAE uses Bayes' theorem to sample z by leveraging the prior distribution $P(z)$, the likelihood $P(x|z)$, and the posterior $P(z|x) = \frac{P(x|z)P(z)}{P(x)}$. Assume that $P(z)$ is a standard Gaussian distribution and that $P(x|z)$ is a Gaussian distribution whose mean and covariance matrix are defined by deterministic functions of z . A deep neural network takes in a set of percepts x as input, and passes it through layers to a latent space. The latent space consists of a set of two vectors of nodes: one representing the sampling means of the probability distribution of each latent variable, μ_z , and the other vector representing the log of their variance ($\log \sigma_z^2$). The segment of the network that takes in percepts and approximates z from the marginal likelihood $Q_\varphi(z|x)$, for variational parameters φ , is known as the *encoder*. The *decoder* segment of the network takes in samples these latent feature distributions and approximates x from the marginal likelihood $P_\theta(x|z)$, for generative model parameters θ .

The conditional variational autoencoder (CVAE) extends this concept by introducing a conditioning vector c , typically an encoded labeling for x , c is used as the input to the encoder alongside x , as well as to the decoder alongside a sampling of z . This means that the encoder and decoder now use the marginal likelihoods, respectively, by maximizing the marginal likelihoods $Q_\varphi(z|x, c)$ and $P_\theta(x|z, c)$, respectively. Like the VAE, the CVAE learns these parameters by minimizing the Kullback-Leibler (KL) divergence between $Q_\theta(z)$ and the standard Gaussian. The model parameters are trained to optimize the following lower bound based on the ELBO method:

$$L(\theta, \varphi; x) = -D_{KL}(Q_\varphi(z|x, c) || P_\theta(x|z, c)) + E_{Q_\varphi(z|x, c)} [\log P_\theta(x|z, c)] \quad (1)$$

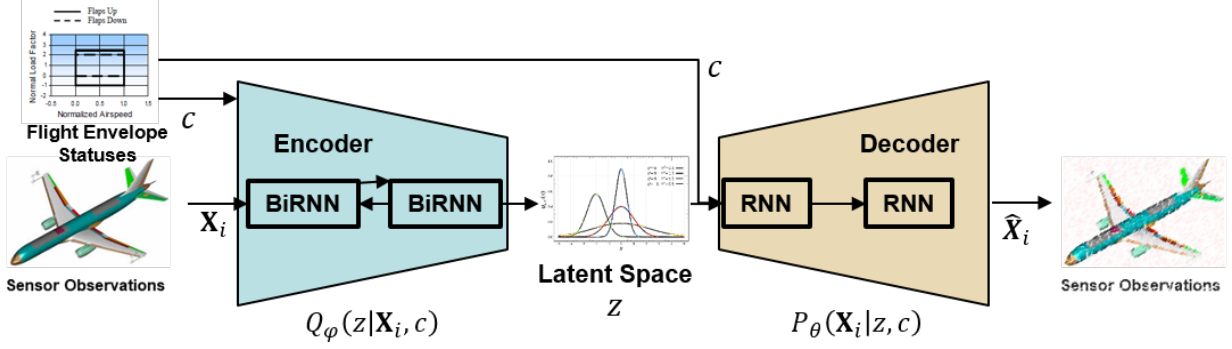


Fig. 1 Overview of the CVAE model applied to T-2 Aircraft flight data. In-flight sensor data are used for input, along with a binary-encoded conditioning vector representing the flight envelope status for that input, to an encoder comprised of bidirectional RNN layers. The decoder is composed of unidirectional RNN layers.

B. Conditional Variational Autoencoder Architecture

Fig. 1 illustrates the architecture of our CVAE. We construct the encoder portion of the CVAE using variants of bidirectional recurrent neural network (RNN) layers and use unidirectional layers to construct the decoder portion. Intuitively, only the encoder side of the CVAE needs to be bidirectional. The latent space distributions that are encoded by the encoder will capture information about future timesteps, as well as past timesteps. The data are encoded in such a way that only the sequence of the distributions matter.

We specify the input to our configuration in a way similar to that of work on VAE-based anomaly detection using Gaussian Mixture principles [22]. We consider a flight vehicle with D sensors. We denote the physical state of the system at time t by $\mathbf{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \dots, x_s^{(t)}, \dots, x_D^{(t)}]$, where $x_s^{(t)}$ represents the value of sensor s ($s = 1, 2, \dots, D$) at time t . Each sensor is read as a scalar or a vector, and then $\mathbf{x}^{(t)}$ is flattened as a single vector. An arbitrary time series of T measurements from all sensors forms a $D \times T$ matrix denoted as

$$\mathbf{X}_i = [\mathbf{x}^{(i,1)}; \mathbf{x}^{(i,2)}; \dots; \mathbf{x}^{(i,T)}]_{D \times T} \quad (2)$$

where $\mathbf{x}^{(i,t)} = \mathbf{x}^{((i-1) \times T + t)}$.

In this study, we develop and train two architectures based on two popularly-used forms of RNN: Gated Recurrent Unit (GRU) [26] and Long-Short Term Memory (LSTM) [27]. $\mathbf{X}_i$ is fed into a layered Bidirectional RNN encoder one timestep at a time. The internal states of the unidirectional RNNs are governed by the following equations:

Unidirectional GRU

$$\begin{aligned} z_t &= \sigma(W^{(z)}\mathbf{X}_t + U^z h_{t-1} + b^{(z)}) \text{ (Update Gate)} \\ r_t &= \sigma(W^{(r)}\mathbf{X}_t + U^r h_{t-1} + b^{(r)}) \text{ (Reset Gate)} \\ h'_t &= \tanh(W^{(xh)}\mathbf{X}_t + r_t \odot U^{(rh)} h_{t-1} + b^{(h')}) \text{ (Current Memory)} \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \text{ (Final Memory)} \\ y_t &= F(W^{(y)} h_t + b^{(y)}) \text{ (Output Activation)} \end{aligned}$$

Unidirectional LSTM

$$\begin{aligned} i_t &= \sigma(W^{(i)}\mathbf{X}_t + U^i h_{t-1} + b^{(i)}) \text{ (Input Gate)} \\ f_t &= \sigma(W^{(f)}\mathbf{X}_t + U^f h_{t-1} + b^{(f)}) \text{ (Forget Gate)} \\ o_t &= \sigma(W^{(o)}\mathbf{X}_t + U^o h_{t-1} + b^{(o)}) \text{ (Output/Exposure Gate)} \\ \tilde{c}_t &= \tanh(W^c \mathbf{X}_t + U^c h_{t-1}) \text{ (Current Memory Cell)} \\ c_t &= f_t \odot \tilde{c}_{t-1} + i_t \odot \tilde{c}_t \text{ (Final Memory Cell)} \\ h_t &= o_t \odot \tanh c_t \text{ (Hidden State)} \\ y_t &= F(W^{(y)} h_t + b^{(y)}) \text{ (Output Activation)} \end{aligned}$$

where σ is the sigmoid function, $\odot$ is an element-wise multiplication operator, $W^{(z)}$, $W^{(r)}$, $W^{(xh)}$, $U^{(z)}$, $U^{(r)}$, and $U^{(rh)}$ are learned weight matrices, and $b^{(z)}$, $b^{(r)}$, $b^{(h')}$, and $b^{(y)}$ are biases. The output activation function F can be one of any of the common activation functions for neural networks. In this study, we use the SELU (scaled exponential linear unit) activation function [28] for F , due to its self-normalizing properties.

In the bidirectional case, the RNNs apply temporal directionality to the layer's gates, weights, and biases. In the GRU case, $\vec{h}_t$ and $\overleftarrow{h}_t$ are computed as:

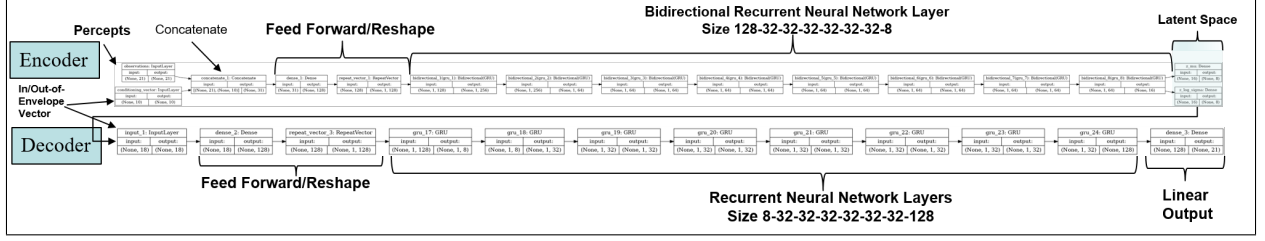


Fig. 2 Detailed description of the CVAE model applied to Flights of the T-2 Aircraft.

$$\begin{aligned}
 \vec{z}_t &= \sigma(\overrightarrow{W^{(z)}} \mathbf{X}_t + \overrightarrow{U^{(z)}} \overrightarrow{h_{t-1}} + \overrightarrow{b^{(z)}}) \\
 \vec{r}_t &= \sigma(\overrightarrow{W^{(r)}} \mathbf{X}_t + \overrightarrow{U^{(r)}} \overrightarrow{h_{t-1}} + \overrightarrow{b^{(r)}}) \\
 \vec{h}'_t &= \tanh(\overrightarrow{W^{xh}} \mathbf{X}_t + \vec{r}_t \odot \overrightarrow{U^{rh}} \overrightarrow{h_{t-1}} + \overrightarrow{b^{(h')}}) \\
 \vec{h}_t &= \vec{z}_t \odot \overrightarrow{h_{t-1}} + (1 - \vec{z}_t) \odot \vec{h}'_t
 \end{aligned}
 \quad \Bigg| \quad
 \begin{aligned}
 \overleftarrow{z}_t &= \sigma(\overleftarrow{W^{(z)}} \mathbf{X}_t + \overleftarrow{U^{(z)}} \overleftarrow{h_{t-1}} + \overleftarrow{b^{(z)}}) \\
 \overleftarrow{r}_t &= \sigma(\overleftarrow{W^{(r)}} \mathbf{X}_t + \overleftarrow{U^{(r)}} \overleftarrow{h_{t-1}} + \overleftarrow{b^{(r)}}) \\
 \overleftarrow{h}'_t &= \tanh(\overleftarrow{W^{xh}} \mathbf{X}_t + \overleftarrow{r}_t \odot \overleftarrow{U^{rh}} \overleftarrow{h_{t-1}} + \overleftarrow{b^{(h')}}) \\
 \overleftarrow{h}_t &= \overleftarrow{z}_t \odot \overleftarrow{h_{t-1}} + (1 - \overleftarrow{z}_t) \odot \overleftarrow{h}'_t
 \end{aligned}$$

The output of the bidirectional GRU is then computed as:

$$\hat{y}'_t = f(W_{\hat{y}}[\vec{h}_t \overleftarrow{h}_t] + b_{\hat{y}}) \quad (3)$$

The same concept of applying temporal directionality applies to the bidirectional LSTM. After some trial and error, we decided to comprise the encoder of a single feed-forward layer with tanh activation, followed by 8 bidirectional RNN layers of decreasing size (from input to latent space). We construct the decoder using 8 (unidirectional) RNN layers (of increasing size) followed by a single feed-forward layer with tanh activation. The latent space is comprised of a set of feed-forward nodes z with linear activations. A final detailed diagram of the GRU case is shown in Fig. 2.

C. Anomaly Detection using Reconstruction Probability

We propose that a drop in the reconstruction of commercial transport flight data from the low-dimensional representation of the CVAE represents a loss-of-control event. We use a variant of the Monte Carlo approach to computing anomalies using the reconstruction probability [22]. At time t , the reconstruction probability $R(\mathbf{X}|\hat{\mathbf{X}})_{[t]}$ is computed once we have estimated the reconstruction $\hat{\mathbf{X}}_t$ of input $\mathbf{X}_t$. The likelihood of observation X , i.e., $P_{\theta}(X)$ in CVAE, is an important output, since we want to see how well we can reproduce an observation in the context of its envelope status given X follows the normal patterns. The Monte Carlo approach computes the probability density of X , by $P_{\theta}(X) = E_{P_{\theta}(Z)}[P_{\theta}(X|Z)]$. While sampling on this prior is not feasible in the real world, we can directly compute the reconstruction probability $E_{Q_{\phi}(Z|X)}[\log P_{\theta}(X|Z)]$. To get the log-likelihood of our reconstruction by observing the we exponentiate the mean square error. First, we encode the test point using the CVAE encoder and then sample a number of times. Therefore, using the CVAE decoder, we can then reconstruct the test point and compute the probability that the test point was generated by the decoded distribution as:

$$\begin{aligned}
 R(\mathbf{X}|\hat{\mathbf{X}})_{[t]} &= \frac{1}{L} \sum_{l=1}^L P_{\theta}(\mathbf{X}_t | \mu_{\hat{\mathbf{x}}|[t,l]}, \sigma_{\hat{\mathbf{x}}|[t,l]}) \\
 &\approx E_{Q_{\phi}(Z|X)}[\log P_{\theta}(X|Z)] = e^{(\hat{\mathbf{X}}_t - \mathbf{X}_t)^2}
 \end{aligned} \quad (4)$$

The lower the number, the more anomalous the test point. Across all observations $(\mathbf{X}, \hat{\mathbf{X}})$, regardless of conditioning vector c , we establish a threshold α_r such that if $R(\mathbf{X}|\hat{\mathbf{X}})_{[t]} < \alpha_r$, we declare that the observation at that timestep is anomalous. This indicates that loss-of-control is happening at that specific timestep.

We also observe the reconstruction probability of observations in the context of conditioning vector c . For each c , we use min-max normalization to identify the reconstruction probability in the context of a particular data label as:

$$R(\mathbf{X}|\hat{\mathbf{X}}, c)_{[t]} = \frac{1}{L} \sum_{l=1}^L P_{\theta}(\mathbf{X}_t | \mu_{\hat{\mathbf{x}}|[t,l]}, \sigma_{\hat{\mathbf{x}}|[t,l]}, c) \quad (5)$$

To use $R(\mathbf{X}|\hat{\mathbf{X}}, c)$ in anomaly detection, we compute $R(\mathbf{X}|\hat{\mathbf{X}})$, once using our Monte Carlo method, and label each observation with its corresponding vector c . For each labeling c , we then perform a min-max normalization to obtain the corresponding vector:

$$R(\mathbf{X}|\hat{\mathbf{X}}, c)' = \frac{R(\mathbf{X}|\hat{\mathbf{X}}, c) - R(\mathbf{X}|\hat{\mathbf{X}}, c)_{\min}}{R(\mathbf{X}|\hat{\mathbf{X}}, c)_{\max} - R(\mathbf{X}|\hat{\mathbf{X}}, c)_{\min}} \quad (6)$$

Again, we set a threshold α_c such that if $R(\mathbf{X}|\hat{\mathbf{X}}, c)'_{[t]} < \alpha_c$, the observation is anomalous in the context of the label (in our case, the envelope status).

D. Anomaly Detection using Gaussian Shift

Fundamentally, the Gaussian distribution predicted at each timestep t represents an approximation of the stochastic process that causes our data to be observed. A drastic shift in the shape (variance) or position (mean) of this distribution indicates that a sufficiently different stochastic process is causing our observations. We identify these sufficient differences as follows. At each timestep t and subsequent timestep $t + 1$, we analyze a shift in the latent Gaussian distribution $P(z_t)$ required to reconstruct an input $\mathbf{X}_t$ to the Gaussian distribution $P(z_{t+1})$ required to reconstruct an input $\mathbf{X}_{t+1}$.

The shift between subsequent observations can be measured in terms of the Kullback-Leibler (KL) divergence. The KL-divergence between an arbitrary multivariate Gaussian distribution p with mean μ_1 and variance σ_1^2 and an arbitrary multivariate Gaussian distribution q with mean μ_2 and variance σ_2^2 is given by:

$$\begin{aligned} D_{\text{KL}}(p||q) &= \int \left[\frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) + \frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right] \times p(x) dx \\ &= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2} \text{tr} \{ E[(x - \mu_1)(x - \mu_1)^T] \Sigma_1^{-1} \} + \frac{1}{2} E[(x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2)] \\ &= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2} \text{tr} \{ I_d \} + \frac{1}{2} (\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \text{tr} \{ \Sigma_2^{-1} \Sigma_1 \} \\ &= \frac{1}{2} \left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - d + \text{tr} \{ \Sigma_2^{-1} \Sigma_1 \} + (\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) \right]. \end{aligned} \quad (7)$$

The KL-Divergence is an information-theoretic measurement of relative surprise. For two arbitrary probability distributions, p and q , how much information is lost if q is an approximation of p ? This is not a symmetric measurement. So this applies to the latent space of our architecture in two ways:

- Let $p = P(z_t)$ and $q = P(z_{t+1})$. Then the KL-Divergence tells us how much information is lost if the processes that spawn the next observation is used to approximate the processes that spawned the current distribution. In other words, how surprising is it that the aircraft experienced the dynamics that resulted in $\mathbf{X}_{t+1}$ after experiencing the dynamics that resulted in $\mathbf{X}_t$?
- Let $p = P(z_{t+1})$ and $q = P(z_t)$. Then the KL-Divergence tells us how much information is lost when the processes that spawn the current observation are used to approximate the processes that spawned the next observation. In other words, how surprising is it that the aircraft experienced the dynamics that resulted in $\mathbf{X}_t$ prior to experiencing the dynamics that resulted in $\mathbf{X}_{t+1}$?

Two answer both of these questions in one measurement, we use the Jensen-Shannon (JS) divergence, a symmetric version of the KL Divergence, defined as follows:

$$D_{\text{JS}}(p || q) = \frac{D_{\text{KL}}(p || m) + D_{\text{KL}}(q || m)}{2} \quad (8)$$

where $m = \frac{1}{2}(p + q)$. Given two multivariate Gaussians p and q , $m \sim \mathcal{N}(\mu_m, \Sigma_m)$, where $\mu_m = \frac{1}{2}(\mu_p + \mu_q)$ and $\Sigma_m = \frac{1}{2}(\Sigma_p + \Sigma_q)$. The JS-Distance is the metric $\hat{D}_{\text{JS}}(p, q) = \sqrt{D_{\text{JS}}(p || q)}$. We hypothesize that by monitoring $\hat{D}_{\text{JS}}$ over subsequent timesteps, we will be able to detect qualitative changes in the state of the aircraft. Identifying a threshold for $\hat{D}_{\text{JS}}$ for which to raise a detection alert will require some empirical testing for an application. Empirical testing can result in an arbitrary metric that does not hold for many cases. However, the latent space representation of an ELU-based VAE decoder produces a smooth manifold[29, 30]. If properly trained on an ELU-based network, the gradient of the change in the latent space representations over time should be smooth, until a significant change has occurred. Therefore,

we monitor the gradient and raise a detection alert when it is above an arbitrarily small threshold τ . Formally, an alert signifying a future notable state change is raised if

$$|\nabla \hat{D}_{JS}(Q_{\varphi}(z_t|x_t, c_t), Q_{\varphi}(z_{t+1}|x_{t+1}, c_{t+1}))| > \tau \quad (9)$$

V. Experiments

A. Test Aircraft and Flight Data

The test aircraft used is the NASA T-2, which is shown in Fig. 3. The T-2 is a 5.5% dynamically-scaled version of a generic, transport-type aircraft, with twin jet engines mounted under the wings and retractable tricycle landing gear. It has a 6.8 ft wingspan and weighs about 50 lbf.

The T-2 was used in the NASA Aviation Safety Program flight test capability called Airborne Subscale Transport Aircraft Research (AirSTAR). The aircraft is piloted after takeoff by a research pilot in a mobile control room using synthetic vision displays drawn from telemetry data and a database of the local terrain. Control deflection commands are generated from the research pilot and the ground-based flight control system.

The aircraft is instrumented with research-quality hardware. Two inertial measurement units are housed near the aircraft nominal center of mass that provide rate gyroscope and accelerometer measurements, as well as attitude solutions. Each wingtip has a boom extending to an air data probe. Each of the 16 independently-actuated control surfaces are instrumented with potentiometers on the rotation axes for measuring deflections. Flight data were measured at 200 Hz, downsampled to 50 Hz, and digitally low-pass filtered to 16 Hz for analysis.

A total of 58 flights, between 2010 and 2011, were conducted at Fort Pickett, near Blackstone, Virginia, using the T-2 airplane. The purpose of these flights were to conduct flight research related to the Aviation Safety Program, including fault and failure detection, real-time system identification, adaptive control, and others. Maneuvers included small perturbations about steady flight conditions, high angle of attack stalls, post-stall flight, rapid global maneuvering, and others.

B. LOC Flight Envelopes

The variables needed to construct the LOC flight envelopes described in Ref. [3] are listed in Table 1. Generation of the five LOC envelopes were as follows.

Adverse Aerodynamics (AA) Envelope. This envelope maps normalized sideslip angle β_{norm} against normalized angle of attack α_{norm} . Data for α and β were obtained by averaging flow angle measurements at the two wing tips. Sideslip was normalized by the maximum demonstrated cross-wing landing (10 deg) and angle of attack was normalized by the stall-warning angle of attack (12 deg).

Unusual Attitude (UA) Envelope. This envelope maps the bank angle against the pitch angle. Bank and pitch angle data were obtained from measurements of the attitude solution by the inertial measurement units.

Structural Integrity (SI) Envelope. This envelope maps normalized airspeed V_{norm} against the normal load factor n .



Fig. 3 T-2 aircraft in flight (credit: NASA Langley Research Center)

Table 1 Variables and measurements used to define LOC envelopes.

Symbol	Name	Unit
α	Angle of attack	deg or rad
β	Sideslip angle	deg or rad
V	Airspeed	ft/s
ϕ	Roll angle	deg or rad
θ	Pitch angle	deg or rad
δ_e	Elevator (pitch control)	deg or rad
δ_a	Aileron (roll control)	deg or rad

The normalized airspeed was computed as

$$V_{norm} = \frac{V_e - V_{sw}}{V_{mo} - V_{sw}}, \quad \text{for flaps up} \quad (10a)$$

$$V_{norm} = \frac{V_e - V_{sw}}{V_{fe} - V_{sw}}, \quad \text{for flaps down} \quad (10b)$$

where V_e was the equivalent airspeed, $V_{sw} = 50$ ft/s was the stall warning equivalent airspeed in 1-g flight, $V_{mo} = 200$ ft/s was the maximum operating equivalent airspeed (flaps up), and $V_{fe} = 160$ ft/s was the maximum flaps extended operating equivalent airspeed (flaps down). Data for the equivalent airspeed were obtained by averaging true airspeed measurements from the two wing tips. Data for the normal load factor were obtained from measurements of a vertical accelerometer installed near the aircraft nominal center of mass.

Dynamic Pitch Control (DPC) Envelope. This envelope maps pitch axis control authority against dynamic pitch attitude. The pitch axis control authority is the elevator deflection angle, obtained from potentiometer measurements, normalized by the maximum deflection angles (-30 to 20 deg). The dynamic pitch attitude was defined

$$\theta' = \theta + \dot{\theta} \quad (11)$$

where pitch angle measurements were again obtained from the inertial measurement unit. The pitch angle rate was computed as [31]

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (12)$$

where the pitch rate q and yaw rate r were obtained from rate gyroscopes installed near the aircraft nominal center of mass.

Dynamic Roll Control (DRC) Envelope. This envelope maps roll axis control authority against dynamic roll attitude. The roll axis control authority is the aileron deflection angle, obtained from potentiometer measurements, normalized by the maximum deflection angles (-20 to 20 deg). The dynamic roll attitude was defined

$$\phi' = \phi + \dot{\phi} \quad (13)$$

where roll angle measurements were again obtained from the inertial measurement unit. The roll angle rate was computed as [31]

$$\dot{\phi} = p \tan \theta (q \sin \phi + r \cos \phi) \quad (14)$$

similar to Eq. (12).

Additional LOC Variables. Additional variables can be included for other LOC envelopes. The variables weight, fuel weight, initial altitude, flap/gear position, and center of mass location were monitored, but were never exceeded for the T-2 flight data.

Mapping of flight test data onto these envelopes is shown for an example case in Fig. 4. The envelopes are drawn using the descriptions in Ref. [3] and the values mentioned above. The flight data is from the T-2 Flight 16. Green markers indicate when the airplane is within the respective flight envelopes, and red markers indicate the airplane is outside the envelope. Only up-and-away flight data is shown; take-off, landing, and check-out segments have been removed. The data were recorded at 50 Hz and span almost 17 minutes of flight. This flight included several slow stall

Airspeed	$(\frac{\text{ft}}{\text{sec}})$	Elevator	(deg)
Sideslip angle	(deg)	Aileron	(deg)
Angle of attack	(deg)	Rudder	(deg)
RPY Rates	$(\frac{\text{deg}}{\text{sec}}, \frac{\text{deg}}{\text{sec}}, \frac{\text{deg}}{\text{sec}})$	Inbound-Outbound Flap	(deg)
RPH Angles	(deg,deg,deg)	Dynamic Pressure	$(\frac{\text{lb}}{\text{ft}^2})$
Accelerations	(g,g,g)	Mach Number	
Air Density	$(\frac{\text{slug}}{\text{ft}^3})$	Altitude	(ft)
Mass	(slug)		

Table 2 T-2 Flight percepts that serve as input to the neural network

maneuvers, intended for system identification of global aerodynamic models, where the pilot slowly pulled back on the stick while multi-axis excitations were injected to the control surfaces. As this happens, the aircraft climbs, increases in pitch angle, and decreases in airspeed. After stalls occurred, the aircraft nose dropped and the aircraft rolled. The pilot then regained airspeed and leveled the wings.

Figure 5 shows flags indicating whether the airplane was inside (green marker, = 1) or outside (red marker, = 0) each of the flight envelopes, over time. For T-2 Flight 16, the airplane was inside the five LOC envelopes for 97%, 81%, 98%, 94%, and 67% of the data shown, respectively. Much of the excursions outside these envelopes occurred during the slow-stall maneuvers and the subsequent recoveries. Some of the excursions may have been exacerbated by the moderate to severe turbulence experienced during a portion of the flight [32]. Despite the large excursions experienced during this flight, LOC was (correctly) never indicated because three of the envelopes were never violated.

C. Loss-of-Control Detection using Reconstruction Probability

The T-2 flight data contains 140 output flight variables for measurement. We identify 22 variables that vary during individual missions, and that should indicate qualitative changes in the state of the vehicle during flight. These flight variables are described in Table 2. For each flight, we resample the respective time series (using mean measurements) to 1 Hz, such that we do not raise alerts for sub-second events potentially due to effects of measurement noise or turbulence. We label every observation in the T-2 flight data with a binary vector based on whether the aircraft at the time of the observation is in (1) or out (0) of the envelopes referred to in Fig. 6. Fig. 6(a) demonstrates that only seven of the envelopes are ever violated over the course of these 58 flights. Fig. 6(b) shows the distribution of violated envelopes. Normal measurements overwhelm the out-of-envelope ones by a large margin. And there is a significantly small number of LOC measurements (< 3 violated envelopes). This is amenable to triggering anomalies for rare events. However, it does not permit our CVAE to be properly trained on certain out-of-envelope flight behavior for the T-2 aircraft.

We use the Keras deep learning framework [33] to implement the CVAE specified in Section IV.B. We trained two CVAEs: one comprised of GRU layers and one comprised of LSTM layers. Both CVAEs have a latent space of size 11. Each network was trained for a maximum of 10,000 epochs, with early stopping if the validation loss of the network was non-decreasing over 2,000 epochs. During each epoch, 88% of the flights were used for training and 12% of the flights were used for validation. For each epoch, we shuffled the order in which the training flights were processed. The training was performed on the NASA Langley Research Center OCIO Computing Cluster, using an NVIDIA Tesla K-40 GPU.

We continue to explain our measurements for experimentation using T-2 Flight 16. We reconstruct the Flight 16 observations 1,000 times and plot the Kernel density estimation of the reconstruction probabilities in Fig. 7. For an individual flight mission, if more than three envelopes were shown to have been violated, we labeled observations at the time of these violation as LOC observations. Note the distinct separation in the density of reconstruction probabilities for LOC observations and non-LOC observations by both the GRU-CVAE (Fig. 7(a)) and the LSTM-CVAE (Fig. 7(b)) models. We highlight the median reconstruction probabilities for LOC events that occurred during Flight 16 for both types of observation as candidates for the LOC detector threshold.

When $\tau = 69\%$, the GRU-based CVAE detects 100% of LOC observations, whereas if $\tau = 72\%$, the LSTM-based CVAE detects 97% of LOC observations. 50% of LOC observations would be detected by the GRU and LSTM-based detectors for $\tau = 18\%$ and $\tau = 37\%$, respectively. Because these observations are 1 Hz samples, the GRU-based detector still practically performs well if it is set to the median reconstruction probability of the LOC observations. This is illustrated in the line graph of Flight 16 reconstruction probabilities over time shown in Fig. 8. In a real scenario, if our

detector alerted the C2 agent that the aircraft was in an LOC state at the highlighted times, the agent would perform further inspection in and around that time. Both CVAEs capture the 10 distinct periods in which the aircraft was in an LOC state for this flight. The false positives for LOC in this figure correspond roughly to times at which the aircraft was operating outside of two flight envelopes (see Fig. 11), which, in reality, may be identified as softer LOC periods. Further investigation of the flight at these time periods would need to be done to uncover the ground truth. The dataset is imbalanced, so we show both the ROC and Precision-Recall curves for Flight 16 in Fig. 9 with τ set to the medians of reconstruction probability for LOC observations, non-LOC observations, and the midpoint between these observations. Both the LSTM-CVAE and GRU-CVAE detectors separate the LOC observations from non-LOC observations well at the lower values of τ . From a precision-recall standpoint, the largest τ performs best.

Fig. 10 shows the performance of both CVAEs when breaking down the density of reconstruction probabilities by envelope. In cases where either 0 or 1 envelopes are violated, the reconstruction probabilities are very high, rarely dropping below 60%. The CVAE’s reconstruction capability, for this flight, makes a clear distinction between these observations and those where more envelopes are violated.

D. Loss-of-Control Detection using Gaussian Shift

Here, we continue to examine our measurement of Flight 16 by explicitly measuring the encodings of the GRU-CVAE and LSTM-CVAE. For context, Fig. 11 shows the envelope statuses for Flight 16 over the course of the mission. Figure 12 shows a pair-wise plot of the last six variables of the GRU-CVAE latent space when encoding the observations of Flight 16. Each pair-wise plot forms a crescent shape, which could potentially be clustered and mapped to the number of violated envelopes at that encoding, with quantifiable accuracy. For the raw means of the latent space, encodings that represent 0–1 envelope violations are highly clustered. Encodings that correspond to an observation where 2 envelopes have been violated form much of the arc of the crescent. And encodings corresponding to additional envelope violations are once again clustered together, but away from those corresponding to non-LOC observations. For the LSTM case (Fig. 13), we see a fainter structure of an arc in each pair plot. This space could easily be bifurcated into separate parts of the mission.

We directly compute and examine the gradients of the JS-distance ($\hat{D}_{JS}$). In these results, the gradient is computed using NumPy v1.2 [34], which computes the gradient of an N-dimensional array. This implementation The gradients for flight 16 turned out to be very small, as illustrated in Fig. 14. We measured the number of envelope shifts, whether the aircraft slips into or out of an envelope, over the course of the mission and measured the average gradient of the JS-distance of the latent space with respect to this number. Formally, let e_t be the latent space encoding, expressed as a Gaussian distribution, at time t . We compute $D_{JS}(e_t||e_{t+1})$ and $\hat{D}_{JS}(e_t||e_{t+1})$ and compare it to the envelope shift at time $t + 2$ (the next second). The plot shows that the average JS-distance for Flight 16 varies significantly (relative to this small scale) for the encodings of the GRU-CVAE when more than 1 or more envelopes change. The curve is much more flat for $\hat{D}_{JS}$ until more than 2 envelopes change in the next second. In contrast, $D_{JS}(e_t||e_{t+1})$ and $\hat{D}_{JS}(e_t||e_{t+1})$ roughly follow the same trend for the LSTM-CVAE’s latent space. In addition, we measured simpler envelope shifts, labeling each of these measurements based on whether the vehicle shifted in or out of any envelope at time $t = 2$. In the case of this flight, the GRU-CVAE shows relatively larger differences in both measurements when a state change occurs in the next step compared to when a state change does not occur.

Finally, we applied the t-distributed Stochastic Neighbor Embedding (t-SNE) [35] algorithm to visualize the similarity of latent space means across all 11 dimensions of our CVAEs, leveraging the underlying low-dimensional manifolds that form the space. We provide the latent space means of both CVAEs as input to the t-SNE algorithm and run it for maximum of 10,000 iterations, stopping if there is no progress for 300 iterations. In Fig. 15, we plot the 2-dimensional embedding of our 11-dimensional latent space means using t-SNE.* The embeddings show much clearer segments and shapes in the latent space. The low-dimensional embedding of the GRU-CVAE latent space maintains its crescent structure. The LSTM-CVAE reveals several distinct areas of encoding on the low-dimensional embedding. With additional resampling, each of the clusters of the embedding for the LSTM-CVAE latent space could reveal a consistent mapping of 1-2 envelopes on each cluster. And we could investigate a mapping of their straight-line structure to timesteps.

*Note that t-SNE is a probabilistic approach to dimension reduction, rather than a mathematical function that generates an exact solution. So exact regeneration of these plots are unlikely.

E. Results

Here, we describe the results obtained by executing the measurements from the previous sections across all 58 T-2 flights and maneuvers. We use both CVAEs to reconstruct the time series for each of the 58 flight missions and 34 maneuvers. We capture both the latent space encodings, expressed as Gaussian distributions, and reconstruction probabilities for every observation. Like in the individual case of Flight 16 in V.C, we reconstructed each flight and maneuver 1,000 times to obtain the reconstruction probability for each observation under this scenario.

Fig. 17 shows a histogram of the reconstruction probabilities. The GRU-CVAE seems to have a better separation of reconstruction probabilities between LOC and Non-LOC observations. Overall, the LSTM-CVAE has better accuracy for reconstructing its inputs. However, most of the LOC observations for the GRU-CVAE have reconstruction probabilities below 50%. This is much more spread out for the LSTM-CVAE. This is illustrated in the KDE plots comparing the two CVAEs in Fig. 16. Using the non-LOC median for the detector threshold, τ , would result in catching 97% of LOC observations when using the GRU-CVAE and 94% for the LSTM-CVAE. A more conservative detector would use the median GRU-CVAE reconstruction probability for non-LOC observations, to ensure a high rate of true positives. Fig. 18 shows the performance of an LOC detector based on each of the CVAEs. The structure of the performance curves reflect the structure seen when examining the individual Flight 16. The detector based on the GRU-CVAE demonstrates a higher AUROC and AUC for the precision-recall curve for each candidate detector. Overall, these results show that a detector that differentiates between the LOC and Non-LOC observations of the T-2 aircraft would perform better using the GRU-CVAE.

Fig. 19 shows a breakdown of the density of CVAE reconstruction probabilities based on the number of envelopes that the aircraft is operating in. For both CVAEs, we see a clear separation in the density of reconstruction probabilities for observations that occur when the aircraft is operating within all 10 envelopes and observations that occur when the aircraft is operating within only 6 envelopes. For the GRU-CVAE, there is significant overlap between the aircraft operating within 10 envelopes and 9 envelopes, and when the aircraft is operating within 6 envelopes and 7 envelopes. And for the LSTM-CVAE, there is significant overlap when the aircraft is operating between 7 and 9 envelopes. The bottom two figures of Fig. 19 show the min-max normalized distribution of reconstruction probability categorized by the number of envelopes that the aircraft is operating within. The figures show the spread of reconstruction probabilities across each category. Most observations for the aircraft operating within all envelopes are on the higher side of that envelope category's distribution. In contrast, observations that correspond to the aircraft operating only within 6 envelopes trend toward the lower side of its distribution.

We observe the distribution of CVAE latent space means across all flights by labeling each encoding with the number of envelopes that are violated in the observation *after* it and whether an LOC observation occurred after it. We reduce the latent space means to 3 dimensions using the t-SNE algorithm. Fig. 20 shows a 3D plot of the low-dimensional manifold learned by t-SNE for all flights. The 3-dimensional t-SNE of the latent space encodings seems to approximate a 3-sphere[36] (accounting for temporal order as the 4th dimension). The encodings that have LOC observations after them are clustered at various points on the outer section of the sphere. As far as the number of envelopes that are violated after an encoding, encodings where no change is observed afterwards form much of the center of the sphere. Most encodings in which changes to the flight envelopes occur afterwards are on the outer edges. A sufficient mapping (one that quickly be computed) of the raw means of the latent space to this 3-sphere can be used to indicate that a change in envelope status is occurring.

Finally, we observe any indications that a shift in the gradient of the JS-distance ($\hat{D}_{JS}$) may be impacting the envelope statuses. Fig. 21 shows how $\hat{D}_{JS}$ is distributed, categorized by whether or not an envelope change occurred after the encoding. The KDEs highly overlap for the GRU-CVAE and show slight discrepancy for the LSTM-CVAE. The mean values of $\hat{D}_{JS}$ do not appear to have significant differences between encodings where a state change follows and the encodings where they do not. We do see some non-negligible differences in Fig. 22, where we assess the number of envelopes that change after the encoding (as opposed to whether there was a change). The LSTM-CVAE shows an increasing gradient (and raw JS-distance) over a growing number of envelope changes. Overall, the LSTM-CVAE shows better utility in identifying if large shifts in envelope statuses will occur in a subsequent observation.

F. Discussion

The reconstruction probability seems to be very good at telling us when the shift to Loss-of-Control has occurred (within 2–4 seconds of the shift). As the majority of flight observations follow the scope of normal physics, it makes sense that it would have an issue reconstructing observations that push up against the limits of a vehicle's physics (i.e. violating a larger number of flight envelopes). However, we have already seen that we can calculate when loss-of-control

has happened directly (by simply looking at our observations and the envelope constraints). So there are two real applications that we see for this methodology:

- **Design-Time Definition of the Flight Envelope:** For a vehicle with an unknown or uncertain flight envelope, the reconstruction probability can be used to explore the boundaries of predictable behavior at a moment in time. Monte Carlo-style simulations or testing can occur during design time to aid in the definition of the boundaries of the envelope.
- **Dynamic Contingency Planning:** Uncertainty about how to reconstruct observations can be useful when the vehicle encounters a new, unforeseen situation or phenomena. This is particularly useful when we need to explore the short-term and long-term effects of optional contingencies that place the vehicle in a new state of flight. Construction of a new plan can include an estimate of the new physical observations that are likely to be seen during that plan (desired velocities, accelerations, angles, etc.). The reconstruction probability can be used to determine the certainty with which the contingency will be carried out without loss-of-control.

Looking back at Fig. 12, each pair-wise plot of the latent space forms a crescent shape. The pair-wise plots of Fig. 13 can easily be bifurcated. In many cases, it appears that one of the clusters of this bifurcation would result in a straight line. Both of these formations in the latent space are likely due to the temporal features of our CVAEs. An update from the previous encoding inference is likely causing a small shift of the mean of the encoding from timestep to timestep. In future work, we will explore the mapping of the shift in the latent space means to that memory. This can lead to a method of quantifying how much the network is incorporating information from previous observations into the encoding. Understanding that mapping is the key to using Gaussian shift as a reliable technique.

The distributions of reconstruction probabilities in Figs. 19(c) and 19(d) yield information that can be used to establish in-envelope detectors that show when something anomalous is occurring in the context of the number of envelopes that the aircraft is in. Categories of such anomalies are likely to include a category that signifies the aircraft is shifting in or out of an envelope. Further investigation of qualitative activities of the aircraft during these observations can be used as a labeling for the classifiers that comprise such detectors.

Overall, our analysis showed that the LSTM-CVAE has better utility in understanding whether a larger shift in envelope status will occur in subsequent observations. However, the scale of actual numbers in this analysis is quite small. A transform may need to occur to better highlight the discrepancies in envelope status over subsequent observations.

VI. Conclusion

In this work, we analyzed the utility of an LOC detector, based on the RNN-CVAE, in identifying periods of loss-of-control for a generic, transport-type aircraft. The detector demonstrated an ability to distinguish loss-of-control states for observations at 1s intervals with a true positive rate of 97%. Lower thresholds for the detector provide a more practical true negative rate, providing a mechanism to accurately address loss-of-control without significant false alarms. We compared the utility of the LSTM and GRU as structural variants for the underlying CVAE of this detector. The GRU-CVAE showed better performance in an ability to distinguish LOC observations from non-LOC observations.

We also assessed the structure of latent space means for the CVAE. We identified what are Gaussian means that are sufficiently separated in terms of their relation to loss-of-control. Our analysis showed that the lower-dimensional embedding of the latent space forms an approximate 3-sphere. Encoding corresponding to in-envelope flight are highly clustered towards the center of this 3-sphere, whereas encoding corresponding to 1 or more envelopes being violated seem to form the outer layers. Future work should analyze how to quickly approximate this 3-sphere and characterize its regions.

We introduced a technique for directly assessing the latent space to identify if an impending change in the aircraft's flight envelope status (whether it is moving in or out of certain envelopes) is happening. The analysis showed small-scale differences in the gradient of the JS-distance between subsequent encoding of the latent space. We showed that changes in this gradient more in the LSTM-CVAE when an envelope shift is occurring in subsequent timesteps. A transform of the gradient of the JS-distance will need to occur to increase the level of fidelity under comparison.

Acknowledgments

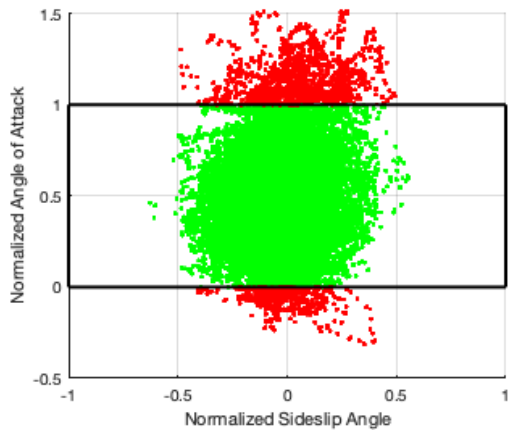
This research was supported by the NASA Aeronautics Research Mission Directorate (ARMD), Transformative Tools and Technologies (TTT) project, under the Autonomous Systems / Intelligent Contingency Management subproject. The T-2 flight data used was generated by the NASA Langley Research Center AirSTAR team under the NASA Aviation

Safety Program, Vehicle Systems Safety Technologies (VSST) project.

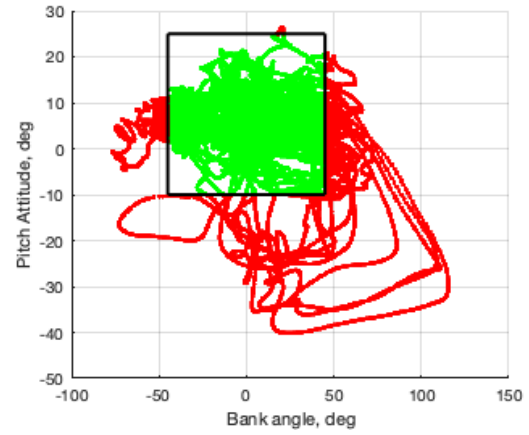
References

- [1] Anon., “Advanced Air Mobility Overview,” <https://www.nasa.gov/aam>, accessed May 2020.
- [2] Scheidt, D., “Command and Control of Autonomous Unmanned Vehicles,” *Hand-28 book of Unmanned Aerial Vehicles*, 2015, pp. 1273–1298.
- [3] Wilborn, J., and Foster, J., “Defining Commercial Transport Loss-of-Control: A Quantitative Approach,” 2004. <https://doi.org/10.2514/6.2004-4811>.
- [4] Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M., “Semi-supervised learning with deep generative models,” *Advances in neural information processing systems*, 2014, pp. 3581–3589.
- [5] Sohn, K., Lee, H., and Yan, X., “Learning structured output representation using deep conditional generative models,” *Advances in neural information processing systems*, 2015, pp. 3483–3491.
- [6] Shin, J., “The NASA Aviation Safety Program: Overview,” NASA TM-2000-209810, March 2000.
- [7] Belcastro, C., and Foster, J., “Aircraft Loss-of-Control Accident Analysis,” 2010. <https://doi.org/10.2514/6.2010-8004>.
- [8] Belcastro, C., and Klyde, D., “Special Issue on “Aircraft Loss of Control”,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 4, 2017, pp. 733–1084. <https://doi.org/10.2514/1.G002815>.
- [9] Tekles, N., Chongvisal, J., Xargay, E., Choe, R., Talleur, D., Hovakimyan, N., and Belcastro, C., “Flight Envelope Protection System for NASA’s Transport Class Model,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 4, 2017, pp. 863–877. <https://doi.org/10.2514/1.G001728>.
- [10] Ali, A. H., Markarian, G., Tarter, A., and Kölle, R., “Feasibility demonstration of diagnostic decision tree for validating aircraft navigation system accuracy,” *Journal of aircraft*, Vol. 47, No. 6, 2010, pp. 2096–2102.
- [11] Afridi, M. J., Awan, A. J., and Iqbal, J., “AWG-Detector: A machine learning tool for the accurate detection of Anomalies due to Wind Gusts (AWG) in the adaptive Altitude control unit of an Aerosonde unmanned Aerial Vehicle,” *2010 10th International Conference on Intelligent Systems Design and Applications*, 2010, pp. 1125–1130.
- [12] Smart, E., Brown, D., and Denman, J., “A Two-Phase Method of Detecting Abnormalities in Aircraft Flight Data and Ranking Their Impact on Individual Flights,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 3, 2012, pp. 1253–1265.
- [13] Chen, Y., Wang, B., Liu, W., and Liu, D., “On-line and non-invasive anomaly detection system for unmanned aerial vehicle,” *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 2017, pp. 1–7.
- [14] Liu, D., Wang, Z., Wang, S., Pang, Y., and Liu, L., “UAV sensor data anomaly detection using predictor with uncertainty estimation and its acceleration on FPGA,” *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2018, pp. 1–6.
- [15] Burrows, B. J., and Allaire, D. L., “A comparison of naive bayes classifiers with applications to self-aware aerospace vehicles,” *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 3819.
- [16] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., “Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1,” MIT Press, Cambridge, MA, USA, 1986, Chap. Learning Internal Representations by Error Propagation, pp. 318–362. URL <http://dl.acm.org/citation.cfm?id=104279.104293>.
- [17] Kingma, D. P., and Welling, M., “Auto-Encoding Variational Bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [18] Rezende, D. J., Mohamed, S., and Wierstra, D., “Stochastic backpropagation and approximate inference in deep generative models,” *arXiv preprint arXiv:1401.4082*, 2014.
- [19] An, J., and Cho, S., “Variational autoencoder based anomaly detection using reconstruction probability,” *Special Lecture on IE*, Vol. 2, No. 1, 2015.
- [20] Wang, X., Du, Y., Lin, S., Cui, P., and Yang, Y., “Self-adversarial variational autoencoder with gaussian anomaly prior distribution for anomaly detection,” *arXiv preprint arXiv:1903.00904*, 2019.
- [21] Zhao, S., Song, J., and Ermon, S., “InfoVAE: Information Maximizing Variational Autoencoders,” *arXiv preprint arXiv:1706.02262*, 2017.

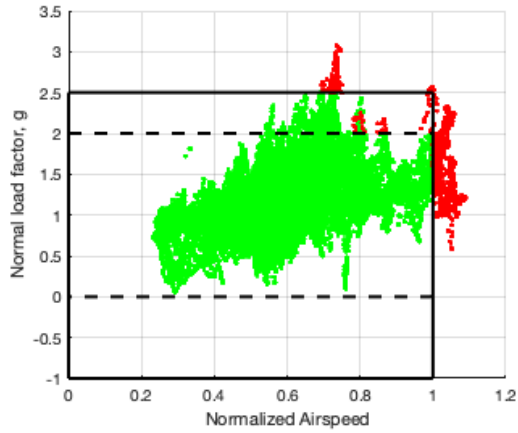
- [22] Guo, Y., Liao, W., Wang, Q., Yu, L., Ji, T., and Li, P., “Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach,” *Asian Conference on Machine Learning*, 2018, pp. 97–112.
- [23] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., and Lloret, J., “Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT,” *Sensors (Basel, Switzerland)*, Vol. 17, No. 9, 2017.
- [24] Pol, A., Berger, V., Cerminara, G., Germain, C., and Pierini, M., “Anomaly Detection With Conditional Variational Autoencoders,” *ICMLA 2019-18th IEEE International Conference on Machine Learning and Applications*, 2019.
- [25] Pol, A., Berger, V., Cerminara, G., Germain, C., and Pierini, M., “Trigger Rate Anomaly Detection with Conditional Variational Autoencoders at the CMS Experiment,” *Machine Learning and the Physical Sciences Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [26] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y., “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.
- [27] Hochreiter, S., and Schmidhuber, J., “Long Short-Term Memory,” *Neural Comput.*, Vol. 9, No. 8, 1997, p. 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>, URL <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [28] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S., “Self-normalizing neural networks,” *Advances in neural information processing systems*, 2017, pp. 971–980.
- [29] Shao, H., Kumar, A., and Thomas Fletcher, P., “The riemannian geometry of deep generative models,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 315–323.
- [30] Shukla, A., Uppal, S., Bhagat, S., Anand, S., and Turaga, P., “Geometry of deep generative models for disentangled representations,” *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, 2018, pp. 1–8.
- [31] Morelli, E., and Klein, V., *Aircraft System Identification: Theory and Practice*, 2nd ed., Sunflyte, Williamsburg, VA, 2016.
- [32] Morelli, E., and Cunningham, K., “Aircraft Dynamic Modeling in Turbulence,” 2012. <https://doi.org/10.2514/6.2012-4650>.
- [33] Chollet, F., et al., “Keras,” <https://keras.io>, 2015.
- [34] Oliphant, T. E., *A guide to NumPy*, Vol. 1, Trelgol Publishing USA, 2006.
- [35] Maaten, L. v. d., and Hinton, G., “Visualizing data using t-SNE,” *Journal of machine learning research*, Vol. 9, No. Nov, 2008, pp. 2579–2605.
- [36] Lemaître, G., *Quaternions et espace elliptique*, ex aedibus Academicis, 1948.



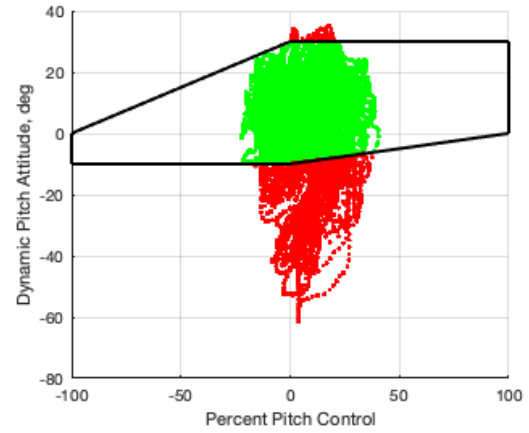
(a) Adverse aerodynamics envelope.



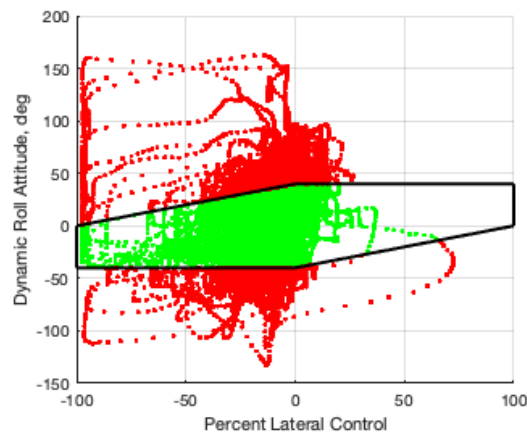
(b) Unusual attitude envelope.



(c) Structural integrity envelope.



(d) Dynamic pitch control envelope.



(e) Dynamic roll control envelope.

Fig. 4 LOC envelopes and data from T-2 Flight 16. Green markers indicate flight within the envelope, whereas red markers indicate flight outside the envelope.

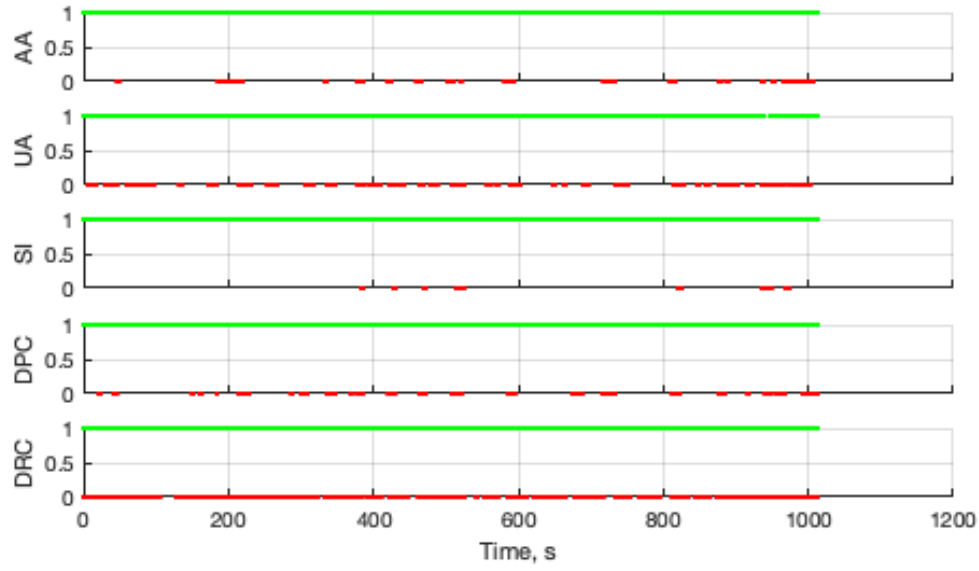


Fig. 5 LOC envelope excursion flag time histories for T-2 Flight 16.

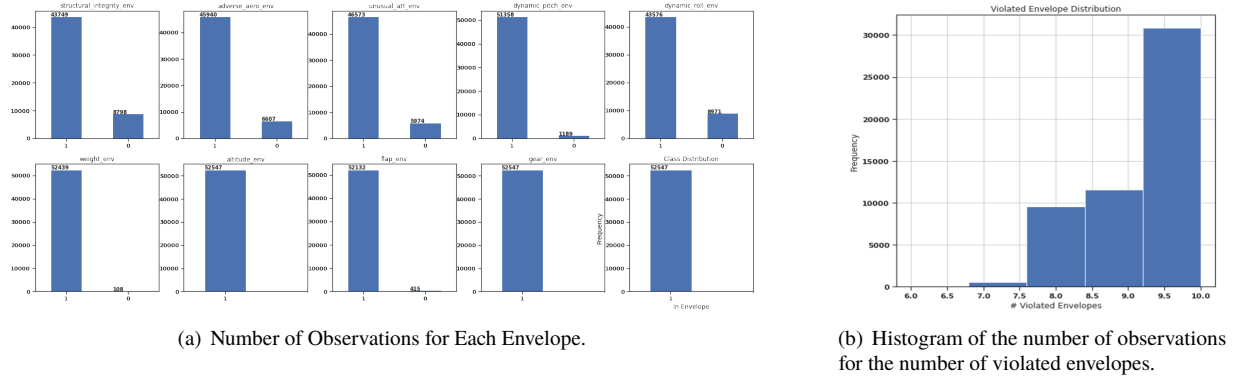


Fig. 6 Breakdown of Envelope Status over all Observations for 58 Flights of the T-2 Aircraft

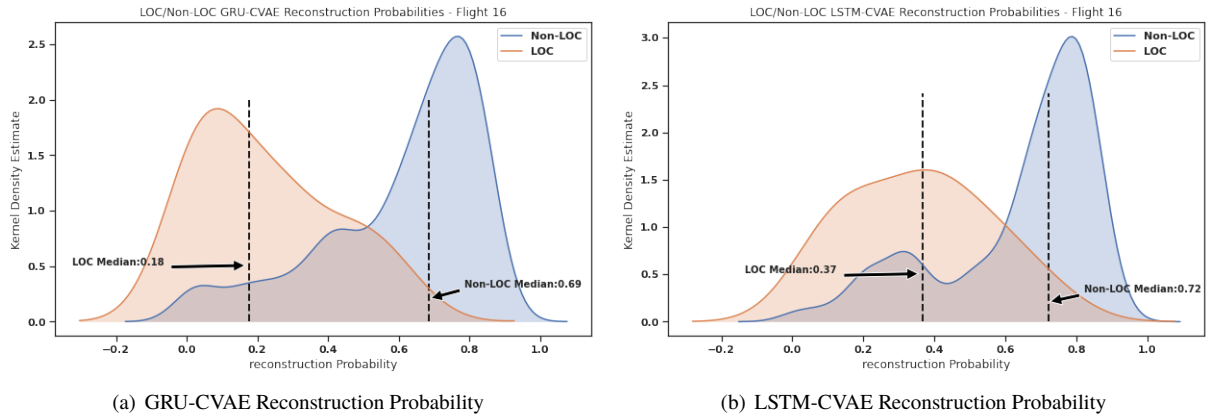


Fig. 7 Reconstruction Probability for Flights 16 the T-2 Aircraft (by LOC/Non-LOC Observations)

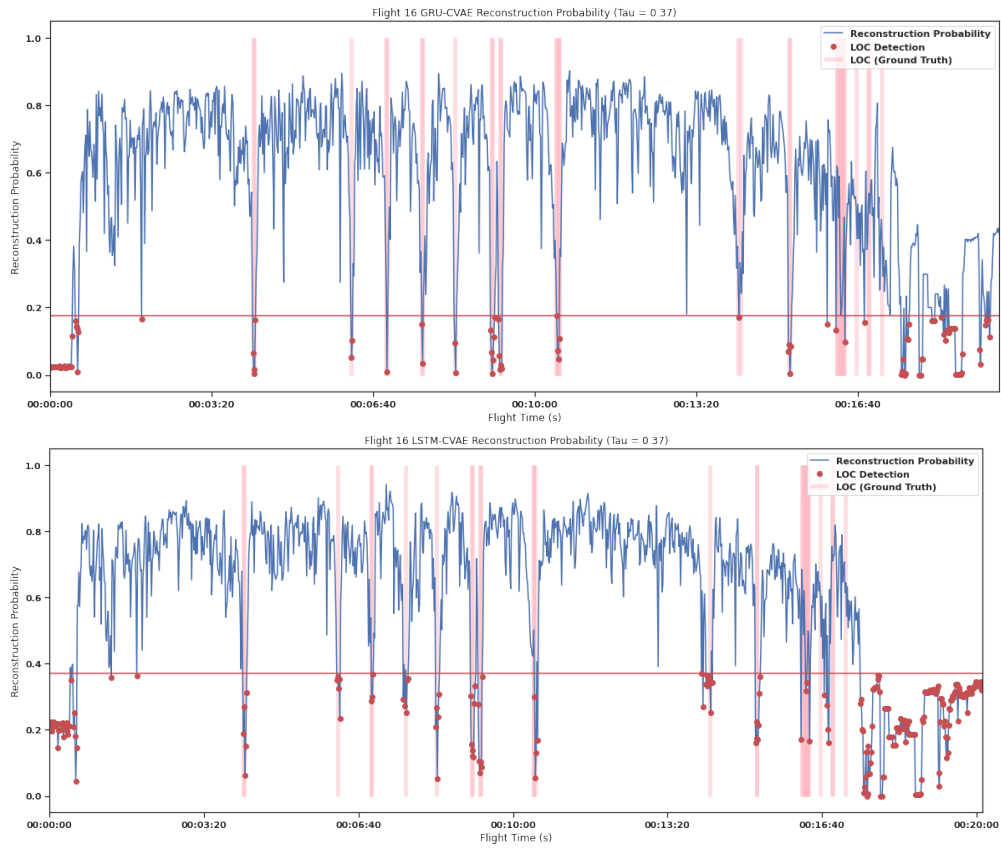
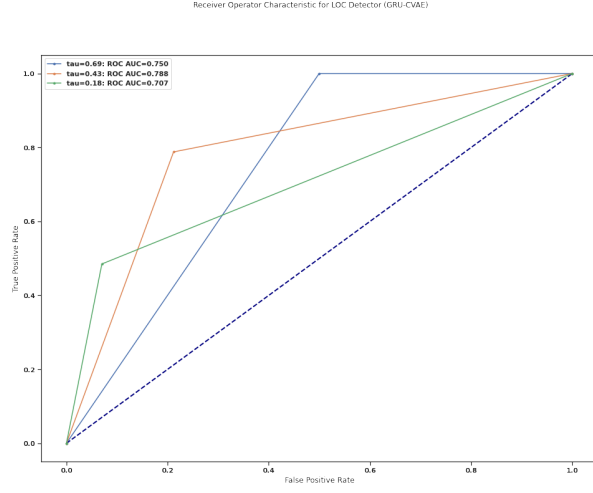
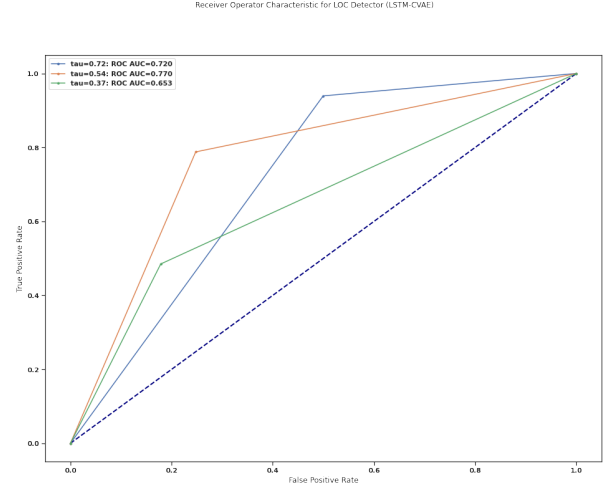


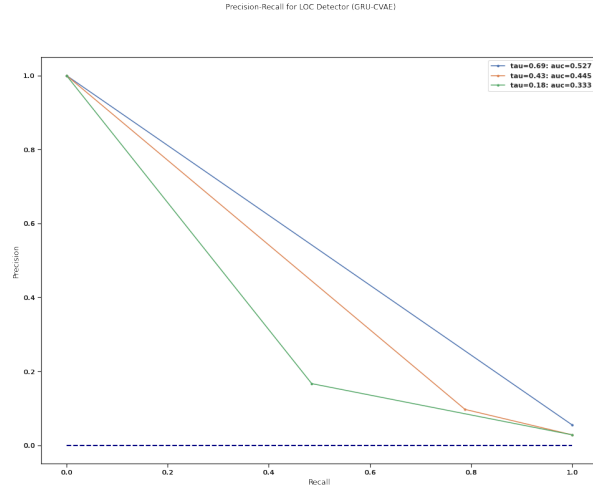
Fig. 8 CVAE Reconstruction Probability over time For Flight 16.



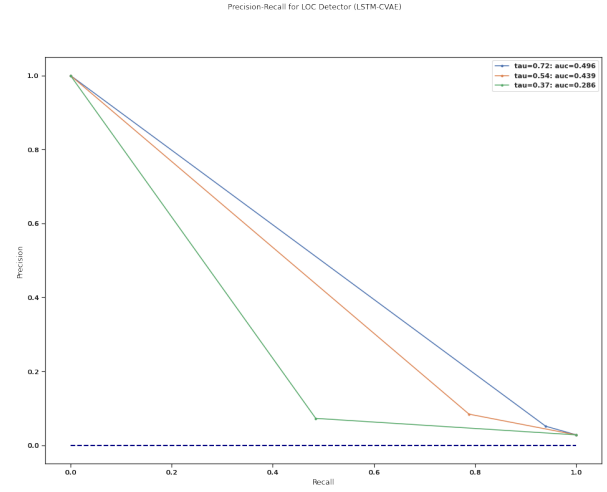
(a) ROC Curve for LOC detector based on GRU-CVAE



(b) ROC Curve for LOC detector based on LSTM-CVAE

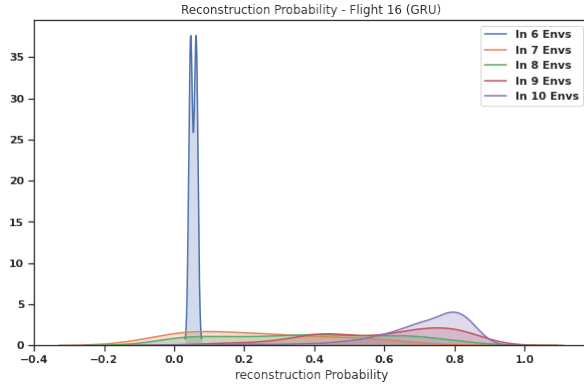


(c) Precision-Recall Curve for LOC detector based on GRU-CVAE

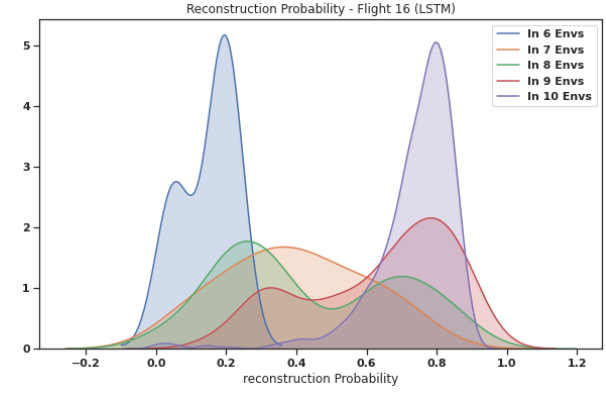


(d) Precision-Recall Curve for LOC detector based on LSTM-CVAE

Fig. 9 Flight 16 Precision-Recall Curves for LOC Detectors based on Reconstruction Probability.



(a) GRU-CVAE Reconstruction Probabilities Per Envelope



(b) LSTM-CVAE Reconstruction Probabilities Per Envelope

Fig. 10 Kernel density estimate of reconstruction probability for flights 16 for the T-2 Aircraft.

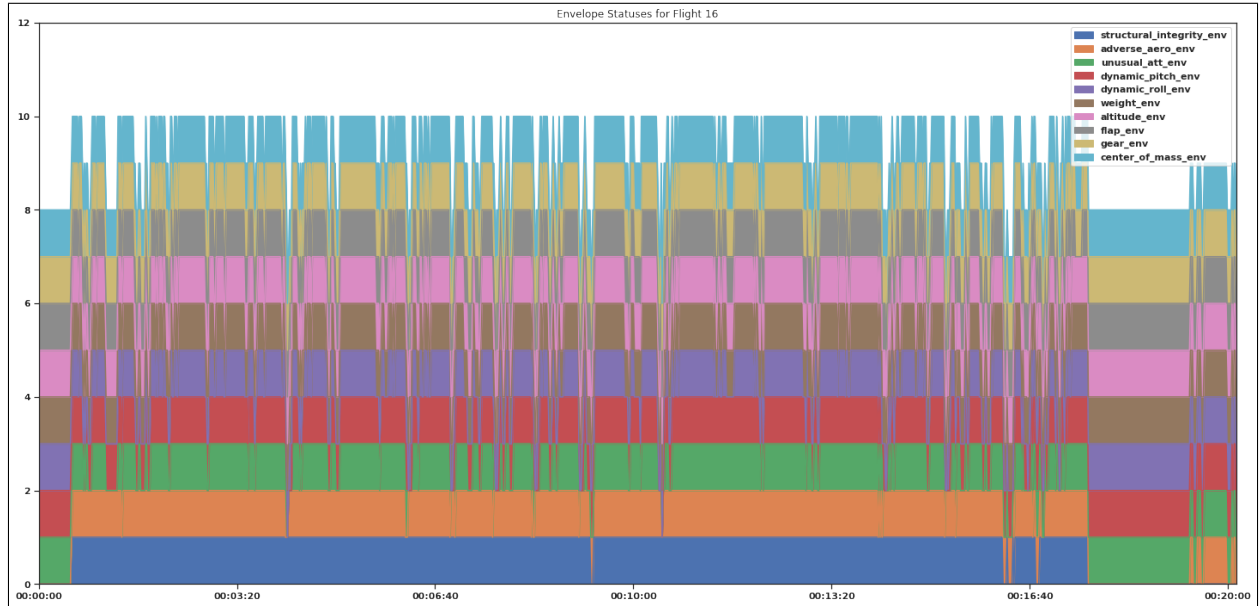


Fig. 11 Envelope Statuses for Flight 16 over Time.

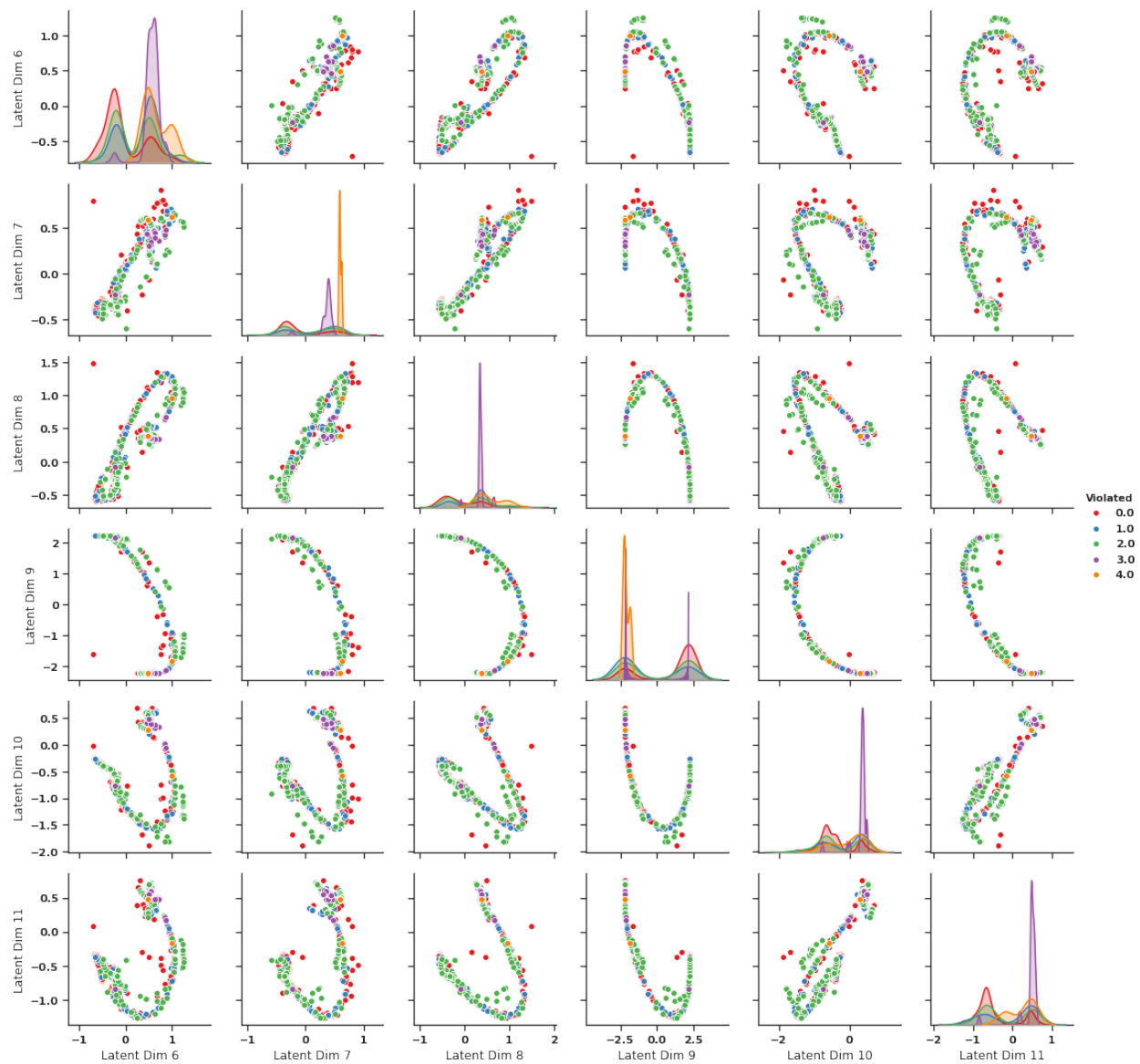


Fig. 12 Pair plot showing the GRU-CVAE encodings (means) of Flight 16 for the latter 6 latent variables. Each encoding is color-coded by the number of violated flight envelopes.

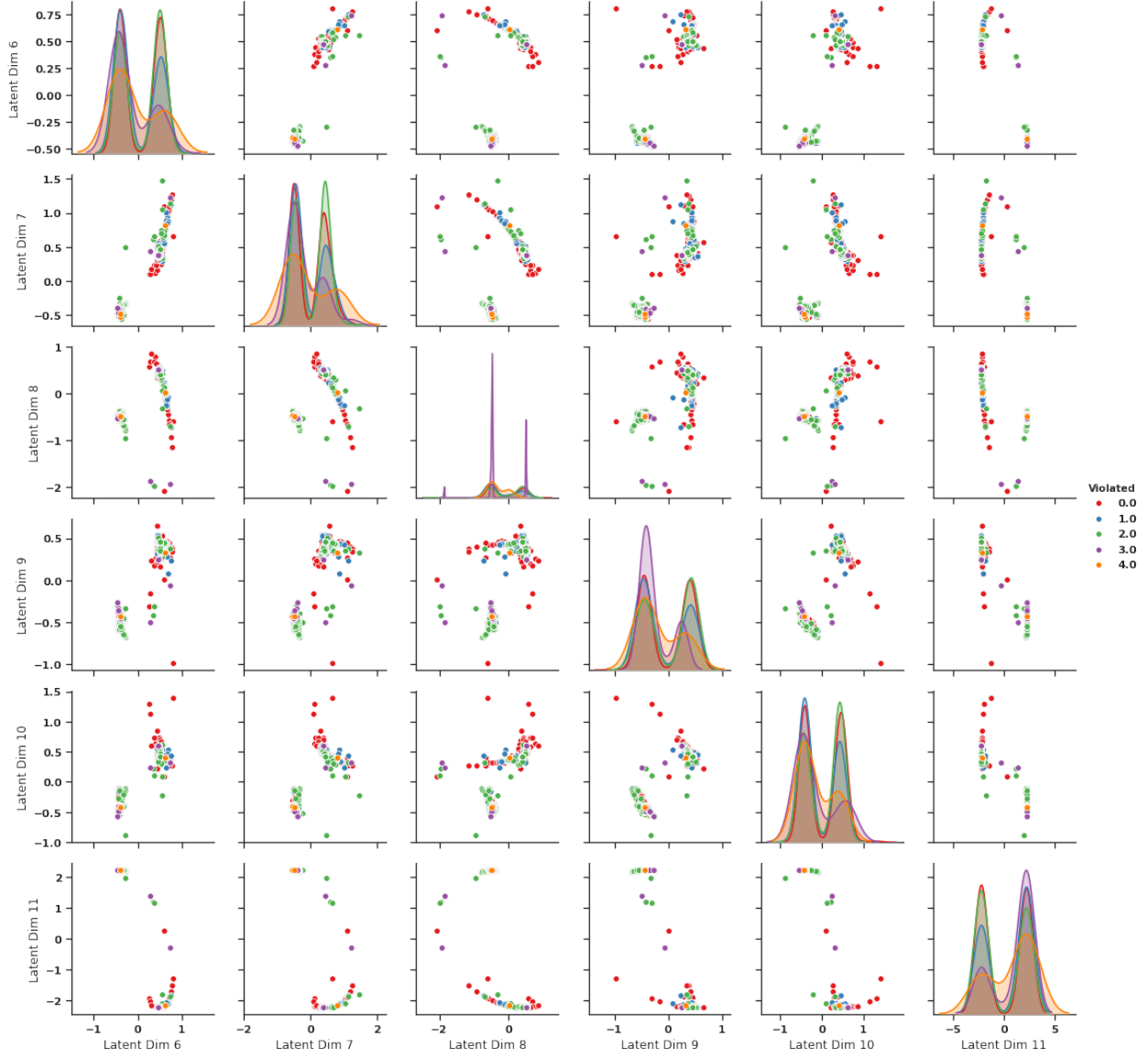
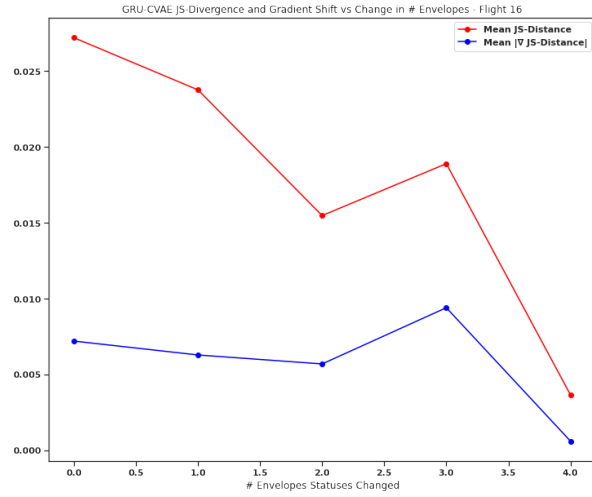
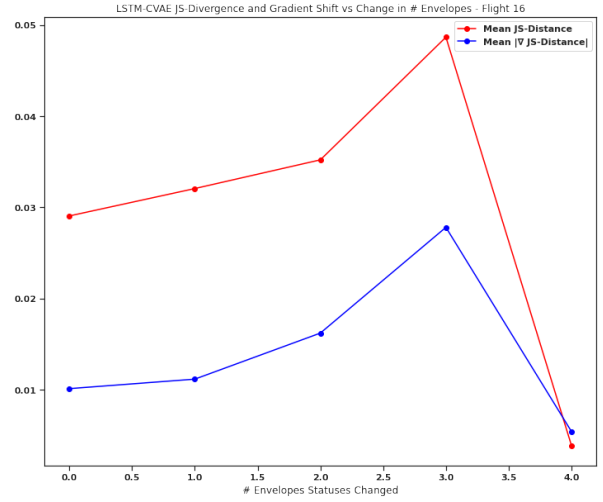


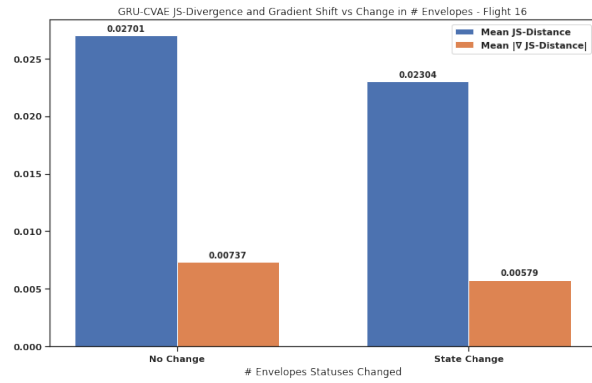
Fig. 13 Pair plot showing the LSTM-CVAE encodings (means) of Flight 16 for the latter 6 latent variables. Each encoding is color-coded by the number of violated flight envelopes.



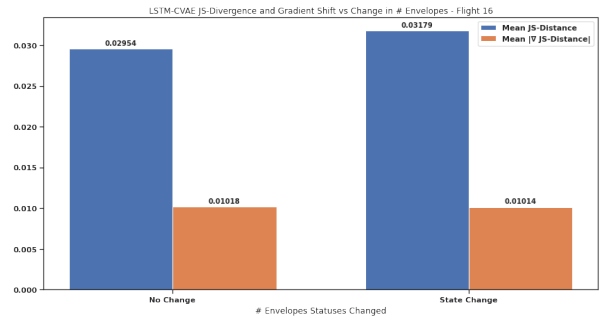
(a) GRU-CVAE Latent Space JS-Distance Gradient and Divergence per # Envelopes Changed.



(b) LSTM-CVAE Latent Space JS-Distance Gradient and Divergence per # Envelopes Changed.



(c) GRU-CVAE Latent Space JS-Distance Gradient and Divergence during State Change.



(d) LSTM-CVAE Latent Space JS-Distance Gradient and Divergence During State Change

Fig. 14 Flight 16 Gaussian Shift for Envelope Changes in the next Second.

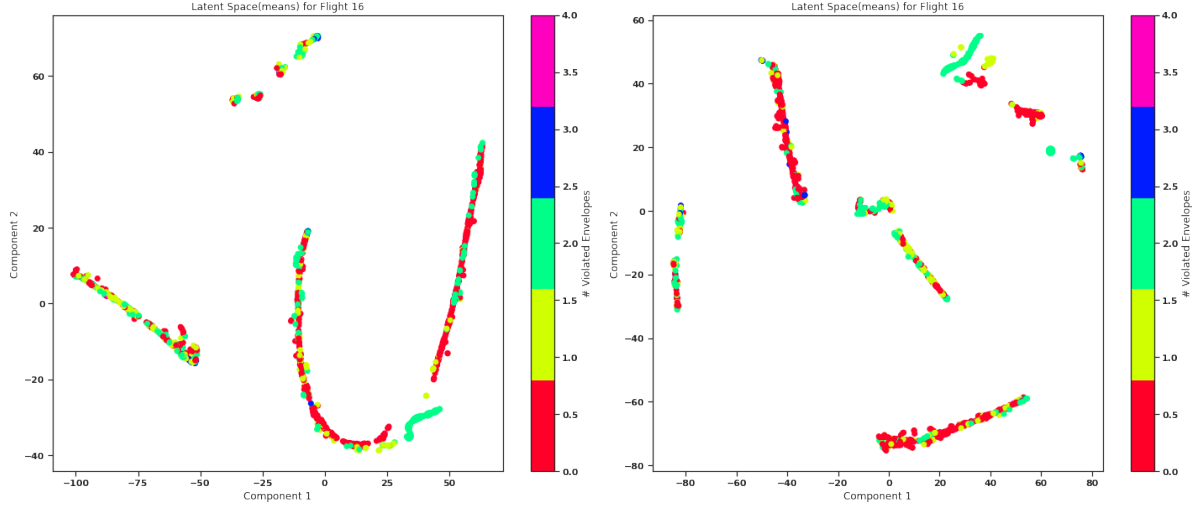


Fig. 15 t-SNE plot of the Flight 16 latent space means for the GRU-CVAE (left) and LSTM-CVAE (right).

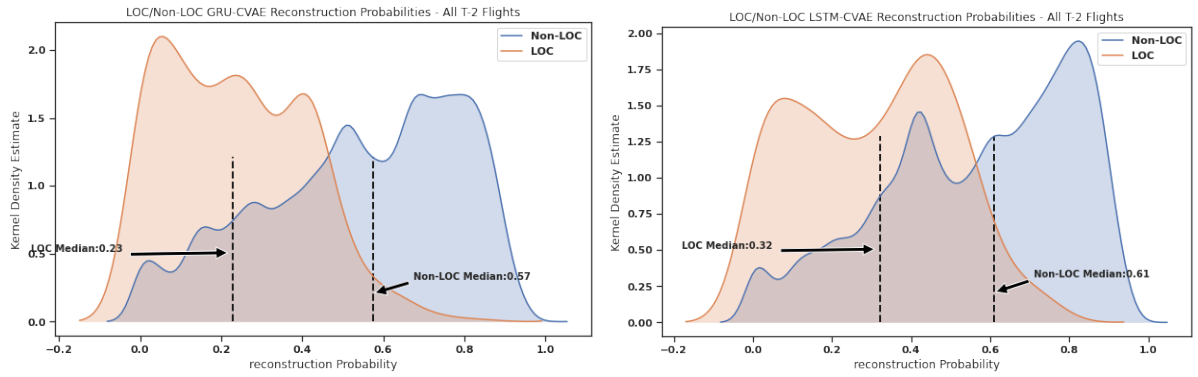
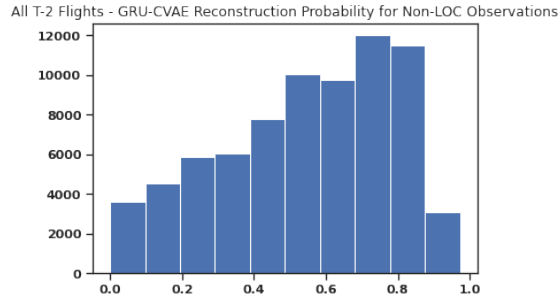
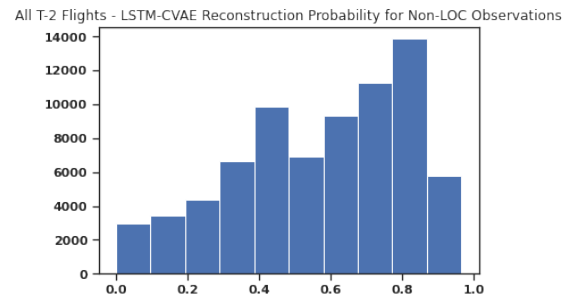


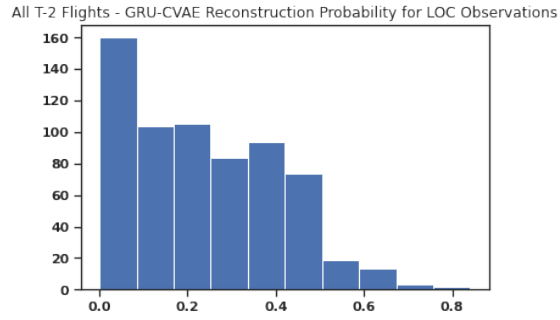
Fig. 16 KDE plots showing the separation of reconstruction probabilities for the GRU-CVAE (left) and LSTM-CVAE (right).



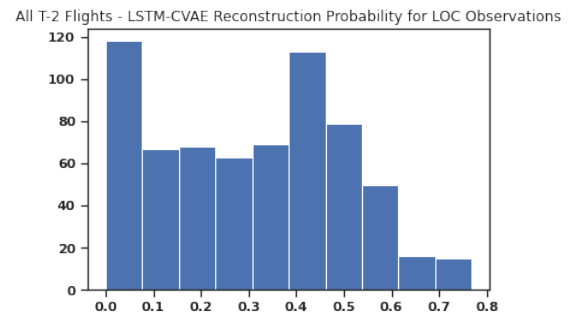
(a) GRU-CVAE Reconstruction Probabilities for Non-LOC Observations.



(b) LSTM-CVAE Reconstruction Probabilities for Non-LOC Observations.

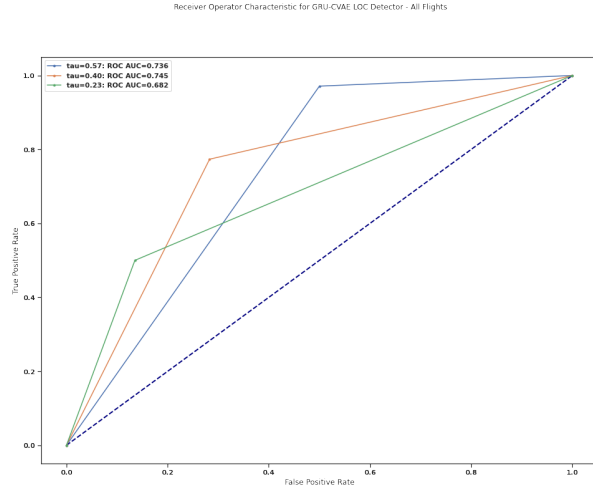


(c) GRU-CVAE Reconstruction Probabilities for LOC Observations.

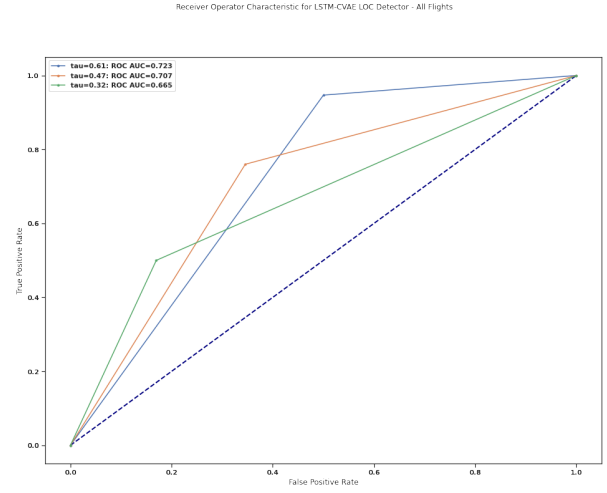


(d) LSTM-CVAE Reconstruction Probabilities for LOC Observations.

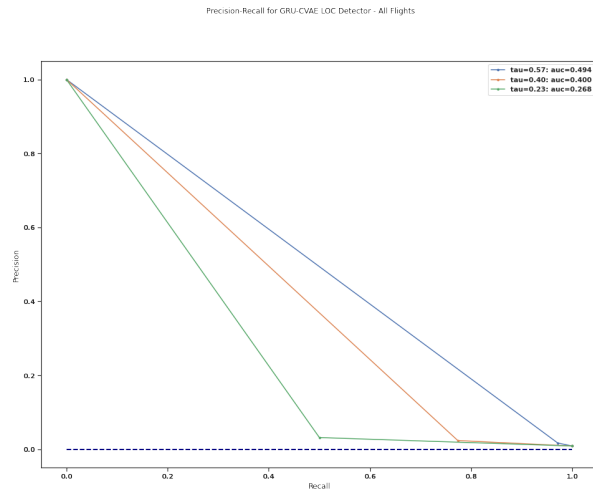
Fig. 17 Histograms of Reconstruction Probabilities for T-2 Flight 16. The LOC reconstruction probabilities for the LSTM-CVAE(right) are more distributed than those of the GRU-CVAE(left).



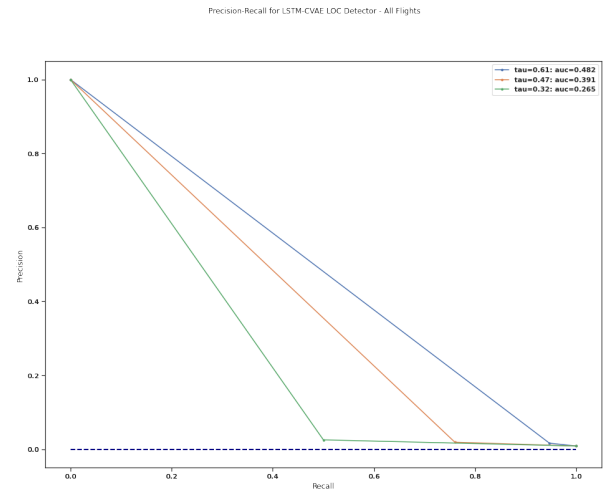
(a) All Flights - ROC Curve for LOC detector based on GRU-CVAE



(b) All Flights - ROC Curve for LOC detector based on LSTM-CVAE

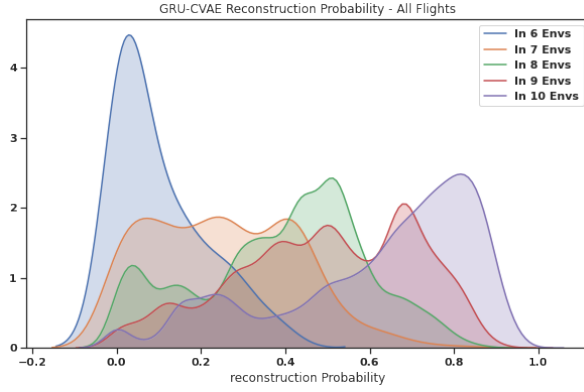


(c) All Flights - Precision-Recall Curve for LOC detector based on GRU-CVAE

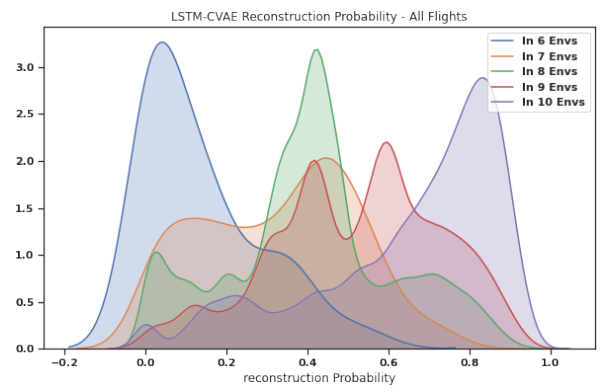


(d) All Flights - Precision-Recall Curve for LOC detector based on LSTM-CVAE

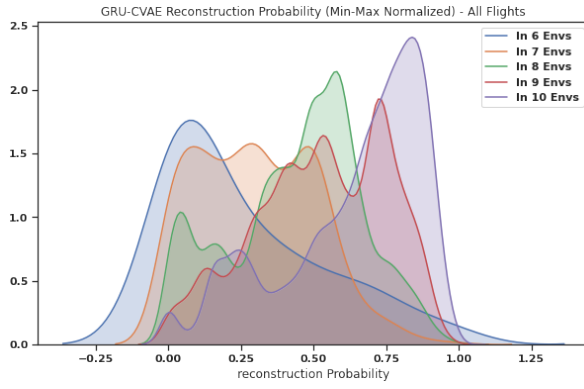
Fig. 18 ROC and Precision-Recall Performance of LOC Detectors based on Reconstruction Probability for All T-2 Flights and Maneuvers.



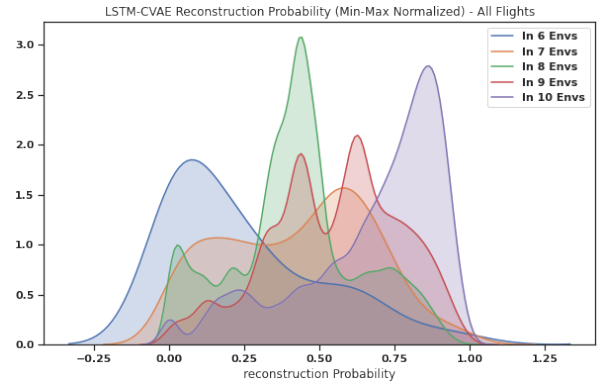
(a) GRU-CVAE Reconstruction Probabilities Densities for number of Envelopes that the aircraft is operating in across all flights.



(b) LSTM-CVAE Reconstruction Probabilities Densities for number of Envelopes that the aircraft is operating in across all flights.

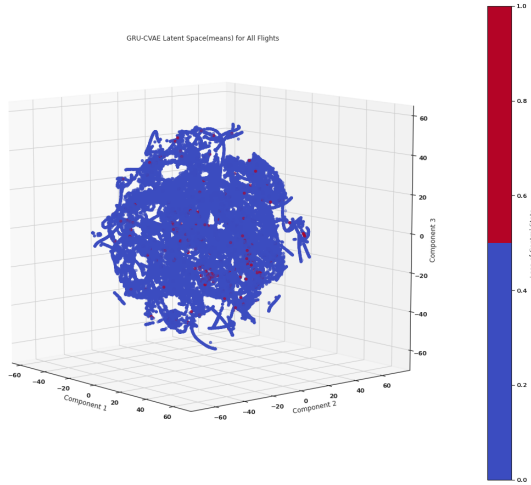


(c) GRU-CVAE Reconstruction Probabilities Densities that have been min-max normalized across all observations for which the aircraft is operating in a number of envelopes.

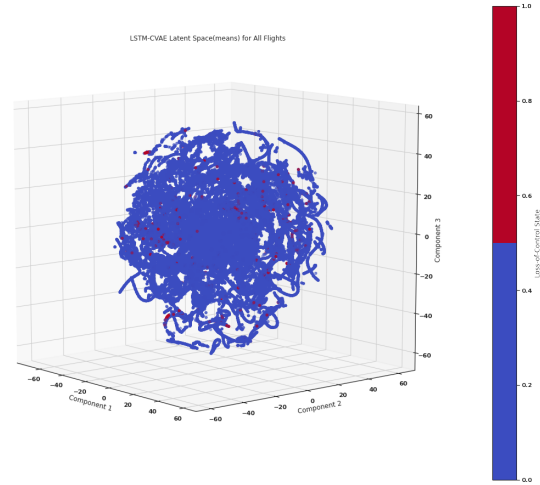


(d) LSTM-CVAE Reconstruction Probabilities Densities that have been min-max normalized across all observations for which the aircraft is operating in a number of envelopes.

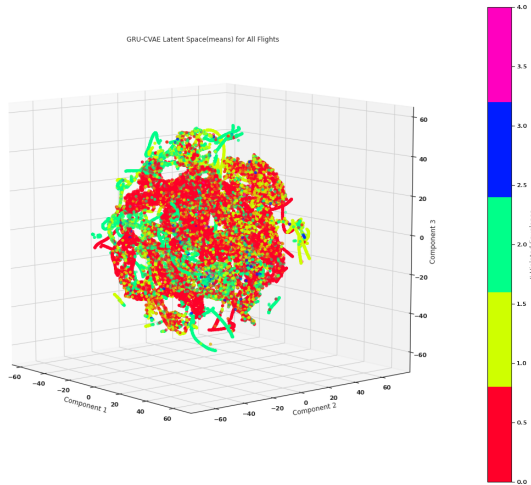
Fig. 19 KDE Plots of CVAE Reconstruction Probabilities categorized by the number of envelopes that the T-2 aircraft is in.



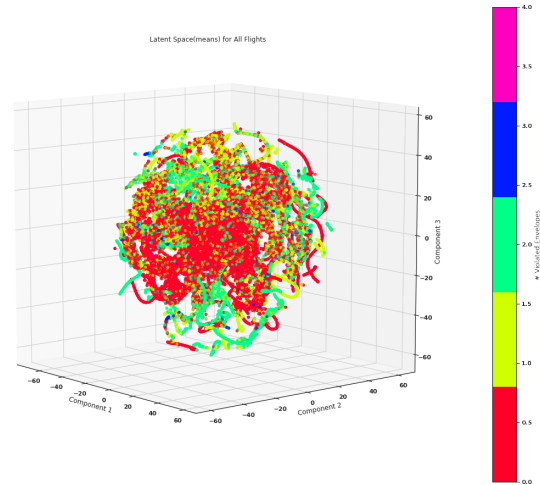
(a) 3D Manifold of GRU-CVAE latent space means for all flights, labeled by LOC events occurring after the encoding.



(b) 3D Manifold of LSTM-CVAE latent space means for all flights, labeled by LOC events occurring after the encoding.

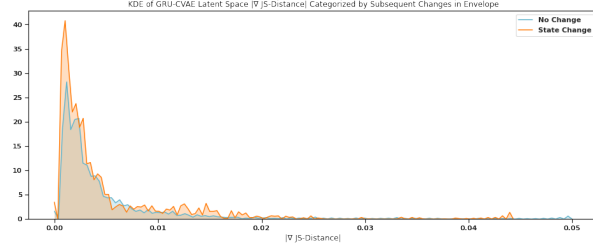


(c) 3D Manifold of GRU-CVAE latent space means for all flights, labeled by the number of envelopes violated by the aircraft for the observation after the encoding.

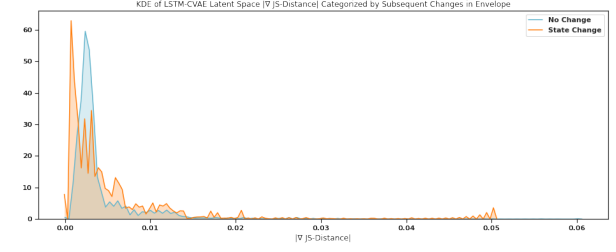


(d) 3D Manifold of LSTM-CVAE latent space means for all flights, labeled by the number of envelopes violated by the aircraft for the observation after the encoding.

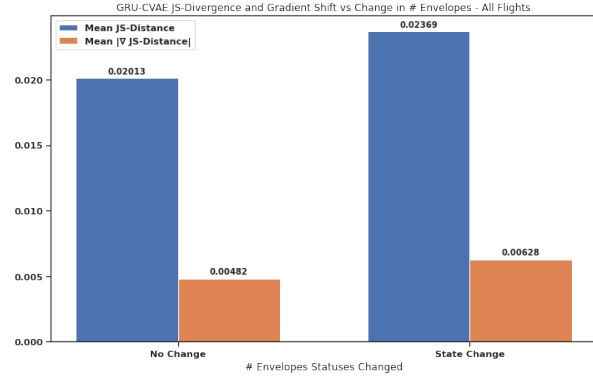
Fig. 20 Reduced 3D Manifold of CVAE latent space means for the encodings of all flights and maneuvers. The encoding labels correspond to the observations that happen after the encoding.



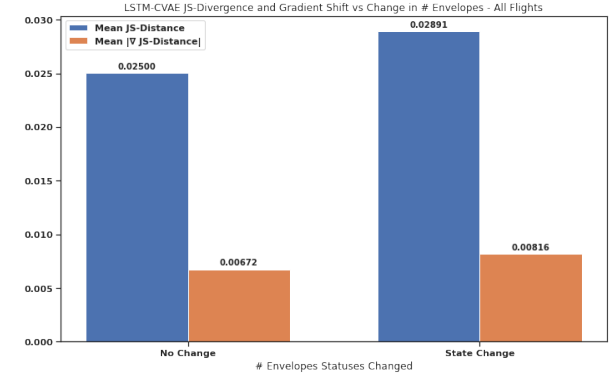
(a) KDE of GRU-CVAE $\hat{D}_{JS}$ over all flights, categorized by state change after the encoding.



(b) KDE of LSTM-CVAE $\hat{D}_{JS}$ over all flights, categorized by state change after the encoding.

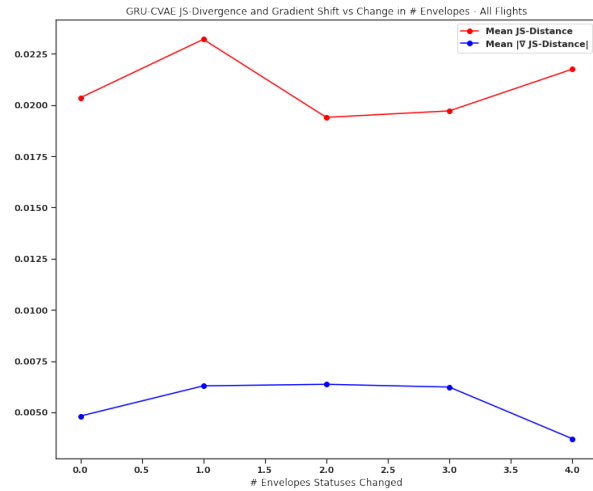


(c) Mean GRU-CVAE $\hat{D}_{JS}$ over all flights, categorized by state change after the encoding.

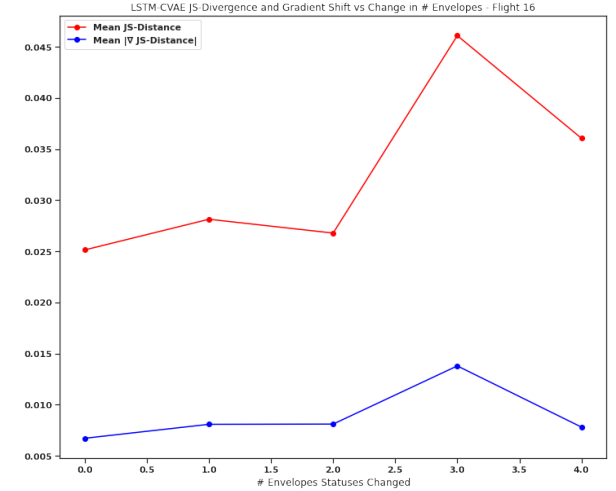


(d) Mean LSTM-CVAE $\hat{D}_{JS}$ over all flights, categorized by state change after the encoding.

Fig. 21 Plots of the gradient of the JS-Distance across all flights, categorized by whether a change in envelope.



(a) GRU-CVAE $\hat{D}_{JS}$ over all flights, categorized by the number of envelopes that change after the encoding.



(b) LSTM-CVAE $\hat{D}_{JS}$ over all flights, categorized by the number of envelopes that change after the encoding.

Fig. 22 Plots of the gradient of the JS-Distance across all flights, categorized by the number of envelopes that change after the encoding.