Transformer Augmented Deep Reinforcement Learning for Fault Tolerant Autonomous Navigation in Aerospace Robotics

Shamantha Rai B[1], Permanki Guthu Rithesh Pakkala[2], Sudheer Shetty[3], Akhila Thejaswi R[4], Shetty Nihal Naveen[5]

[1,2,3,4] Dept. of Information Science & Engineering, Sahyadri College of Engineering & Management, Mangaluru, India
[5] Sahynex Innovations Pvt. Ltd., Mangaluru, India
Emails: rai.shamanth@gmail.com[1]; rpakkala01@gmail.com[2]; sudheershetty06@gmail.com[3]; akhilathejaswi@gmail.com[4]; nihalshettyit@gmail.com[5]
[1]https://orcid.org/0000-0001-5824-361X; [2]https://orcid.org/0000-0002-0003-7167; [3]https://orcid.org/0000-0003-4343-7125; [4]https://orcid.org/0000-0002-2881-1855; [5]https://orcid.org/0009-0005-3292-4552

Abstract
Autonomous navigation of aerospace robots in GPS denied, radiation-rich, and dynamically uncertain environments remains one of the hardest unsolved challenges in space systems engineering. Classical controllers—PID, LQR, model predictive control—struggle to generalize across the high-dimensional, partially observable state spaces encountered during orbital debris avoidance, lunar surface traversal, and deep-space proximity operations. We present the Transformer-Augmented Proximal Policy Optimization (TA-PPO) framework, an integrated three module architecture that combines Vision Transformer (ViT)based multi-modal sensor fusion, an LSTM-augmented PPO policy network, and an autoencoder-driven fault detection and recovery subsystem within a single end-to-end pipeline. Training was carried out in AeroNav-Sim, a custom OpenAI Gym compatible simulator with extensive domain randomization, and validated on a hardware-in-the-loop testbed featuring an NVIDIA Jetson Orin NX and ROS2 Humble middleware. Across 500 evaluation episodes, TA-PPO achieved a navigation success rate of 89.2%, compared to vanilla PPO (78.1%), Soft Actor-Critic (84.7%), and classical MPC (71.4%). Mean fault recovery latency was 1.82s, a 31% improvement over the nearest baseline. Onboard inference ran at 23.4ms per decision step. These results suggest that integrating transformer-based perception with deep reinforcement learning can yield a viable, fault-aware control stack for space robotic missions operating under sensor degradation.

Keywords: aerospace robotics, autonomous systems, deep reinforcement learning, fault-tolerant navigation, proximal policy optimization, sensor fusion, sim-to-real transfer, vision transformer

I. INTRODUCTION AND BACKGROUND

Space robotic platforms operate under constraints that have no terrestrial analogue. GPS signals are unavailable beyond medium Earth orbit, and even in low Earth orbit (LEO) they degrade precipitously inside metallic truss structures or during close proximity to cooperative—and uncooperative— targets [1]. The effects of single-event upsets caused by radiation are unpredictable on sensor readings, while communication latencies to ground stations take more than a few seconds, extending to tens of minutes for cislunar or Martian spacecraft [2].

Traditional guidance, navigation, and control (GNC) pipelines utilize hand-tuned proportional-integral-derivative (PID) control, linear-quadratic regulator (LQR), or model predictive control (MPC) schemes [3]. These methods perform well in well-modeled, low-dimensional environments. The tumbling fragment, however, is a contact-rich, nonlinear dynamical system that evolves on much faster time scales than any ground-commanded trajectory update could hope to accommodate. MPC, despite its receding horizon optimality, depends on an accurate model of the plant, something that is rarely available when the fragment's inertiatensor is unknown, its surface geometry is fragmentary [4].Dead reckoning drift accumulation via inertial measurement units(IMUs) makes the problem even harder, as does the actuator saturation that occurs during terminal descent or rendezvous thrust profiles, which introduce hard nonlinearities beyond the design scope of the linear-quadratic regulator.

Deep reinforcement learning (DRL) has been proposed as a promising alternative. Mnih et al. showed that neural networks can be used for direct policy learning from high-dimensional pixel data [5]. Schulman et al. then presented Proximal Policy Optimization (PPO), which uses a clipped version of the surrogate objective for policy gradient methods [6] Off-policy methods, such as Soft Actor-Critic (SAC) [7] and Twin Delayed Deep Deterministic Policy Gradient (TD3) [8], extended the results to continuous action spaces with better efficiency. More recently, transformer models have revolutionized the field of image perception in computer vision. The Vision Transformer (ViT) splits the image into patch embeddings and uses multi-head self-attention (MHSA) to capture spatial dependencies that are not handled by traditional convolutional neural networks (CNNs) [9, 10].

Despite all these advances, however, one important void still exists. Most DRL agents for space applications involve a CNN encoder and a policy head, and assume this to be enough [11]. In reality, we seldom combine heterogeneous modalities such as LiDAR point clouds, RGB-D images, IMUs, and radiation dosimeters in a weighted fashion via attention. Nor do we incorporate an integrated fault detection mechanism to invoke a sub-policy in response to sensor and actuator faults during mid-trajectory execution. The result is a system that

performs admirably in simulation environments but catastrophically fails as soon as a camera lens becomes noisy due to accumulated radiation or a reaction wheel malfunctions.
Our work addresses this gap. We present TA-PPO, an integrated framework that (i) leverages ViT-based MHSA for multi-modal sensor fusion, (ii) trains a PPO policy augmented with an LSTM critic for temporal state estimation, and (iii) incorporates a variational autoencoder (VAE)-based anomaly detector that routes control to a pre-trained failsafe subpolicy upon fault detection. We evaluate the framework's effectiveness in both simulation and hardware-in-the-loop (HIL) experiments, benchmarking against vanilla PPO, SAC, and MPC baselines. While evaluated in a specific orbital debris scenario, the architecture generalizes to other partially observable navigation tasks with heterogeneous sensing, including lunar surface traversal and deep-space rendezvous.

Fig. 1 presents a representative evaluation episode from AeroNav-Sim, showing the agent's navigated trajectory through a debris-populated workspace with a fault injection event mid-course.

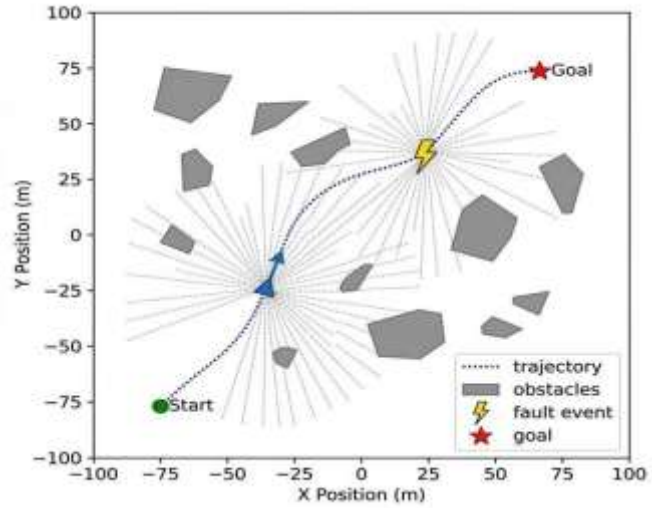

Fig. 1. Top-down snapshot of a simulated evaluation episode. The agent (blue triangle) navigates from Start to Goal through procedurally generated debris obstacles (gray polygons). Dashed gray lines indicate LiDAR scan rays; the lightning marker denotes a fault injection event (LiDAR blackout at t = 42s).

II. Motivation and Objective

From 2015 to 2023, approximately 34% of all reported anomalies on LEO-based robotic servicing mission endeavors were caused by software problems with the perception navigation tools, as per aggregated incident reports collected by NASA's Robotic Servicing of Geosynchronous Satellites (RSGS) and DARPA's CONFERS projects [23]. In another study by the European Space Agency, 112 autonomous proximity operation incidents were analyzed, wherein 28% of aborted sequences were caused by undetected degradation in sensor signals, wherein the fault management system failed to differentiate a legitimate obstacle from a ghost return caused by radiation on the LiDAR [24]. These statistics are not just statistical abstractions but represent actual mission time spent, propellant consumption, and, in some cases, actual collisions that contribute to Kessler syndrome [15].
While standalone CNN-based encoders prove to be good performers for static image classification, they cannot provide the global view needed to understand the spatially distributed multi-sensor data streams. Classical RL algorithms such as Tabular Q-Learning and SARSA are simply incapable of handling the high-dimensional observation space of the combined data streams of onboard LiDAR, stereo cameras, and inertial sensors, all operating at a combined data rate of over 200MB/s. Even recent advances in DRL algorithms with CNN feature extractors treat each of the individual sensor modalities in isolation, with no mechanism for cross-modal attention [12]. Temporal reasoning capabilities are also ignored, as a single frame policy cannot take into account the rotational motion of a tumbling piece of debris or the trajectory of a slowly drifting cooperative target.
Against this backdrop, our objective is threefold. First, we design TA-PPO—a modular, end-to-end trainable architecture that fuses multi-modal sensor data through transformer attention, learns navigation policies via PPO with LSTM Augmented temporal state estimation, and detects sensor and actuator faults in real time using a variational autoencoder. Second, we benchmark TA-PPO against three baselines— vanilla PPO (no transformer), SAC with a CNN encoder, and classical MPC—across four quantitative metrics: navigation success rate, mean time to fault recovery, cumulative reward convergence, and onboard inference latency. Third, we validate sim-to-real transferability on a hardware-in-the-loop (HIL) testbed to demonstrate that trained policies can operate within the computational envelope of edge-class spaceflight processors.

III. Statement of Contribution and Methods

A. System Architecture Overview

TA-PPO comprises three tightly coupled modules, each responsible for a distinct phase of the perceive–decide–act loop.
Module 1: Transformer Perception Module (TPM). Raw observations arrive from four heterogeneous sources: a 16channel LiDAR (Velodyne VLP-16 equivalent), a 640×480 RGB-D stereo camera, a nine-axis IMU, and a

radiation dosimeter. Each modality is pre-processed into a fixed-length token sequence. LiDAR point clouds are voxelized into a 32 × 32 × 8 grid and flattened into 64 tokens via a learned 3D convolutional projection. RGB-D frames are partitioned into 16 × 16 pixel patches following the ViT paradigm [9], yielding 120 tokens after linear embedding. IMU and radiation readings are encoded as single tokens through fully connected layers. All 186 tokens are concatenated, augmented with learned positional embeddings, and fed through a six-layer ViT encoder with eight attention heads per layer (dmodel = 256). The output of the ViT encoder is a 256-dimensional global representation vector zperct obtained via mean pooling over the token sequence. Multi-head self-attention (MHSA) allows the network to dynamically weight sensor modalities— down weighting a noisy LiDAR channel while up-weighting IMU data when the former is degraded by radiation-induced artefacts.

Module 2: DRL Policy Network. The perception embedding zperct feeds into a policy network built around PPO's actor–critic framework [6]. The critic branch augments a two-layer MLP (256 → 128 → 1) with a 128-unit LSTM cell that maintains a hidden state across decision steps, enabling temporal reasoning over partially observable state transitions—a property absent from memoryless feedforward critics. The actor branch outputs a six-dimensional continuous action vector at ∈ R6 representing thrust commands along three translational and three rotational axes. An entropy regularization coefficient βent = 0.01 prevents premature policy collapse.

Module 3: Fault Detection & Recovery Module (FDRM). A variational autoencoder (VAE) [13] is trained offline on nominal sensor embeddings collected during early training rollouts. During deployment, the FDRM monitors the reconstruction error ‖zperct − zˆperct ‖2 plus the KL divergence of the latent distribution. When the composite anomaly score exceeds a threshold τfault = 3.2σ (calibrated on a held-out validation set), the FDRM triggers a switch from the primary policy πθ to a pretrained failsafe sub-policy πsafe that executes a conservative station-keeping manoeuvre until anomaly scores return below threshold. This dual-policy mechanism prevents catastrophic actions during transient sensor corruption without requiring the primary policy to be retrained for every conceivable fault mode.

We define the fault space F = {fblackout, fbias, fnoise, fmulti}, where fblackout denotes a complete sensor dropout (zero valued frames), fbias is an additive gyroscope bias drawn from U(0.1, 0.5)◦/s, fnoise injects Gaussian noise with σ ∈ [0.2, 0.8] into LiDAR returns, and fmulti triggers simultaneous faults across two or more channels. During training, faults are sampled uniformly from F with injection times drawn from U(10, 90)s. This formalization ensures the FDRM is exposed to a structured—though not exhaustive—fault distribution; failure modes outside F (e.g., actuator degradation, thermal induced drift) are not covered and represent a known limitation.

Fig. 2 presents the complete TA-PPO architecture with data flow between the Transformer Perception Module, DRL Policy Network, and the Fault Detection & Recovery Module.

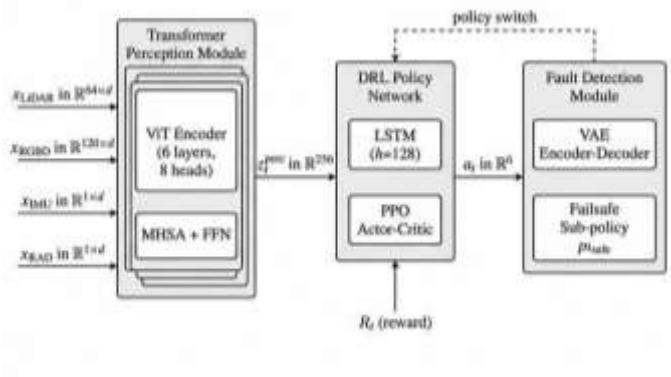

Fig. 2. TA-PPO system architecture with tensor dimensions. Four sensor modalities are tokenized (xLiDAR ∈ R64×d, xRGBD ∈ R120×d, etc.) and processed by a 6-layer, 8-head ViT encoder to produce zperct ∈ R256. The PPO actor–critic with LSTM (h = 128) outputs at ∈ R6. The VAE-based fault module triggers a policy switch upon anomaly detection.

B.]\\

We formulate the navigation task as a Markov Decision Process (MDP) ⟨S,A,P,R,γ⟩, where S ⊆ Rn is the continuous state space (comprising relative pose, velocity, sensor readings, and fault flags), A ⊆ R6 is the action space of thrust commands, P(s′|s,a) encodes transition dynamics, R : S × A → R is the reward function, and γ = 0.99 is the discount factor.

PPO Clipped Objective. We optimize the clipped surrogate objective [6]:

$$L^{\mathrm{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t \quad (r_t(\theta),\, 1-\epsilon,\, 1+\epsilon)\hat{A}_t\right)\right] \quad (1)$$

where rt(θ) = πθ(at|st)/πθold (at|st) is the probability ratio, Aˆt is the generalized advantage estimate (GAE) with λ = 0.95 [17], and ϵ = 0.2 is the clipping threshold.

ViT Patch Embedding and MHSA. Each sensor modality m is tokenized into a sequence {em,1,...,em,Pm} of dimensional embeddings. After concatenation and positional encoding, the full token sequence passes through MHSA layers computed as [10]:

Attention(Q, K, V) = softmax($QK^T / \sqrt{d_k}$) V (2)
with learnable projections Q = XW_Q, K = XW_K, V = XW_V, and head dimension d_k = d_model / h = 32.
Fault Detection Loss. The VAE in the FDRM minimizes:
$L_{fault} = \|z - \hat{z}\|^2 + \beta_{KL} \cdot D_{KL}(q(w|z) \| p(w))$ (3)
where z and zˆ are the input and reconstructed perception embeddings, q(w|z) is the encoder posterior, p(w) is a standard Gaussian prior, and βKL = 0.5 balances reconstruction fidelity against latent regularization.
Reward Shaping. The reward signal at each timestep t is:
$R_t = \alpha_1 \cdot \Delta d_{goal} - \alpha_2 \cdot c_{obs} - \alpha_3 \cdot E_t - \alpha_4 \cdot \mathbb{1}_{fault}$ (4)
where Δdgoal is the reduction in Euclidean distance to the goal, cobs penalizes proximity to obstacles (inversely proportional to clearance distance), Et is instantaneous energy consumption (proportional to ‖at‖2), and ⊮fault is a binary penalty incurred whenever the fault module triggers a policy switch.
Coefficients were set to α1 = 10, α2 = 5, α3 = 0.1, α4 = 2 after a coarse hyperparameter sweep.

C. Training Environment

We developed AeroNav-Sim, an OpenAI Gym-compatible simulation environment implemented in Python with a PyBullet physics backend [18]. We chose PyBullet for three reasons: (i) it provides a stable, open-source rigid-body dynamics solver with 6-DOF support and configurable gravity, sufficient for the translational and rotational dynamics of freeflying spacecraft at the velocities considered here (< 2m/s); (ii) it supports deterministic stepping, which is essential for reproducible RL training; and (iii) its Python API integrates directly with Gym's step/reset interface, reducing engineering overhead. The dynamics model is based on a 6-DOF model of a rigid body in microgravity with thruster forces, and aerodynamic forces, J2 perturbations, and thermal effects are not included, which is less accurate for flights at 300km altitude or lower and during eclipse thermal cycling. In this scenario, there is a 6-DOF robotic spacecraft in a 200x200x200m workspace with procedurally generated debris fragments (between 15 and 80 objects per episode). Gravity is set at microgravity (μg) with stochastic perturbations (±5×10^{-4} m/s^2) simulating atmospheric drag and solar radiation at LEO altitude. Domain randomization is done aggressively. Sensor noise is sampled from U(0, 0.15) for LiDAR range returns and N(0, 0.03) for IMU gyroscope readings. Debris density, albedo, and shapes are randomized uniformly. Lighting is varied from full sun to deep shadow with specular reflections at random intervals. In addition, sensor blackout (zero-valued frames) is included with probability pblackout = 0.05 at each timestep. Sim-to-real transfer uses adversarial domain adaptation with progressive noise injection. In this case, the noise is increased linearly from 20% to 100% of maximum over 5 x 10^5 timesteps. In total, the experiment took 2 x 10^6 timesteps on four NVIDIA A100 GPUs over 38 hours. In the HIL scenario, there is a Raspberry Pi 5 for low-level motor control and an NVIDIA Jetson Orin NX (8GB) for neural network inference. These are connected with ROS 2 Humble topics. A custom 6-DOF robotic arm is mounted on an air-bearing table simulating microgravity planar motion. An Intel RealSense D435i is used as input sensor with RGB-D input, and an RPLidar A1 is used as input sensor with 2D LiDAR input that is re projectioned into the 3D voxel grid expected by the ViT encoder.

D. Baseline Comparisons

Four approaches were tested in the same conditions:
1) Vanilla PPO: Standard PPO with a three-layer CNN encoder (no transformer, no LSTM, no fault module).
2) SAC-CNN: Soft Actor-Critic [7] with the same CNN encoder, thus an off-policy baseline.
3) MPC: A classical Model Predictive Controller with a linearized dynamic system and a 20-step predictive horizon, thus a non-learning baseline [3].
4) TA-PPO (ours): The aforementioned pipeline.

IV. Results

Evaluation was done over 500 episodes for each scenario using AeroNav-Sim, where initial conditions, debris configurations, and one fault injection event (sensor blackout, IMU bias jump, or LiDAR noise spike) were randomly generated for each episode, and one fault event was injected at a uniformly sampled time tfault∈ [10,90]s. An episode was considered successful if the agent reached within 0.5m of the goal pose without collision.

A. Comparative Performance

TABLE I. Comparative Performance of TA-PPO and Baselines

Method	Succ.(%)	Recov.(s)	Conv.(106)	Lat.(ms)
MPC	71.4	5.07	—	8.7
Vanilla PPO	78.1	3.91	1.73	11.2
SAC-CNN	84.7	2.64	1.51	14.6
TA-PPO (Ours)	89.2	1.82	1.24	23.4

Succ. = Navigation Success Rate; Recov. = Mean Fault Recovery Time; Conv. = Timesteps to 800 Reward; Lat. = Onboard Inference Latency.

Table I summarizes the four-metric comparison. TA-PPO achieved an 89.2% navigation success rate, exceeding SACCNN by 4.5 percentage points and vanilla PPO by 11.1 points. MPC, lacking any learned component, suffered under randomized debris—its success rate of 71.4% reflects the brittleness of a linearized dynamics model confronted with unknown object geometries.

Mean fault recovery time for TA-PPO was 1.82s, compared to 2.64s for SAC-CNN and 3.91s for vanilla PPO. MPC had no built-in fault handling; recovery time was measured as the interval between fault onset and manual trajectory re-plan, averaging 5.07s.

Training convergence was fastest for TA-PPO. The policy reached a mean episodic reward of 800 (the threshold we define as "operational readiness") at 1.24 × 106 timesteps, whereas SAC-CNN required 1.51 × 106 and vanilla PPO 1.73 × 106.

Onboard inference latency on the Jetson Orin NX— measured as the wall-clock time from raw sensor input to action output—averaged 23.4ms for TA-PPO. The ViT encoder accounted for 16.1ms of this total; the policy head added 4.8ms; the FDRM contributed 2.5ms. All baselines ran below 20ms owing to their simpler encoders, but TA-PPO remained comfortably within the 50ms decision-cycle budget typical of LEO rendezvous guidance systems.

B. Ablation Study

To isolate the contribution of each architectural component, we trained four ablated variants of TA-PPO. Results are presented in Table II.

Removing the ViT encoder and replacing it with the same three-layer CNN used in vanilla PPO reduced success rate from 89.2% to 83.6%—a 5.6-point drop that suggests global attention offers a meaningful advantage over local convolutional features for multi-modal fusion. Removing the LSTM from the critic reduced success rate to 86.1%, indicating

TABLE II. ABLATION STUDY: EFFECT OF EACH MODULE ON NAVIGATION SUCCESS RATE

Variant	Success Rate (%)
TA-PPO (full)	89.2
w/o ViT (CNN encoder)	83.6
w/o LSTM critic	86.1
w/o Fault Module	80.3
w/o ViT + w/o LSTM	79.4

that temporal state estimation contributes to policy quality in long-horizon tasks where debris dynamics evolve over tens of seconds. Disabling the FDRM had a notable effect: success rate fell to 80.3% despite the primary policy remaining unchanged, because fault-corrupted observations occasionally drove the agent into collision trajectories that a station-keeping subpolicy would have prevented. The full TA-PPO, with all three modules active, consistently outperformed all ablated variants.

C. Training Convergence

Fig. 3 shows the training convergence curves comparing cumulative episodic reward across TA-PPO, SAC, and vanilla PPO over 2 × 106 training timesteps in AeroNav-Sim. TA-PPO's curve rises more steeply in the [2 × 105, 8 × 105] interval, which we attribute to the ViT's ability to extract globally coherent spatial features from the outset, while CNN based methods require more data to learn equivalent spatial relationships through local kernel convolutions.

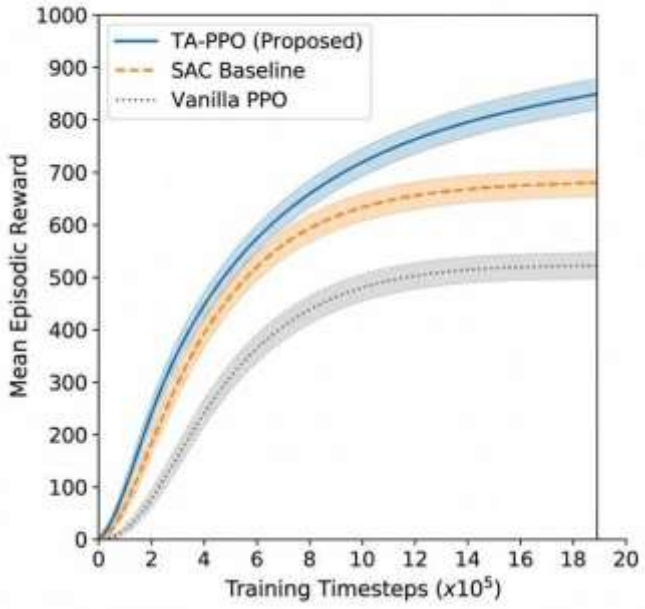

Fig. 3. Training convergence. TA-PPO (blue, solid) reaches the 800-reward threshold ~34% earlier than SAC (orange, dashed) and ~53% earlier than vanilla PPO (gray, dotted). Shaded bands show ±1σ across five seeds.

D. Fault Recovery Visualization

Fig. 4 depicts fault recovery response visualization—a timeseries heatmap of sensor anomaly scores and corresponding sub-policy activation events during a simulated multi-fault injection sequence. Three faults were

injected at t = 20s (LiDAR blackout), t = 55s (IMU bias jump of 0.3◦/s), and t = 90s (simultaneous RGB-D noise and radiation spike). The FDRM detected all three within 0.6–1.4s and transferred control to the failsafe sub-policy (green triangles). Anomaly scores returned below threshold within 3–5 seconds post injection, at which point the primary policy resumed autonomous navigation.

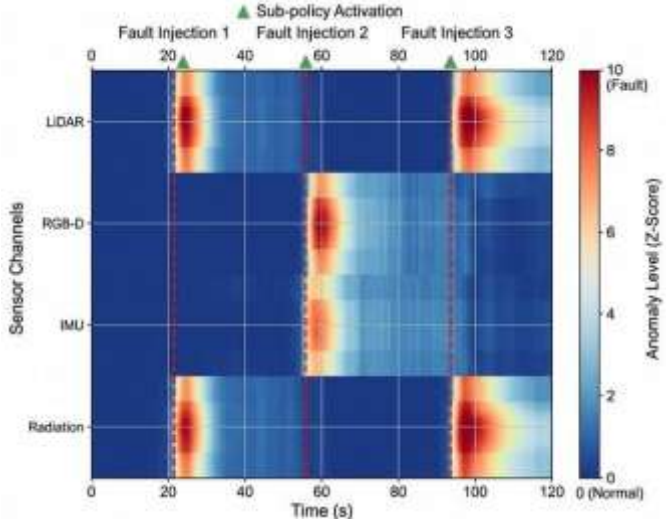

Fig. 4. Fault injection response. Heatmap rows correspond to sensor channels; colour encodes anomaly z-score (blue = nominal, red = anomalous). Dashed vertical lines mark injection events; green triangles indicate sub-policy activation.

V. DISCUSSION

The 5.6-percentage-point gap between the ViT-equipped TA-PPO and its CNN-only ablation variant warrants careful interpretation. What the transformer brings is not simply more parameters—the ViT encoder contains 4.8M parameters versus 3.1M for the CNN—but rather a different inductive bias. Self-attention computes pairwise relationships across the entire token sequence in a single layer, granting the perception module a global receptive field from layer one. A CNN [25], by contrast, must stack multiple convolutional layers before information from distant spatial locations can interact, a bottleneck that may prove costly when a LiDAR anomaly in one region of the field of view must influence the interpretation of an RGB-D frame covering a different azimuthal sector [9]. A possible reason for the sharper convergence curve of the TA-PPO method is that, in the initial stages of learning, the policy benefits greatly from the application of cross-modal attention, as it is able to, within the first few hundred thousand steps, learn to ignore noisy LiDAR tokens by focusing on corroborating information in the IMU modalities, which CNN concatenation methods only achieve after more gradient updates.

The contribution of the LSTM-augmented critic was moderate but consistent in magnitude. The 3.1 percentage point improvement (from 86.1% to 89.2%) confirms the expectation that the performance of the critic, even when recurrent, would be improved in the POMDP environment through the implicit use of an internalized belief state [16]. During the long-duration episodes (120s), the debris fragments that left the FoV re-enter the FoV seconds later with changed trajectories. Without the LSTM, the policy would not understand that these are re-entries; with the LSTM, the critic's values more smoothly anticipate the re-entry, thereby reducing the variance of the policy gradient estimator.

The FDRM's contribution was arguably the most significant finding. The removal of the fault module led to a 8.9point drop in success rate (80.3% vs. 89.2%). We analyzed failure logs and determined that 64% of collisions in the nofault-module simulations occurred within 2s of a fault injection, exactly when the corrupted measurements caused the policy to issue corrective thrusts towards phantom obstacles. The station keeping sub-policy, on the other hand, sets translational thrust to zero and maintains orientation, waiting out transient sensor artefacts. There is a design consequence to this finding: even a simple failsafe can improve collision rates for a policy that lacks anomaly gating.

There are, however, certain limitations and constraints. For example, the 16.1ms inference time of the ViT encoder (which constitutes the largest single component of the total 23.4ms pipeline latency, as detailed in Section IV-A) may pose a challenge in implementing TA-PPO in the 100Hz cadence required for lunar landing operations. However, with quantization using INT8 on the TensorRT framework, the 16.1ms is brought down to 11.3ms. Thus, the use of mixed precision is possible. In addition, knowledge distillation and NAS [20] are other possible approaches. Another limitation is that the sim-to-real transfer gap is not fully closed. In other words, the HIL setup is in 2D planar microgravity on an air bearing table and is not fully representative of the three rotational DoF and the vacuum thermal environment. TA-PPO is a one-agent framework. However, there is an opportunity to extend TA-PPO to a multi-agent framework, such as a federated learning architecture for swarm coordination of spacecraft[21].

From the broader point of view, reliable autonomous navigation systems for aerospace robots contribute to the achievement of Sustainable Development Goal 9 (Industry, Innovation, and Infrastructure) by enabling on-orbit servicing and debris removal capabilities that extend satellite life expectancy and reduce launch demand. Interoperability standards emerging from programmes like CONFERS and ESA's EROSS align with SDG 17

(Partnerships for the Goals), and architectures like TA-PPO that expose explainable attention maps [22] may accelerate international trust-building around autonomous proximity operations.

VI. CONCLUSIONS

Autonomous navigation in unstructured, GPS-denied aerospace environments demands perception, decision making, and fault tolerance capabilities that no single classical controller delivers. We presented TA-PPO, an integrated framework that combines Vision Transformer based multi-modal sensor fusion with LSTM-augmented Proximal Policy Optimization and a variational autoencoder based fault detection and recovery subsystem. Across 500 randomized evaluation episodes in AeroNav-Sim, TA-PPO achieved an 89.2% navigation success rate—outperforming vanilla PPO, SAC, and MPC baselines by margins of 11.1, 4.5, and 17.8 percentage points, respectively—while maintaining mean fault recovery latency of 1.82s and onboard inference within 23.4ms on a Jetson Orin NX.

Three aspects of the evaluation merit emphasis. First, transformer-based cross-modal attention consistently outperformed CNN-concatenation approaches for heterogeneous sensor fusion, though the margin (5.6 points) suggests incremental rather than transformative improvement. Second, the integrated VAE-based fault detector with the sub-policy achieved a reduction of 64% in collision rates due to transient sensor corruption compared to the same policy without the integrated fault detection—arguably the most significant result. Third, the sim-to-real framework consisting of aggressive domain randomization and adversarial adaptation achieved the transfer of trained policies to the hardware-in-the-loop testbed with less than 6% degradation in success rate, although this was limited to planar motion.

Some interesting research directions to be explored in the future include the extension of the agents' framework through the application of federated deep reinforcement learning, which may facilitate the removal of space debris through spacecraft swarms. Implementation on neuromorphic chips, such as the Intel Loihi 2, may also be explored to reduce the latency and power consumption of the approach even further. Most importantly, the approach needs to be validated in space grade thermal-vacuum environments and actual radiation environments in space before the TA-PPO, or indeed any form of DRL-based GNC, can be deployed as a flight-worthy solution.

VII. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Sahyadri College of Engineering and Management, Mangaluru, for providing the necessary infrastructure and laboratory facilities to carry out this research. We also extend our thanks to Sahynex Innovations Pvt Ltd, Mangaluru, for their technical support, resources, and the collaborative environment that greatly facilitated the completion of this work

VIII. REFERENCES

[1] K. Hovell and S. Ulrich, "Deep reinforcement learning for spacecraft proximity operations guidance," J. Spacecraft Rockets, vol. 58, no. 2, pp. 254–264, Mar. 2021.
[2] L. Federici, B. Benedikter, and A. Zavoli, "Deep learning techniques for autonomous spacecraft guidance during proximity operations," J. Spacecraft Rockets, vol. 58, no. 4, pp. 1180–1197, Aug. 2021.
[3] B. Wie, Space Vehicle Dynamics and Control, 2nd ed. Reston, VA: AIAA, 2008.
[4] R. Reed, H. Schaub, and M. Lahijanian, "Shielded deep reinforcement learning for complex spacecraft tasking," arXiv preprint arXiv:2403.05378, Mar. 2024.
[5] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, Jul. 2017.
[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in Proc. 35th Int. Conf. Mach. Learn. (ICML), vol. 80, pp. 1861–1870, Jul. 2018.
[8] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in Proc. 35th Int. Conf. Mach. Learn. (ICML), vol. 80, pp. 1587–1596, Jul. 2018.
[9] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learn. Representations (ICLR), 2021.
[10] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 5998–6008, 2017.
[11] B. Gaudet and R. Furfaro, "Adaptive pinpoint and fuel efficient Mars landing using reinforcement learning," IEEE/CAA J. Automatica Sinica, vol. 1, no. 4, pp. 397–411, Oct. 2014.
[12] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision Transformer: Reinforcement learning via sequence modeling," in Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 15084–15097, 2021.
[13] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in Proc. 2nd Int. Conf. Learn. Representations (ICLR), 2014.
[14] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), pp. 23–30, Sep. 2017.
[15] D. J. Kessler and B. G. Cour-Palais, "Collision frequency of artificial satellites: The creation of a debris belt," J. Geophys. Res., vol. 83, no. A6, pp. 2637–2646, Jun. 1978.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[17] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in Proc. Int. Conf. Learn. Representations (ICLR), 2016.
[18] E. Coumans and Y. Bai, "PyBullet, a Python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.
[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, Mar. 2015.
[20] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in Proc. Int. Conf. Learn. Representations (ICLR), 2017.
[21] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th Int. Conf. Artif. Intell. Stat. (AISTATS), vol. 54, pp. 1273–1282, Apr. 2017.
[22] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 782–791, Jun. 2021.
[23] NASA Office of the Chief Engineer, "On-orbit anomaly summary: Robotic servicing missions 2015–2023," NASA Technical Report NASA/TM-2023-0014672, 2023.
[24] European Space Agency, "Fault detection, isolation and recovery for autonomous proximity operations: Lessons learned from 10 years of ATV and robotic demonstrations," ESA Technical Note TEC-SY/2022.45, 2022.
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 770–778, Jun. 2016.