2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Online Fault Detection in Manipulation Tasks via Generative Models

Michael W. Lanighan and Oscar Youngquist

***Abstract*— This paper introduces a method, Generative Adversarial Networks for Detecting Erroneous Results (GANDER), leveraging Generative Adversarial Networks to provide online error detection in manipulation tasks for autonomous robot systems. GANDER relies on mapping input images of a trained task to a learned manifold that contains only positive task executions and outcomes. When reconstructed through this manifold, the input images from successful task executions will remain largely unchanged, while the images from a failed task will change significantly. Using this insight, GANDER enables inspection and task outcome verification capabilities using a large number of positive examples but only a small set of negative examples, thus increasing the applicability of autonomous robot systems. We detail the design of GANDER and provide results of a proof-of-concept system, establishing its efficacy in an autonomous inspection, maintenance, and repair task. GANDER produces favorable results compared to baseline approaches and is capable of correctly identifying off-nominal behavior with 91.65% accuracy in our test task. Ablation studies were also performed to quantify the amount of data ultimately needed for this approach to succeed.**

I. Introduction

Supervisory control frameworks [1], [2], [3], [4] have shown promise in enabling robot systems to perform a host of tasks that were previously considered unautomatable. However, these approaches require skilled operator interaction, attention, and feedback to perform verification due to the wide variety of unknown failure cases, which limits the scope of their deployment. An effective solution to this problem is to enable autonomous systems to provide self-supervised feedback. To address this shortcoming, we propose the Generative Adversarial Networks for Detecting Erroneous Results (GANDER) system to leverage Generative Adversarial Networks (GANs) to provide online error detection for autonomous robot systems. Inspired by Hirose *et al.* and their work on GONet [5], [6], GANDER relies on mapping input images from a trained task to a learned manifold that contains only positive task executions and outcomes. Images from successful task executions will therefore remain largely unchanged, while images from a failed task will change significantly. GANDER increases the inspection and task outcome verification capabilities and enables fail-active behavior, such that robot systems can detect and react to failures before damaging themselves, manipulated objects, or the environment. We detail the design of GANDER and provide results from a proof-of-concept system, establishing efficacy in an autonomous inspection, maintenance, and repair task.

Authors are with TRACLabs Inc. 1331 Gemini Street; Houston, TX 77058. Corresponding Author: mlanighan@tracblabs.com. This work was supported by NASA contract 80NSSC22PB229.

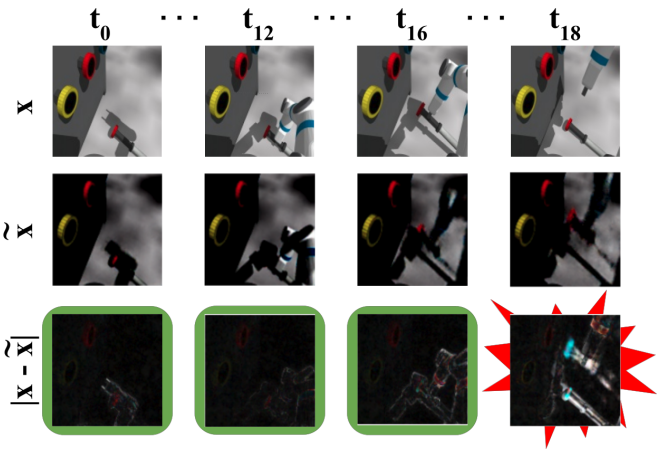


Fig. 1. Rollout of a failed simulated hose mating task from the robot's POV. $\mathbf{x}$: Input images from robot's POV. $\tilde{\mathbf{x}}$: Input images $\mathbf{x}$ reconstructed on the manifold of positive task outcomes. $|\mathbf{x}-\tilde{\mathbf{x}}|$: Difference between input and reconstruction. Small differences indicate nominal behavior ($t_0 \rightarrow t_{17}$) while off-nominal behavior generates large differences (t_{18}), allowing GANDER to detect the failure and abort execution.

II. Related Work

Anomaly Detection (AD) is the process of differentiating between the expected and unexpected behavior of a system. Recently, anomaly detection has seen a surge in work exploring the use of learned generative models [7]. Auto-Encoders (AE), in particular Variational AEs (VAE), are popular choices for generative models. Ji *et al.* propose the Supervised-VAE which uses both "normal" and anomalous data to jointly train a VAE on high-dimensional inputs and a classifier consuming the latent embeddings of the VAE and lower-dimensional inputs to detect anomalies [8]. Park *et al.* introduce the LSTM-VAE, used to learn the temporal dependencies in multi-model data in robotic assisted feeding tasks [9].

Schlegl *et al.* introduced one of the first GAN-based AD methods, AnoGAN, in which a GAN model learns a nominal class manifold using only data from nominal instances [10]. Similar to Park *et al.* the distance between the input and its reconstruction through the learned GAN manifold is used to predict anomalies via thresholding. However, AnoGAN requires an expensive optimization routine to generate images similar to the original. To address this shortcoming, other works have explored training an encoder model jointly with the GAN generator [11], [12]. Hirose *et al.* used GANs in GONet to determine terrain traversability by training a GAN to generate images from positive (traversable) data only [5], [6]. After training the GAN, it is frozen, and an "inverted" generator model is trained with a reconstruction

objective using the frozen GAN to enable image-to-image capabilities. Finally, both the encoder and the GAN model are frozen and used to train a classifier using a small set of positive and negative classes. The authors demonstrate that exploiting the learned manifold of the GAN minimizes the amount of training data required for the classifier, while still generalizing to novel environments. GANDER leverages the same insight to reduce the number of expensive negative samples required for training the classifier.

Although a VAE can be used directly for image-to-image translation, the representation relies on a pixel-level error signal rather than a feature-level signal, often resulting in blurry reconstructions. The VAEGAN network architecture [13] provides a more principled way of achieving similar image-to-image functionality as GONet by combining a VAE and a GAN. In doing so, the learned feature representations from the GAN discriminator can be used in the VAE reconstruction objective. This effectively forces the combined generator/decoder to learn using richer features from the GAN discriminator instead of relying solely on pixel-level errors, leading to a decoder/generator that yields better reconstructions than a traditional VAE. Baur *et al.* introduce the AnoVAEGAN, which uses a VAEGAN architecture consisting of spatial VAEs to detect anomalies in high-resolution MRI brain scans through threasholding [14]. GANDER avoids the pitfalls inherent to thresholding-based approaches by employing a trained, recurrent classifier for detecting anomalous task outcomes using information from across the observed trajectory. Additionally, by replacing the solely GAN-based generative model, we avoid the additional encoder training step required by GONet while still reaping the added benefits of the discriminator during training.

III. Approach

An overview of the proposed system can be seen in Fig. 2. GANDER employs a generative model to map visual input to the positive domain. Features extracted from the original input image, the reconstruction, and the GAN discriminator are then used to train a classifier head to predict the probability of task success.

A. Generative Feature Extractor

In this work, we leverage a VAEGAN network to provide the image-to-image mapping required for the feature extraction component of the system. A VAEGAN model collapses the decoder from a VAE and the generator from a GAN such that the generator/decoder network is trained to produce a reconstruction $\tilde{x}$ of an input x, $\tilde{x} \sim Gen(z) = p(x|z)$, from the latent representation z that is sampled from the encoder network, $z \sim Enc(x) = q(z|x)$, to fool the discriminator network into incorrectly labeling the input reconstructions as the genuine samples from the underlying data distribution.

As originally proposed, a VAEGAN network achieves this by optimizing the loss function $\mathscr{L}$, shown in Equation 1, which trains a VAE and a GAN concurrently. A combination of several loss terms, the first enforces a *prior* $p(z)$ over latent distribution that encourages coverage and locality of the latent space (Eq. 2). Specifically: $z \sim \mathscr{N}(0,I)$. The second is a *reconstruction* loss that encourages accurate reconstructions of the input from the latent space (Eq. 3), and the last the a *GAN* loss, which encourages generating outputs representative of the target domain that fool the discriminator (Eq. 4). Note that the formulas are presented in the non-saturating loss formulation [15]. Here, Dis_l refers to the latent representation of the lth layer of the GAN discriminator.

$$\mathscr{L} = \mathscr{L}_{prior} + \mathscr{L}_{\ell}^{Dis_l} + \mathscr{L}_{GAN} \tag{1}$$

$$\mathscr{L}_{prior} = D_{KL}(q(z|x)||p(z)) \tag{2}$$

$$\mathscr{L}_{\ell}^{Dis_l} = -E_{q(z|x)}[\log p(Dis_l(x)|z)] \tag{3}$$

$$\begin{aligned}\mathscr{L}_{GAN} = \log(Dis(x)) + \log(Dis(Gen(z))) \\ + \log(Dis(Gen(Enc(x))))\end{aligned} \tag{4}$$

However, not all network parameters, θ, are updated with the combined loss and instead each network is updated via the following rules:

$$\theta_{enc} \overset{+}{\leftarrow} -\nabla_{\theta_{enc}}(\mathscr{L}_{prior} + \mathscr{L}_{\ell}^{Dis_l}) \tag{5}$$

$$\theta_{gen} \overset{+}{\leftarrow} -\nabla_{\theta_{gen}}(\gamma\mathscr{L}_{\ell}^{Dis_l} - \mathscr{L}_{GAN}) \tag{6}$$

$$\theta_{dis} \overset{+}{\leftarrow} -\nabla_{\theta_{dis}}\mathscr{L}_{GAN} \tag{7}$$

In practice, we found that several implementation details were critical to achieving stable training. We enumerate them in the remainder of this section.

1) Eliminate Double Dipping in GAN Loss: The original VAEGAN $\mathscr{L}_{GAN}$ loss (Eq. 4) accounts for "synthetic" data twice. In Equation 4, the first synthetic-loss term—$\log(Dis(Gen(z)))$—serves to regularize the latent space of the GAN prior using a sample z drawn from the prior. The second synthetic term—$\log(Dis(Gen(Enc(x))))$—uses the learned encoding as input. However, because the encoder's VAE loss is already regularizing the latent space used by the GAN, this effectively double dips as $Enc(x) \approx z$; likely leading to gradient issues during training. As such, we modified the $\mathscr{L}_{GAN}$ loss to remove the first synthetic term altogether, yielding:

$$\mathscr{L}_{GAN} = \log(Dis(x)) + \log(Dis(Gen(Enc(x)))) \tag{8}$$

which equally weights the real and synthetic data when training the system.

2) Augmented Reconstruction Loss: Other image-to-image networks that combine a GAN-loss with an AE loss, such as [16], include a pixel-level loss in addition to the standard GAN loss for the generator/decoder only. This addition helps guide the gradient in early training, where pixel-level differences will provide more guidance than discriminator features. With this in mind, we added a standard VAE pixel loss to the VAEGAN generator's loss with weighting λ. This transforms the generator update in Equation 6 to

$$\theta_{gen} \overset{+}{\leftarrow} -\nabla_{\theta_{gen}}(\gamma(\mathscr{L}_{\ell}^{Dis_l} - \lambda E_{q(z|x)}[logp(x|z)]) - \mathscr{L}_{GAN}) \tag{9}$$

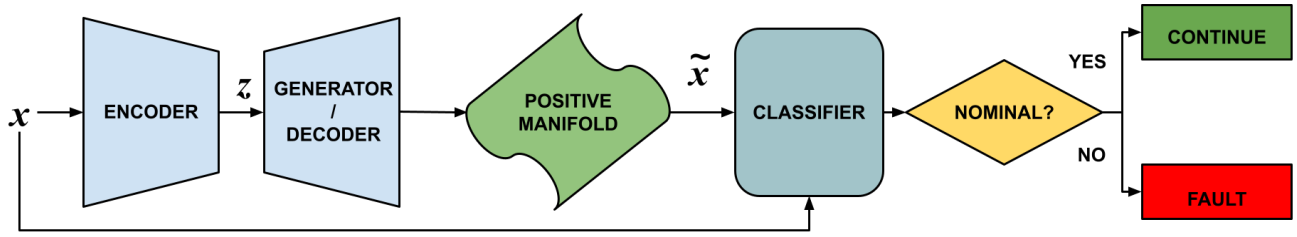


Fig. 2. High-level system diagram for GANDER. Runtime images $\boldsymbol{x}$ are fed into an encoder to map to a latent representation $\boldsymbol{z}$. This latent representation is then mapped to the positive manifold through a reconstruction yielding $\tilde{\boldsymbol{x}}$. The original input and reconstruction are then fed into a classifier. If the input $\boldsymbol{x}$ and reconstruction $\tilde{\boldsymbol{x}}$ diverge significantly, the input did not originally belong to the manifold, indicating that the input was capturing off-nominal behavior.

3) Cyclical KL Weighting: In addition to the above VAE-GAN loss and update rules, we introduced a cyclic weighting of $\mathscr{L}_{prior}$ in Eq. 5 in order to emphasize coverage and locality of the latent space or input reconstruction [17]. Cycling this weighting helps avoid local minima during training and results in the new update rule:

$$\theta_{enc} \overset{+}{\leftarrow} -\nabla_{\theta_{enc}}(\lambda_{prior}\mathscr{L}_{prior} + \mathscr{L}_{\ell}^{Dis_l}) \tag{10}$$

4) Hyperparameter Search: We performed random hyper-parameter searches to identify a promising parameterization for training the network. This search identified the relative learning rates of the VAE and GAN discriminator and the $\mathscr{L}_{prior}$ cycle length to have the greatest impact on performance.

B. Classifier Models

GANDER deals with streaming time-series data, so a recurrent approach, like an LSTM, is ideally suited to detect whether an input trajectory is evolving toward off-nominal behavior. However, to generate initial results for our proof-of-concept examples, we leveraged a standard fully connected (FC) classifier to determine if input images were nominal or not. This "snapshot" classifier (GANDER FC) then provided a baseline performance that was contrasted with the LSTM (GANDER LSTM) performance. Both approaches were fed features that were extracted from the input and reconstruction images. Specifically, the residual difference between the input and its reconstruction $|x-\tilde{x}|$, the difference between discriminator features $|Dis_l(x)-Dis_l(\tilde{x})|$, and lastly the discriminator features of the original images $Dis_l(x)$. We refer the reader to [5] for more details.

IV. Evaluation Settings

To train and validate the proposed GANDER system, datasets were developed in a robot manipulation task simulating a lunar maintenance task of mating a power/data/fluid line to a habitat (Fig. 3). The task was encoded as an Affordance Template (AT) [1] and performed in a custom Gazebo simulation environment of a lunar habitat with a simulated Zebra Fetch robot. RGB Images were collected from the robot's head-mounted sensor at 3 Hz and resized to 128x128 to reduce the dataset size and ease training.

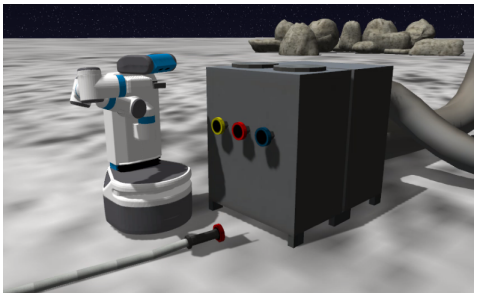


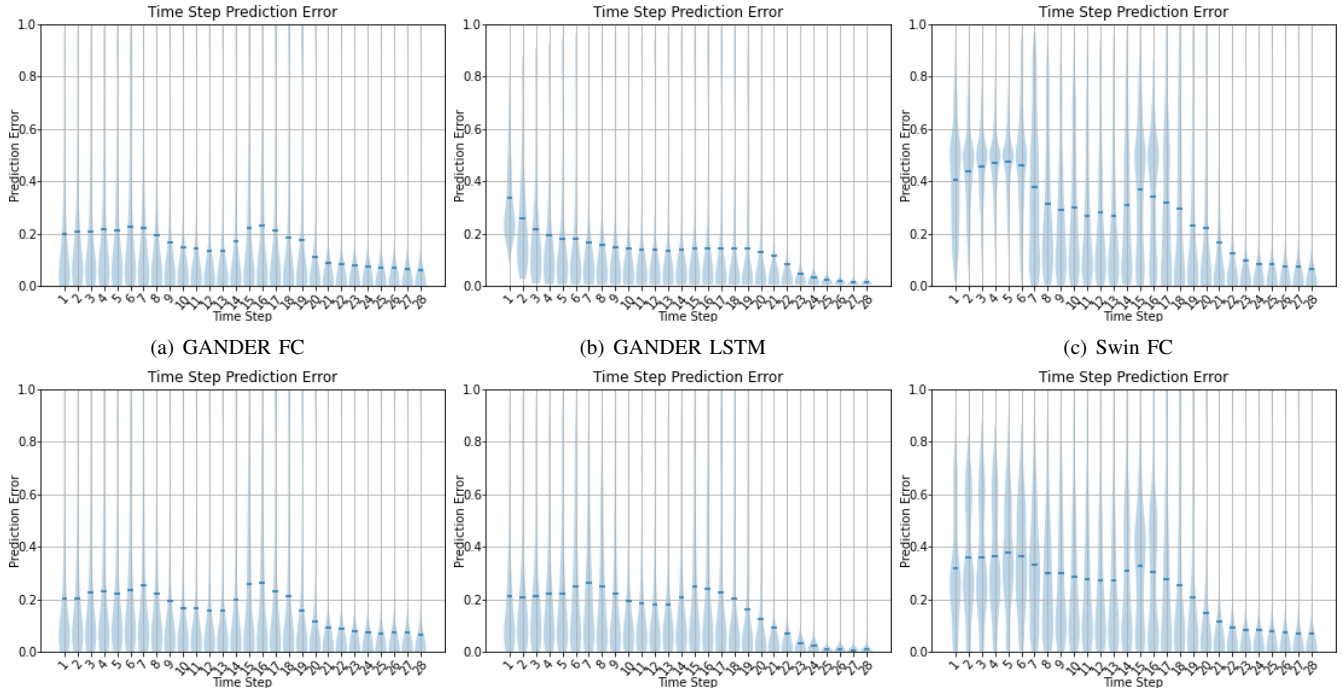
Fig. 3. Experimental setup: A lunar maintenance task in which a robot mates a power/data/fluid line to a habitat module.

A. Data Collection

Data was collected via finite state machines that managed simulation state. Nominal task-execution data were collected from 4828 runs, yielding 140713 images. For the VAEGAN training data, we randomly sampled 140000 nominal images and created training, validation, and testing sets using an 80/10/10 split. In this study, our aim was to capture manipulation failures, such as slips and drops. To force such outcomes, the contact friction properties of the simulated objects were disabled when collecting off-nominal examples. 2351 off-nominal trajectories (63938 images) were collected with these modified properties. A subset of 1200 off-nominal (34236 images) and 1200 nominal (35553 images) trajectories were used to create annotated data for classifier training. Each time step in a trajectory was labeled with the trajectory-level outcome. These 2400 trajectories were split into training, validation, and test sets containing 2000 (58145 images), 200 (5808 images), and 200 (5836 images) trajectories, respectively. For sequential training, all trajectories were truncated or padded to a sequence length of 28 images.

B. Baseline Models

We compared the performance of the two variants of the GANDER system described above (FC and LSTM classifier versions) with four baseline approaches: (1) A fully connected classifier trained directly on the annotated task data (Image FC), (2) A fully connected classifier trained on the extracted features of the images in the annotated task



(a) GANDER FC (b) GANDER LSTM (c) Swin FC

(d) VAE FC (e) VAE LSTM (f) Image FC

Fig. 4. Violin plots of prediction error over all test trajectories for the (a) GANDER FC classifier (b) GANDER LSTM classifier (c) baseline Swin-features FC classifier (d) baseline FC classifier trained directly on the annotated image set, (e) baseline VAE front-end FC classifier, and (f) baseline VAE front-end LSTM classifier trained. Each "violin" visualizes the distribution over prediction error at each time step. The mean of each distribution is indicated.

data[1] (Swin FC), (3) A VAE based variant of GANDER where a VAE maps images to the positive manifold in place of the VAEGAN fed to a fully connected classifier (VAE FC), and (4) A VAE based variant of the LSTM GANDER implementation (VAE LSTM). Each approach was evaluated in the lunar maintenance domain and was trained five times on the full classifier (annotated) dataset to accumulate performance statistics.

V. Experimental Results

A. Classification

Table I displays the mean classification accuracy, mean prediction error, and total AUC across five separate training runs and Fig. 4 shows the distribution of prediction error at each time step. In pure classification, the GANDER variants outperform the baseline systems. Both GANDER approaches exhibit more stable performance throughout the trajectory, especially compared to the two nongenerative FC baselines. However, the VAE front-end classifiers perform very similarly to the GANDER approaches. When comparing the GANDER LSTM and VAE LSTM approaches, GANDER has higher prediction errors in initial time steps, but over the course of the trajectory trends towards zero faster and with greater consistency than the VAE. The distribution of errors at each time step suggests that each system suffers from overconfidence in incorrect predictions, but the distributions of GANDER are noticeably more concentrated near zero and have lower means, suggesting a lower impact from outliers than the VAE. This suggests that the additional features that the VAEGAN architecture learns from the GAN-based losses better support the LSTM classifier subsequently learning the spatial-temporal dependencies of the task.

[1]Features were extracted from a frozen, pre-trained state-of-the-art classifier Swin transformer network [18]

B. Fail-active Behavior

We next investigated the potential fail-active behavior enabled by GANDER. To evaluate the system's potential, we iterated through all trajectories in the test set, split evenly between nominal trajectories that should not be aborted and off-nominal trajectories that should be aborted. For each trajectory we then note at which point the $P(success) < \alpha$, where α is a specified "abort" threshold. When this occurs, the robot stops executing the remaining trajectory. If a trajectory did not fall below the threshold, it is binned under *Miss*. For nominal trajectories, a hit corresponds to a false negative (FN)—an abort should not have been triggered but was—while a miss is a true positive (TP). For off-nominal trajectories, a *Miss* indicates a false positive (FP)—an abort should have been triggered but was not—and a hit represents a true negative (TN). A perfect classifier would have no FN for nominal trajectories, all labeled *Miss*, while having no FP for off-nominal trajectories, none labeled *Miss*.

The ability of each approach to trigger aborts with $\alpha = 0.05$ in test trajectories is summarized in Table II. Although all approaches were quite sensitive to off-nominal trajectories, most exhibited fair nominal trajectory performance. Of note is the GANDER LSTM variant, which significantly

TABLE I

COMPARISON OF MEAN IMAGE CLASSIFICATION ACCURACY, MEAN PREDICTION ERROR, AND AUC OF GANDER AND BASELINES

Approach	Accuracy (%)		Prediction Error		AUC
	μ	σ	μ	σ	
GANDER FC	88.84	0.265	0.1533	0.2685	0.94
GANDER LSTM	**91.65**	0.16	**0.1328**	0.2221	**0.96**
Image FC	86.61	2.02	0.2381	0.2628	0.91
Swin FC	81.89	0.91	0.2729	0.2661	0.88
VAE FC	88.07	0.286	0.1671	0.2664	0.93
VAE LSTM	89.80	0.331	0.1591	0.2244	**0.96**

TABLE II

SUMMARY ABORT MEASURES FOR NOMINAL AND OFF-NOMINAL TRAJECTORIES FOR GANDER AND BASELINES

Approach	Nominal		Off-nominal	
	μ	σ	μ	σ
Perfect	100.0	0.0	0.0	0.0
GANDER FC	83.20	0.40	0.0	0.0
GANDER LSTM	98.0	1.09	0.8	0.75
Image FC	85.0	0.0	0.0	0.0
Swin FC	85.0	0.0	0.0	0.0
VAE FC	82.6	0.8	0.0	0.0
VAE LSTM	92.2	1.94	0.0	0.0

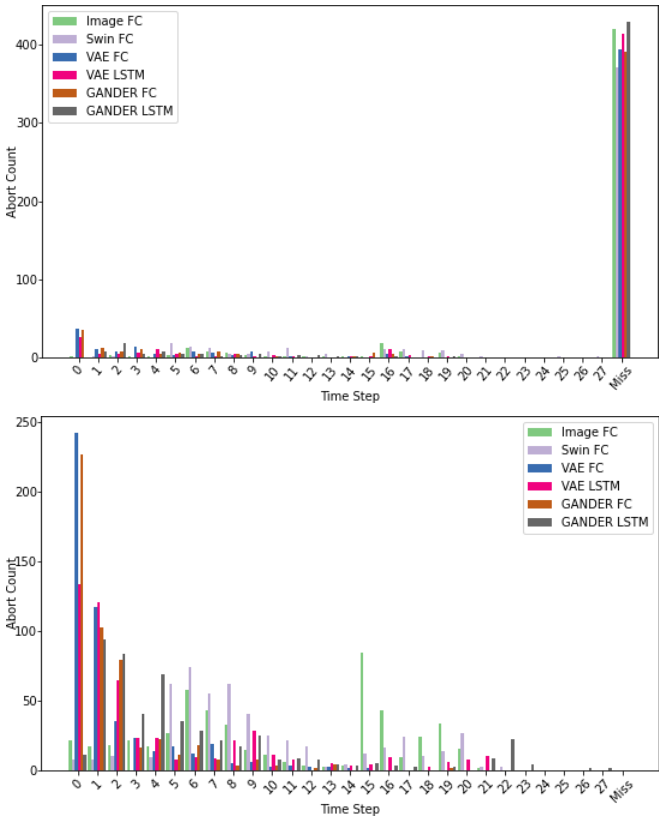


Fig. 5. "Abort" behavior over nominal (top) and off-nominal (bottom) trajectories across five random seeds. To ease analysis, trajectories were processed up to $min(len(trajectory), 28)$. A hit at a time step indicates $P(success) < 0.05$. When this occurs, an abort would be triggered and the robot would stop execution.

outperformed other approaches in nominal trajectories while missing a very small number of off-nominal trajectories. This reduced performance could be due to the inherent stochasticity of the trained models.

The direct image and Swin feature baselines performed poorly compared to the generative models (GANDER FC and VAE FC). Plots summarizing the abort behavior for the baselines and GANDER are shown in Fig. 5. The generative FC models (VAE FC and GANDER FC) labeled TN earlier/more aggressively in test trajectories at the expense of an increase in FN count. The recurrent generative approaches (VAE LSTM and GANDER LSTM) were more hesitant to label TN, resulting in lower FN outcomes. The trade-off in TN/FN performance across these approaches would need to be considered before deploying GANDER.

C. Data Efficiency: VAEGAN Models

To understand the sensitivity of GANDER to the relative training dataset sizes we performed a series of ablation studies, reducing the sizes of the training sets for both the VAEGAN and classifier components. This set of studies ablated the training sets while leveraging the original full validation and test sets. Each model was trained with the same set of hyperparameters and evaluated on the same test set (randomly sampled 10% of the entire dataset).

The VAEGAN training dataset was ablated 3 times, creating datasets of 75% (84000 images), 50% (56000 images), and 25% (28000 images) of the original nominal trajectory dataset. Hyperparameters were held constant for each model, each of which was trained for 100 epochs. Table III shows representative outputs of the ablated models to capture reconstruction quality and the ability of the network to map off-nominal inputs to the positive manifold. As expected, ablating the VAEGAN training dataset diminished its ability to reconstruct images on the positive manifold. However, even when the VAEGAN training set is considerably ablated, key aspects of the task appear to be mapped, for both nominal and off-nominal inputs.

D. Data Efficiency: Classifier Models

In order to quantify how much annotated data is necessary for our approach, we performed an ablation study with respect to the amount of labeled training data for the classifiers. For these studies, we froze the VAEGAN component of GANDER, using a model trained on the full positive manifold dataset, and retrained the classifier five times. The same test set of 200 trajectories (split evenly between nominal and off-nominal) was used to evaluate all ablations. Each consecutive training set ablated the training set by 50%—yielding annotated training dataset sizes of 1000 trajectories/29033 images (50%), 500 trajectories/14535 images (25%), and 250 trajectories/7249 images (12.5%).

Summary performance measures (accuracy, prediction error, and AUC) of the trained models are enumerated in Table IV. As expected, with reduced training data, the performance of the classifiers diminishes. However, their performance indicates that reduced classifier dataset sizes may be possible

TABLE III

REPRESENTATIVE RECONSTRUCTION QUALITY FOR NOMINAL (TOP) AND OFF-NOMINAL (BOTTOM) INPUTS SUBJECT TO ABLATIONS

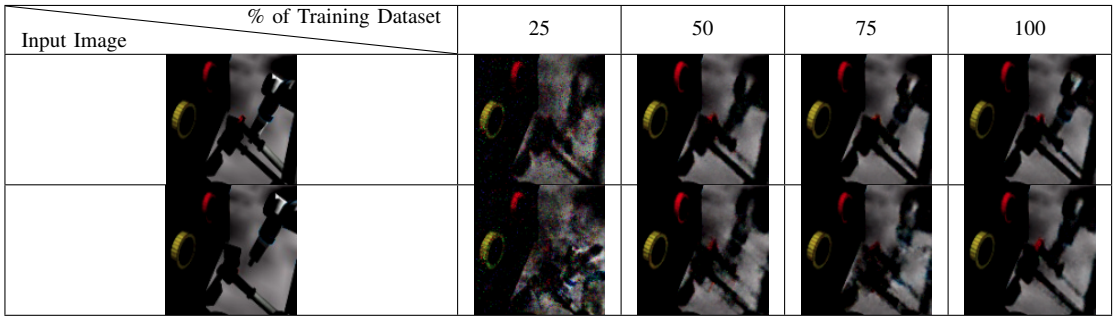


TABLE IV

CLASSIFIER PERFORMANCE SUBJECT TO TRAINING SET ABLATIONS

Approach	Accuracy (%)		Prediction Error		AUC
	μ	σ	μ	σ	
FC / 100%	88.84	0.27	0.1533	0.2685	0.94
LSTM / 100%	91.65	0.16	0.1328	0.2221	0.96
FC / 50%	87.47	0.32	0.1695	0.2714	0.93
LSTM / 50%	89.70	0.33	0.1583	0.2339	0.95
FC / 25%	84.52	0.47	0.2053	0.2760	0.92
LSTM / 25%	88.95	0.25	0.1820	0.2246	0.95
FC / 12.5%	82.41	0.05	0.2294	0.2798	0.90
LSTM / 12.5%	84.63	0.69	0.2476	0.2314	0.92

TABLE V

ABORT MEASURES FOR NOMINAL AND OFF-NOMINAL TRAJECTORIES FOR GANDER MODELS TRAINED ON ABLATED ANNOTATED DATASETS

Approach	Nominal		Off-nominal	
	μ	σ	μ	σ
Perfect	100.0	0.0	0.0	0.0
FC 12.5%	83.4	0.40	0.0	0.0
FC 25%	83.4	0.80	0.0	0.0
FC 50%	82.8	0.40	0.0	0.0
FC 100%	83.2	0.40	0.0	0.0
LSTM 12.5%	1.0	0.0	67.0	24.56
LSTM 25%	99.0	0.89	3.0	0.89
LSTM 50%	99.8	0.4	7.6	5.08
LSTM 100%	98.2	1.09	0.8	0.75

without a severe reduction in performance. The ability of the ablated models to abort when $P(success) < 0.05$ is tabulated in Table V. The FC classifier tended to be more aggressive, detecting all TN in all cases at the expense of increased FN. The LSTM, meanwhile, traded a reduction in FN for a slight decrease in TN detection capability. Recurrent approaches typically require large amounts of training data. As such, the failure of the LSTM (in terms of abort capability) at the 12.5% threshold is not surprising. This result indicates that there is a lower limit for training dataset size for the LSTM for a specific threshold α, where $P(success) < \alpha$ is the threshold to stop the robot. As collecting the annotated datasets is likely the most expensive part of data collection, the overall performance of the GANDER system as the annotated training dataset sizes are reduced is encouraging.

VI. CONCLUSIONS

The proposed system performs fault detection as intended, achieving high performance even when small annotated datasets are used to train the classifier component. In our demonstration task, the GANDER system was able to correctly identify off-nominal behavior with 91.65% accuracy and outperformed all baseline approaches' ability to facilitate *fail-active* behavior by detecting failed trajectories at runtime and aborting subsequent motions. Through the use of the VAEGAN architecture to naturally handle image-to-image mappings, this performance is achieved only using two training steps compared to GONet's three. Furthermore, the improved performance over the VAE baseline indicates that the GAN discriminator incorporated into the VAEGAN training enables the model to learn features that better support learning the downstream AD task. One limitation of this study is that it was performed in simulation. Mapping simulation results to real platforms is a long-standing problem in robotics. Although the presented system was demonstrated in simulation, the low data requirements identified in this study do not preclude collecting training data from deployed hardware. As such, future work will investigate the efficacy of the approach using physical systems.

However, autonomous data collection in simulation would greatly increase the applicability of GANDER. Recent advances have used GANs to transform images obtained from simulation to photorealistic images to train deep-reinforcement learning algorithms [19], [20]. Such an approach would ease training requirements for GANDER by using significantly easier to obtain simulated data and then transforming them to mimic data from real robot operations. Exploring the application of these approaches to GANDER is a promising avenue for future work.

ACKNOWLEDGEMENT

The authors would like to thank Jim Ecker for feedback on system design as well as Khoshrav Doctor, Emily Pruc, and anonymous reviewers for feedback on the manuscript.

REFERENCES

[1] S. Hart, A. H. Quispe, M. W. Lanighan, and S. Gee, "Generalized affordance templates for mobile manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6240–6246.

[2] P. Beeson, S. Hart, and S. Gee, "Cartesian motion planning & task programming with CRAFTSMAN," in *Robotics: Science and Systems Workshop on Task and Motion Planning*, 2016.

[3] A. Pettinger, P. F. Cassidy Elliott, and M. Pryor, "Reducing the teleoperator's cognitive burden for complex contact tasks using affordance primitives," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 513–11 518.

[4] W. Pryor, B. P. Vagvolgyi, A. Deguet, S. Leonard, L. L. Whitcomb, and P. Kazanzides, "Interactive planning and supervised execution for high-risk, high-latency teleoperation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1857–1864.

[5] N. Hirose, A. Sadeghian, M. Vázquez, P. Goebel, and S. Savarese, "GONet: A semi-supervised deep learning approach for traversability estimation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3044–3051.

[6] N. Hirose, A. Sadeghian, F. Xia, R. Martín-Martín, and S. Savarese, "VUNet: Dynamic scene view synthesis for traversability estimation using an RGB camera," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2062–2069, 2019.

[7] M. Sabuhi, M. Zhou, C.-P. Bezemer, and P. Musilek, "Applications of generative adversarial networks in anomaly detection: a systematic literature review," *Ieee Access*, vol. 9, pp. 161 003–161 029, 2021.

[8] T. Ji, S. T. Vuppala, G. Chowdhary, and K. Driggs-Campbell, "Multi-modal anomaly detection for unstructured and uncertain environments," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, 2021, pp. 1443–1455.

[9] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[10] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[11] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 622–637.

[12] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International conference on data mining (ICDM)*. IEEE, 2018, pp. 727–736.

[13] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[14] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain mr images," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*. Springer, 2019, pp. 161–169.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[17] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical annealing schedule: A simple approach to mitigating kl vanishing," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 240–250.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[19] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3722–3731.

[20] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "RL-CycleGAN: Reinforcement learning aware simulation-to-real," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 157–11 166.