InFeR: Informed Failure Resilience in Learned Visual Navigation Control

Zishuo Wang[1] and Joel Loo[1] and David Hsu[1]

***Abstract*—While imitation learning (IL) has enabled successful visual navigation in many common environments, IL policies are prone to unpredictable failures under out-of-distribution (OOD) scenarios. This necessitates *failure-resilient* policies, which not only detect failures, but also recognise their sources and recover from them autonomously. We propose *InFeR*, a general framework for building IL policies with *informed failure resilience* without failure or recovery demonstrations. InFeR retrains an IL policy with a Variational Information Bottleneck (VIB) loss to structure its latent space for OOD failure detection. It applies a visual explainability technique, Grad-CAM, to localise an image region as the source of failure and inform a heuristic policy for recovery. All these are achieved without requiring additional training data. Real-world experiments show that InFeR enables informed failure recovery for IL policies, yielding robust long-range navigation in complex environments.**

I. Introduction

Visual navigation is an attractive approach to robot navigation, leveraging rich visual information from low-cost sensors [5]. Imitation learning (IL) has emerged as a key method to learn visual navigation policies [2, 23, 24], but is inherently limited by training data. IL policies may fail unpredictably on inputs outside the training distribution, often without clear explanation [8, 21, 26]. We consider the problem of building IL policies with **informed failure resilience**, which we argue extends beyond detecting failures to also encompass autonomous, informed recovery from failures.

Current works addressing failure have two key limitations. First, they focus on detecting failures, with handling often limited to triggering operator interventions [12, 27]. Second, they often rely on external failure detectors, which lack access to policies' internal representations and distributions and may fail to capture their true failure modes [19]. These limitations motivate three desiderata for failure-resilient policies: **(i)** the ability to autonomously *recover* from failures; **(ii)** the ability to *recognise* failure sources to inform recovery; and **(iii)** detection and recognition grounded in the policy's internal representations to capture its true failure modes.

We propose **InFeR**, a general framework for building informed, failure-resilient visual navigation policies *without any failure or recovery demonstrations*. InFeR operates within the standard offline IL setting, learning policies from a single, static dataset of successful demonstrations, where failures are naturally approximated as scenarios outside the dataset's distribution [1, 21, 7, 26]. InFeR introduces a regularised training objective to equip the policy to detect and recognise out-of-distribution (OOD) elements in its inputs without additional data or changes to its architecture. To recover from OOD scenarios, InFeR pairs the IL policy with a recovery policy guided by OOD recognition. Learning a recovery policy is ill-posed since failure and recovery examples are absent from data. Instead, InFeR designs a heuristic to greedily distance the robot from the identified failure source, then perturb it back to a nominal, in-distribution state.

Fig. 1. **InFeR enables autonomous recovery from policy failures.** Limited training data causes failures in OOD scenarios (*e.g.* sensor failures, close dynamic obstacles etc.). InFeR *detects* these, and further *recognises* failure sources in the input image, enabling *informed failure recovery*.

Specifically, InFeR retrains an IL policy on the original data, into an OOD-aware policy that detects OOD image inputs and localises the image regions responsible. This is done with a Variational Information Bottleneck (VIB) loss, which structures the latent space between vision encoder and action decoder to distinguish in- and out-of-distribution inputs. This enables us to compute an OOD detection score, and to apply Grad-CAM explainability methods to recognise OOD regions [22, 16]. As shown in Figure 1, InFeR handles diverse failure modes by detecting and recognising OOD elements (*e.g.* dynamic obstacles, close-up obstructions, sensor failures), then performs recovery informed in real-time by feedback from OOD recognition. We show that InFeR enables IL policy DECISION [2] to detect and recover from critical failures in the real world, enabling a robot to complete a challenging 300m indoor/outdoor route without intervention.

[1]School of Computing & Smart Systems Institute, National University of Singapore. {zishuo.wang, joell, dyhsu}@comp.nus.edu.sg

II. Related Work

Navigating with high-dimensional image inputs is challenging for classical robotic systems [5]. Recent progress in end-to-end imitation learning (IL) has proven effective at enabling robust visual navigation in the wild [2, 23, 24, 11]. However, such policies are opaque and offer little indication of impending failure [19]. Robust navigation requires failure detection [20]—however, IL datasets typically include only successful trajectories, necessitating unsupervised methods for failure detection, without any failure data. Thus, failure detection in IL is often framed as identifying OOD inputs that lie outside the dataset's distribution [21]. This can be approached via epistemic uncertainty estimation [15] using ensembles or Bayesian neural networks [19], which tend to be computationally intensive. On the other hand, autoencoders [21, 26], density estimators [25, 6], and Random Network Distillation [13] are more efficient but use detectors external to the policy which may not fully capture its failure modes and tend toward conservative predictions [19]. FAIL-Detect [27] wraps various OOD-scoring approaches with conformal prediction for OOD detection with statistical guarantees.

Robust navigation further requires the ability to independently recover from failures. Existing approaches largely respond to detected failures by terminating the episode and seeking human intervention [14, 19, 12]. A small subset of works autonomously recover from failures by executing predefined actions open-loop—*e.g.* returning to a default state [26] or executing a default behaviour [21]. Some systems layer increasingly aggressive preset actions [17] to improve recovery chances. [9] further requests interventions and learns from subsequent recovery demonstrations. Instead, InFeR informs recovery with OOD recognition. Failure detection and recognition is embedded in the policy, capturing its true failure modes.

III. Problem formulation

IL for visual navigation learns a policy $\pi_{\text{IL}} : \mathcal{O} \rightarrow \mathcal{A}$ from a dataset $\mathcal{D} = \{(o_1, a_1), \ldots, (o_N, a_N)\}$, where $o_i \in \mathcal{O}$ are RGB or depth images, and $a_i \in \mathcal{A}$ are actions typically given as target velocities. An OOD-aware policy is $\pi_{\text{IL-OOD}} : \mathcal{O} \rightarrow \mathcal{A} \times \mathcal{B} \times \mathcal{M}$, mapping observation o_t to action a_t, binary OOD detection $b_t \in \mathcal{B}$ and a representation of the location of OOD regions influencing the detection, $M_t \in \mathcal{M}$. We seek to obtain $\pi_{\text{IL-OOD}}$, given a specific policy network architecture and $\mathcal{D}$. Since many IL datasets only contain successful demonstrations, $\pi_{\text{IL-OOD}}$ must be obtained without requiring additional failure data. We also seek to obtain π_{Rec}, a recovery policy which modulates or overrides $\pi_{\text{IL-OOD}}$ to take corrective actions in OOD scenarios, selecting appropriate actions on the basis of spatial feedback from OOD recognition M_t. π_{Rec}'s actions either recover the robot to an in-distribution state, or halt it and call for external assistance if recovery is not possible. $\pi_{\text{IL-OOD}}$ and π_{Rec} together constitute a failure-resilient policy.

IV. Approach

InFeR (Figure 2) provides a framework to modify visual navigation policies trained with IL, π_{IL}, into OOD-aware policies, $\pi_{\text{IL-OOD}}$. It also provides a heuristic recovery policy, π_{Rec}, that makes corrective actions in OOD scenarios based on OOD recognition information from $\pi_{\text{IL-OOD}}$.

A. OOD-aware learned navigation policy, $\pi_{\text{IL-OOD}}$

OOD-awareness involves detecting OOD scenarios and pinpointing the corresponding OOD regions in the visual input. We emphasise that OOD-awareness should be task-relevant, identifying OOD scenarios and pinpointing OOD regions related to failures on the navigation task the policy is trained for. Our policy, $\pi_{\text{IL-OOD}}$, takes an image input $I_t \in \mathbb{R}^{3\times H\times W}$ and outputs action a_t, a binary OOD flag b_t, and a heatmap $M_t \in \mathbb{R}^{H\times W}$ that identifies the OOD features in I_t. InFeR augments a policy π_{IL} into an OOD-aware one in two phases. First, during training, a structured latent space capturing task-relevant factors from I_t is learned by augmenting behaviour cloning with additional regularisation. Second, during inference, conformal prediction and Grad-CAM are applied to features from the policy's latent space for OOD detection and recognition respectively.

Learning task-relevant OOD-awareness with VIB. We learn task-relevant latent representations Z, with the Variational Information Bottleneck [3]. This compresses input observations O into Z, while maximally retaining information crucial to predicting output actions A. Given $I(\cdot;\cdot)$ as mutual information, we maximise:

$$\mathcal{L}_{\text{VIB}} = I(A; Z) - \beta I(Z; O) \tag{1}$$

Following [3], we apply variational approximations to optimise this intractable objective, yielding the loss:

$$\max_{\phi,\theta} \frac{1}{|\mathcal{D}|} \sum_{(o,a)\in\mathcal{D}} \mathbb{E}_{q_\phi(z|o)}[\log p_\theta(a|z)] - \beta \text{KL}[q_\phi(z|o) \| p(z)] \tag{2}$$

where q_ϕ, p_θ are the visual encoder and action decoder, respectively. The first term simplifies to a behaviour cloning loss, while the second term regularises the latent representation Z to stay close to prior $p(z)$, which we define as a unit Gaussian $\mathcal{N}(0, \mathbf{I})$. This regularisation means that at inference, an OOD input $\hat{o}$ is likely to produce latent feature $\hat{z}$ that diverges significantly from the prior. Thus the KL divergence $\text{KL}[q_\phi(z|o) \| \mathcal{N}(0, \mathbf{I})]$ can serve as an effective scalar score indicative of OOD inputs during inference.

Learning with VIB can be viewed as augmenting a standard IL setup with additional regularisation of the latent features between a policy's visual encoder and action decoder. This serves as a lightweight approach to instill task-relevant OOD-awareness into IL policies spanning a range of architectures, without supervision using explicit failure data.

OOD detection with conformal prediction. Binary detection b_t involves thresholding scalar OOD scores, but setting appropriate thresholds is challenging, especially since the

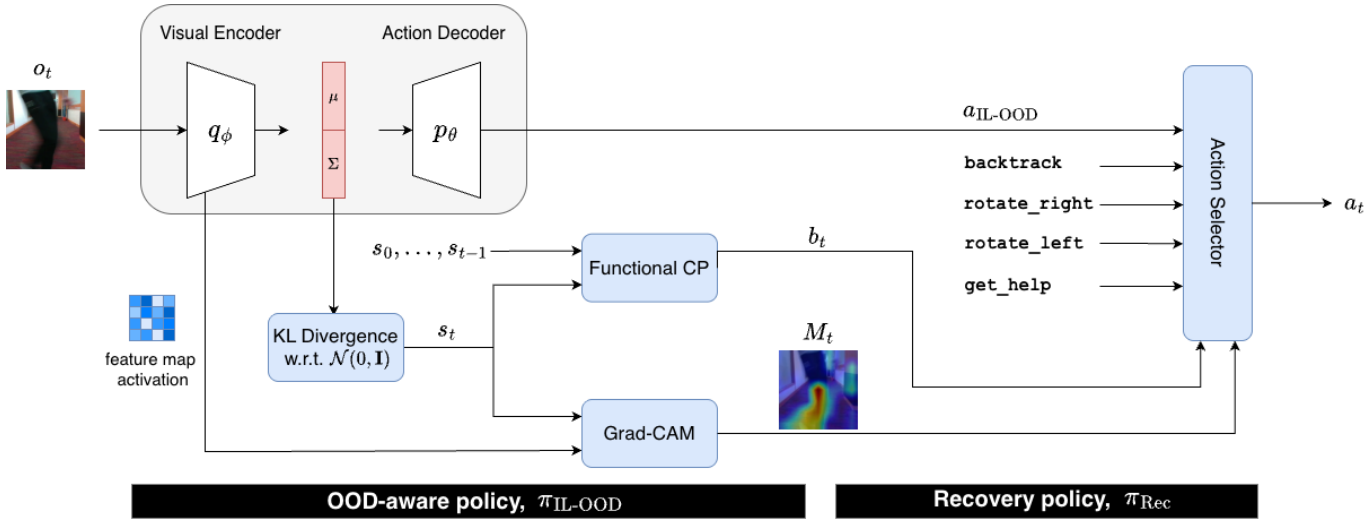

Fig. 2. **InFeR framework for informed failure resilience.** InFeR achieves informed failure resilience by augmenting learned policies with OOD-awareness. Apart from generating actions, OOD-aware policies detect when OOD inputs are encountered (b_t) and localise the OOD features in the observations (M_t). Detection and recognition enable informed recovery from OOD or failure scenarios. InFeR employs a heuristic recovery policy taking b_t, M_t as inputs.

threshold may be time-varying due to changing dynamics and scene appearance [27]. Conformal prediction (CP), particularly functional CP, offers a principled, distribution-free way to compute thresholds by generating a prediction band C_α at a user-defined significance level $\alpha \in [0, 1]$. Under mild assumptions, C_α contains the true score with probability $1 - \alpha$, enabling confident rejection of scores outside the band [10, 27]. Functional CP is an efficient, training-free method that wraps around a sequence of scores from any model.

We apply functional CP to a calibration set of trajectories to obtain time-varying thresholds on the KL scores. Similar to [27], we predict a one-sided C_α that provides an upper bound on whether a score is in-distribution with probability $1-\alpha$. We calibrate on a held-out split from the IL dataset, $\mathcal{C}$. We chunk trajectories into segments of T steps to standardise length, obtaining $\mathcal{C}_{\text{scores}} = \{(s_{i,0}, \ldots, s_{i,T}) : i = 1, \ldots, |\mathcal{C}|\}$. Following [10], we split $\mathcal{C}$ into $\mathcal{C}_\mu$ to compute the mean score across trajectories for each timestep, and $\mathcal{C}_w$ to compute the width of the band C_α. Given the mean scores $\mu = (s_0, \ldots, s_T)$, w is computed based on the maximum deviations from μ per trajectory. Specifically, given

$$S = \{d_j : j = 1, \ldots, |\mathcal{C}_w|\} \text{ where } d_j = \max_{t \in [0,T]} \{s_t - \mu_t\}, \quad (3)$$

w is the $(1 - \alpha)$-quantile of S. The prediction band is the set of intervals $\{[-\infty, \mu_t + w] : t = 1, \ldots, T\}$, where scores above the upper bound are rejected as OOD.

OOD recognition with Grad-CAM. Recognition aims to identify regions in observations $o \in \mathcal{O}$ which contribute to the OOD detection. To do so, we apply Grad-CAM [22], a visual explanation method for interpreting CNN outputs. Grad-CAM is a training-free module integrated into $\pi_{\text{IL-OOD}}$ at test time, offering efficiency as it requires only a single backward pass through the visual encoder.

Grad-CAM produces recognition heatmaps that highlight spatial regions influencing the model's predictions. It computes gradients of the prediction with respect to feature map activations in preceding convolutional layers, obtaining weights capturing the contribution of each feature map to the prediction. To localise OOD-related visual features, we apply Grad-CAM on the KL-based OOD scores predicted by the model. As deeper layers tend to capture higher-level semantic concepts (*e.g.* object parts) [18], we use Grad-CAM on the last layer of the visual encoder q_ϕ to encourage meaningful recognition results. Specifically, we compute the weight for the k-th feature map of the layer as:

$$\alpha_k = \frac{1}{N} \sum_i \sum_j \frac{\partial \text{KL}[q_\phi(z_t|o_t) \| \mathcal{N}(0, \mathbf{I})]}{\partial A^k_{ij}} \quad (4)$$

where A^k is the k-th feature map and N normalises across all A^k. The recognition result is:

$$M_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k A^k\right) \quad (5)$$

The resulting $M_{\text{Grad-CAM}}$ has the same dimensions as A^k, which we upsample with bilinear interpolation to obtain M.

B. Heuristic recovery policy, π_{Rec}

The recovery policy aims to take informed corrective actions when OOD scenarios are detected, and otherwise pass through $\pi_{\text{IL-OOD}}$'s actions during normal operation. Our policy, π_{Rec}, selects the appropriate corrective action from a discrete set of macro-actions. This set is designed to provide primitives that **(i)** *backtrack* the robot through known safe states, **(ii)** *perturb* it out of the OOD scenario, **(iii)** halt the robot and *seek human assistance*, if recovery is unrealistic and some termination condition is reached. We define the appropriate corrective action to be that which maximally avoids OOD features in the observation o_t. π_{Rec} uses the OOD recognition

TABLE I

EVALUATION OF INFER POLICIES. INFER IS APPLIED TO DECISION [2]. A "BLIND" VARIANT THAT DOES NOT USE RECOGNITION RESULTS, AND RANDOMLY SELECTS CORRECTIVE ACTIONS ON OOD DETECTIONS IS ALSO EVALUATED.

		InFeR-DEC			InFeR-DEC-Blind		
Failure type	**Recoverable**	**Det. SR**	**Han. SR**	**Time (s)**	**Det. SR**	**Han. SR**	**Time (s)**
Sensor failure	✗	1	1	5.00	1	1	5.00
Blocked (Dead-end)	✗	1	1	8.48	1	0.7	17.39
Blocked (Local minima)	✓	1	0.8	6.66	1	0.7	10.39
Close-range dynamic obs.	✓	0.9	0.8	1.97	0.9	0.5	1.95

heatmap M_t to identify OOD features and inform selection of corrective actions.

Concretely, π_{Rec}'s action set comprises {`backtrack`, `rotate_left`, `rotate_right`, `get_help`}. π_{Rec} selects among the first 3 actions during recovery, with `get_help` triggered only after $t_{\text{max_tries}}$ attempts to recover without returning to a non-OOD state as indicated by b_t. To select among the first 3 actions, we discretise M_t along the horizontal axis into 3 bins. Each bin is marked as containing OOD features if the sum of pixel values exceeds a threshold. π_{Rec} selects `rotate_left`/`rotate_right` if the right/left bins are OOD, to perturb the robot away from the OOD region. Otherwise, it will `backtrack`, which executes a cached sequence of past actions in reverse.

V. EXPERIMENTS

We evaluate InFeR's effectiveness in enabling real-world failure-resilient robot navigation. We train a DECISION policy [2] with our InFeR framework on 30 hours of teleoperated trajectories of path-following navigation in indoor and outdoor scenes. We identify three types of critical Navigation Failures (NF) which are OOD for this dataset:

NF1 **Sensor failure.** The robot receives blacked out images. The expected outcome is to seek human assistance, as the robot cannot recover on its own.

NF2 **Blocked pathways.** The robot's path is blocked by obstacles or untraversable regions. Since the dataset's trajectories focus on following clear paths, this is OOD. We consider **(i)** *local minima*, from which recovery is possible through a short sequence of local, reactive actions, and also **(ii)** *dead-ends*, which require substantially longer interventions or replanning to escape, and are irrecoverable for a local navigation policy.

NF3 **Close-range dynamic obstacles.** Sudden dynamic obstacles appearing close to the robot. As successful demonstrations avoid dynamic obstacles at a distance, such situations are OOD. The expected outcome is to halt and backtrack/perturb away from the sudden obstacle, then resume navigation along the path.

For each failure type, we run 10 trials, triggering a failure after ∼8s of normal operation. We report detection success rate (Det. SR), combined detection and recovery success rate (Han. SR), and time taken from first detection to recovery in Table I. To illustrate the importance of informed failure resilience, we show in our attached video that InFeR enables the DECISION policy to autonomously complete a 300m route with varied terrain (cluttered indoor plaza, outdoor park with forested paths) despite critical failures. We draw two conclusions:

InFeR effectively recovers from diverse real-world failures. Table I shows that InFeR enables high detection and recovery success across varied failure types, supporting **H2**. Figure 1 shows that InFeR enables task-relevant OOD recognition that contributes to the effective recovery: InFeR-DEC highlights features relevant to the path-following task. It performs especially well in irrecoverable scenarios, where consistent, robust OOD detection while the robot perturbs is needed to appropriately terminate recovery efforts and request human help. While overall detection is stable, InFeR-DEC occasionally fails in local minima due to spurious heatmap activations that guide the robot back into the minima. The exploration strategy can be improved to mitigate this by adding memory or temporally filtering the heatmap. InFeR also responds effectively to close-range dynamic obstacles, selecting timely corrective actions.

OOD recognition is key to efficient, effective recovery. We compare InFeR-DEC with an uninformed variant (InFeR-DEC-Blind) that ignores the recognition heatmap M_t, and randomly selects corrective actions. InFeR-DEC consistently outperforms InFeR-DEC-Blind, especially in recovering from close-range dynamic obstacles (**NF3**), where timely, informed actions are critical. While InFeR-DEC-Blind can occasionally recover in static environments via random perturbation, dynamic scenarios demand precise recognition. Thus, InFeR-DEC succeeds in 8/9 detected cases vs. 5/9 for InFeR-DEC-Blind. Informed recovery proves more efficient across all failure types, with InFeR-DEC recovering up to 2x faster than InFeR-DEC-Blind.

VI. CONCLUSION

We argue that **informed failure resilience** in IL policies extends beyond detection to encompass recognising failure sources and autonomously recovering from them. This paper introduces **InFeR**, a general framework for building failure-resilient visual navigation policies *without failure or recovery demonstrations*. InFeR enables OOD detection and recognition grounded in the policy's latent space, and pairs the policy with a separate recovery policy informed by real-time OOD recognition results. It requires no additional data, and no changes to the IL policy's architecture. InFeR generalises across a range of diverse failure modes across different scenes, enabling IL policies to succeed at long-range navigation in complex environments without intervention. This shows that InFeR is a clear step toward safe, robust navigation with IL.

References

[1] Christopher Agia, Rohan Sinha, Jingyun Yang, Ziang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *CoRL Workshop on Safe and Robust Robot Learning for Operation in the Real World*, 2024.

[2] Bo Ai, Wei Gao, David Hsu, et al. Deep visual navigation under partial observability. In *International Conference on Robotics and Automation (ICRA)*, 2022.

[3] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2017.

[4] Siddharth Ancha, Philip R. Osteen, and Nicholas Roy. Deep evidential uncertainty estimation for semantic segmentation under out-of-distribution obstacles. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[5] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53:263–296, 2008.

[6] Xiaoyi Cai, Siddharth Ancha, Lakshay Sharma, Philip R Osteen, Bernadette Bucher, Stephen Phillips, Jiuguang Wang, Michael Everett, Nicholas Roy, and Jonathan P How. EVORA: Deep Evidential Traversability Learning for Risk-Aware Off-Road Autonomy. *IEEE Transactions on Robotics*, 40:3756–3777, 2024.

[7] Ozan Çatal, Sam Leroux, Cedric De Boom, Tim Verbelen, and Bart Dhoedt. Anomaly detection for autonomous guided vehicles using bayesian surprise. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[8] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[9] Francesco Del Duchetto, Ayse Kucukyilmaz, Luca Iocchi, and Marc Hanheide. Do not make the same mistakes again and again: Learning local recovery policies for navigation from human demonstrations. *IEEE Robotics and Automation Letters*, 3(4):4084–4091, 2018.

[10] Jacopo Diquigiovanni, Matteo Fontana, and Simone Vantini. The importance of being a band: Finite-sample exact distribution-free prediction sets for functional data. *Statistica Sinica*, 2025.

[11] Wei Gao, Bo Ai, Joel Loo, and David Hsu. Intentionnet: Map-lite visual navigation at the kilometre scale. *arXiv preprint arXiv:2407.03122*, 2024.

[12] Cem Gokmen, Daniel Ho, and Mohi Khansari. Asking for help: Failure prediction in behavioral cloning through value approximation. In *International Conference on Robotics and Automation (ICRA)*, 2023.

[13] Nantian He, Shaohui Li, Zhi Li, Yu Liu, and You He. ReDiffuser: Reliable decision-making using a diffuser with confidence estimation. In *International Conference on Machine Learning (ICML)*, 2024.

[14] Simon Hecker, Dengxin Dai, and Luc Van Gool. Failure prediction for autonomous driving. In *Intelligent Vehicles Symposium (IV)*, 2018.

[15] Masha Itkina and Mykel Kochenderfer. Interpretable self-aware neural networks for robust trajectory prediction. In *Conference on Robot Learning (CoRL)*, 2022.

[16] Saebom Leem and Hyunseok Seo. Attention guided cam: visual explanations of vision transformer guided by self-attention. In *AAAI conference on artificial intelligence (AAAI)*, 2024.

[17] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[18] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120 (3):233–255, 2016.

[19] Rowan McAllister, Gregory Kahn, Jeff Clune, and Sergey Levine. Robustness to out-of-distribution inputs via task-aware generative uncertainty. In *International Conference on Robotics and Automation (ICRA)*, 2019.

[20] Quazi Rahman, Peter Corke, and Feras Dayoub. Runtime monitoring of machine learning for robotic perception: A survey of emerging trends. *IEEE Access*, 9: 20067–20075, 2021.

[21] Charles Richter and Nicholas Roy. Safe visual navigation via deep learning and novelty detection. In *Robotics: Science and Systems (RSS)*, 2017.

[22] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2019. ISSN 1573-1405.

[23] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *International Conference on Robotics and Automation (ICRA)*, 2023.

[24] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[25] Lorenz Wellhausen, René Ranftl, and Marco Hutter. Safe robot navigation via multi-modal anomaly detection. *IEEE Robotics and Automation Letters*, 5(2):1326–1333, 2020.

[26] Josiah Wong, Albert Tung, Andrey Kurenkov, Ajay Mandlekar, Li Fei-Fei, Silvio Savarese, and Roberto Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[27] Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Ambrus, Haruki Nishimura, and Masha Itkina. Can

we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. *arXiv preprint arXiv:2503.08558*, 2025.

APPENDIX

A. Baselines comparison for OOD detection and recognition

We evaluate performance of OOD-aware policies on OOD detection and recognition, comparing them against selected baselines (Table II). The OOD-aware policies are based on INet [11] and DECISION [2]. The baselines belong to three common classes: *density-based* approaches learn a density estimator for the training distribution; *one-class discriminators* learn to output a continuous metric without directly modelling the training distribution; *reconstruction-based* approaches indirectly model the training distribution by learning to reproduce observations, with post-hoc metrics applied to detect and recognise OOD inputs.

For density-based methods, we fit a normalising flow (NF) to observations with a Gaussian prior, using estimated likelihoods as the OOD score. For one-class discriminators, we use Random Network Distillation (RND) [13], which trains a predictor to map training data to output of a fixed random function, using prediction error as a measure of sample novelty and the OOD score. In prior art, these two approaches detect OOD inputs but do not support recognition. Reconstruction-based methods include autoencoders (AE) [21] and variational autoencoders (VAE-R) [26]. We apply post-hoc metrics based on reconstruction error: mean-squared error for OOD detection, and pixel-wise errors for recognition. Since InFeR's detection/recognition metrics apply to any VIB-based model, we test them with VAE-KL, which shares weights with VAE-R, but uses KL scores and Grad-CAM for OOD detection and recognition instead.

We collect a test set of 180 trajectories: 90 with a single failure near the end, and 90 normal (Figure 1). Failures are evenly distributed across three types—sensor failure, blocked paths, and close-range dynamic obstacles (**NF1**-**NF3**). We exclude infeasible image goals, which is a failure specific to image goal-conditioned architectures. Half the trajectories are in novel office and garden environments; the rest are in novel areas that resemble training data. Each trajectory is 10s long, with failures occurring at the 8s mark. A failure is detected if any frame from 8s onward is classified as OOD. To annotate ground-truth recognition, we divide each frame vertically into three bins and mark the bins containing features related to navigation failure (*e.g.* dynamic objects, blockages), similar to the approach in subsection IV-B.

We illustrate OOD detection performance via ROC curves across CP significance levels α. To evaluate recognition independently of detection, we restrict analysis to failure trajectories and, for each method, to the detections obtained at a true positive rate of 0.9, normalising detection sensitivity across methods. On this subset, we compute per-bin ROC curves for the recognition heatmap, with scores given by $\mathbb{I}[b_t = 1] \sum_{m \in M_{t,k}} m$, where $b_t = 1$ is an OOD detection, and $M_{t,k}$ is the k-th bin of the heatmap M_t. This yields the per-bin recognition score only when the input is flagged as OOD. The results in Figure 3 support that InFeR produces OOD-aware policies effective at task-relevant OOD detection/recognition. We draw three conclusions:

TABLE II
SELECTED BASELINES FOR OOD DETECTION/RECOGNITION.

Type	Method	Det.	Loc.	Inputs
Density	*NF* [25]	Density	-	O_t
One-class	*RND* [13]	Novelty	-	O_t, A_t
Recon	*AE* [21]	Recon	Recon	O_t
	VAE-R	Recon	Recon	O_t
	VAE-KL	KL	Grad-CAM	O_t
InFeR (BC)	*InFeR-INet*	KL	Grad-CAM	O_t, A_t
	InFeR-DEC	KL	Grad-CAM	$O_{t-H}, \ldots, O_t, A_t$

OOD-aware policies outperform existing methods. InFeR's approach of integrating OOD detectors into IL policies outperforms state-of-the-art OOD detection and recognition approaches. Both InFeR methods achieve high AUC scores across *both* detection and recognition compared with density-based approaches (NF), one-class discriminators (RND), and reconstruction-based approaches (AE, VAE-R). InFeR-DEC achieves the best performance by a clear margin, with high TPR and relatively few false positives. It also outperforms InFeR-INet, since its temporally accumulated features offer richer information to distinguish OOD scenarios. Unlike other methods that detect and recognise independently of the policy (*e.g.* VAE-R, VAE-KL), InFeR embeds both in the policy, reducing computation.

KL scores, Grad-CAM are most effective metrics. Results for *Recon* and *InFeR* methods indicate that approaches using KL-based OOD scores for detection and Grad-CAM for recognition perform best at their respective tasks. AE and VAE-R show similar poor performance on detection and recognition when using MSE and pixel-wise reconstruction error respectively. However, VAE-KL achieves significantly stronger performance on both tasks despite having the same model weights as VAE-R, differing only in the use of KL scores and Grad-CAM as OOD measures. We attribute this to the fact that KL scores and Grad-CAM operate on *latent vectors* instead of high-dimensional reconstructed images. As latent vectors are compressed representations that learn to focus on task-relevant factors, they are less noisy information sources than reconstructed images. Qualitative analysis (Figure 4) bears this out: recognition results from VAE-R are significantly noisier than VAE-KL or InFeR methods, and overfit to high-frequency artifacts in the inputs. While recent works shift the focus to methods like density-based [25] or second-order [4, 27] approaches, we highlight that selecting the right OOD measures enables reconstruction methods to outperform these. This also validates InFeR's choice of scores for OOD detection and recognition.

Action supervision improves OOD recognition. We hypothesised that learning from both observations and actions

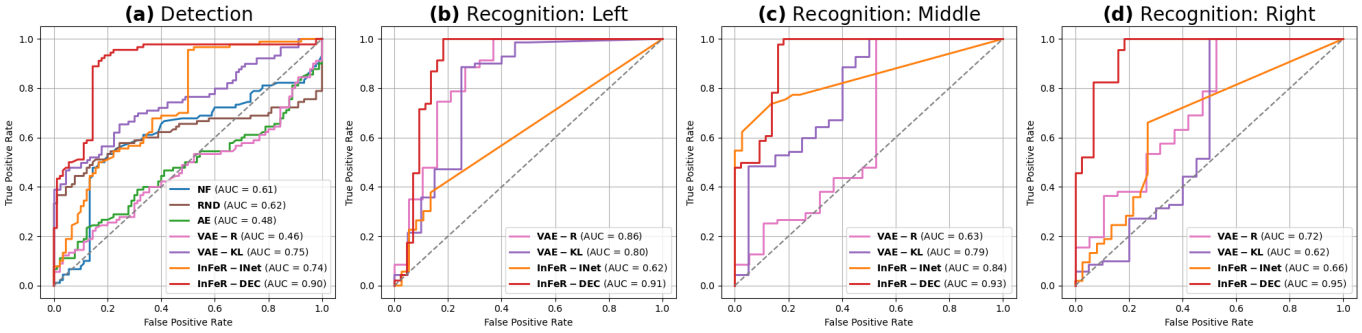

Fig. 3. **Comparison to baselines for OOD detection and recognition.** (*a*) ROC curves for OOD detection across CP significance levels α. (*b-d*): ROC curves for OOD recognition, based on scores from summing pixels across the heatmap. We compute scores per heatmap bin: *Left*, *Middle*, *Right*.

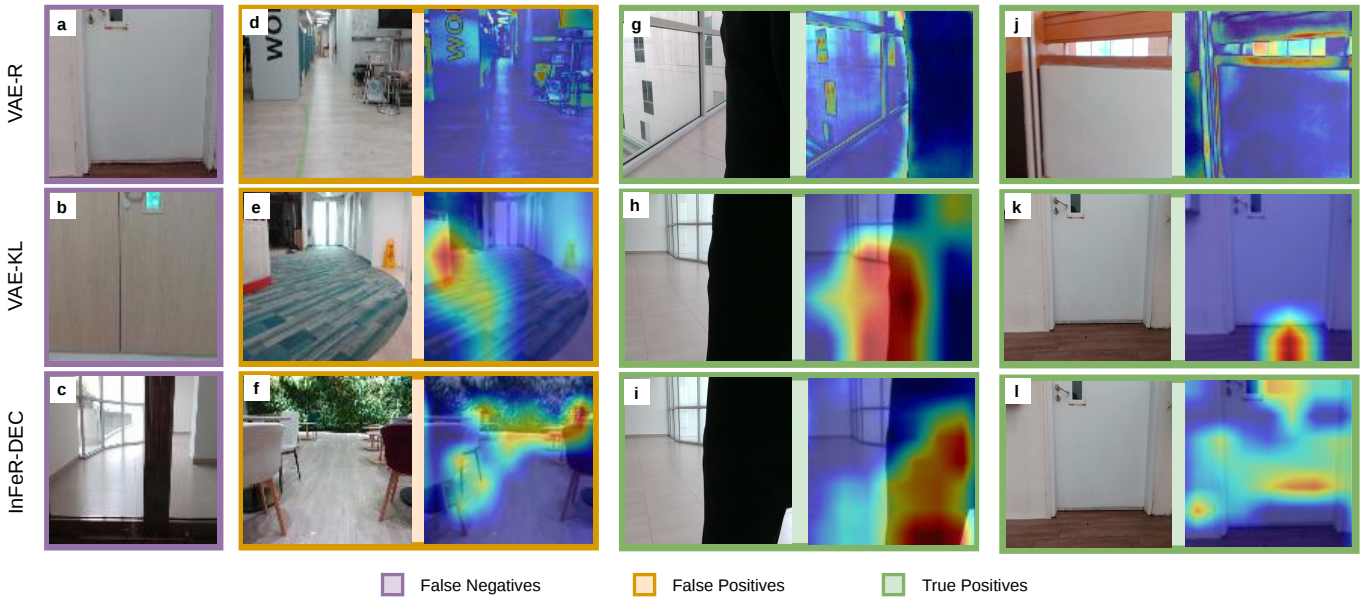

Fig. 4. **OOD detection and recognition examples.** We visualise selected reconstruction and InFeR methods. (*a-c*): Reconstruction-based methods struggle to detect failures like blocked pathways when there are few visual features. (*d-f*): VAE-R indiscriminately highlights high-frequency features, falsely detecting failures. InFeR-DEC is sensitive to obstacles (*e.g.* chairs, tables), prematurely triggering a detection. (*g-l*): VAE-KL offers smoother detections than VAE-R, but may not identify areas directly impacting navigation. In contrast, InFeR-DEC correctly highlights the agent's leg (*i*) and the door (*l*) as failure causes.

encourages more task-relevant representations. Observation-only methods (*e.g. Recon*) perform comparably on detection. However, adding actions enables better, task-relevant OOD recognition: in Figure 4(*k, l*), InFeR-DEC highlights features relevant to the path-following task (a closed door), whereas VAE-KL highlights irrelevant textures on the floor.

B. InFeR generalises across different policy architectures

We further evaluate InFeR's effectiveness in enabling failure-resilient navigation in the real world across different policy architectures. Specifically, we identify an additional failure type **NF4 Infeasible image goals**. It is specific to image goal-conditioned policies where the goal is dissimilar to current observations. We focus on the irrecoverable case where the goal is not locally reachable, and the expected outcome is for the robot to seek human assistance. We evaluate this failure type together with **NF1** and **NF3** on OOD-aware policy InFeR-GNM, which is built upon the General Navigation Model (GNM) [23], an image goal-conditioned policy that combines CNN encoders with MLP heads. Other experimental setup is the same as section V.

TABLE III
EVALUATION OF INFER-GNM. INFER IS APPLIED TO THE GENERAL NAVIGATION MODEL (GNM) [23].

Failure type	Recoverable	Det. SR	Han. SR	Time (s)
Sensor failure	✗	1	1	10.00
Close-range dynamic obs.	✓	0.9	0.7	8.21
Infeasible image goal	✗	1	1	16.00

Table III highlights InFeR's consistent and strong performance across different policy architectures and failure modes. InFeR-GNM's effectiveness on infeasible image goals shows that the VIB mechanism captures architecture-specific issues. Both InFeR-GNM and InFeR-DEC recover from common failures similarly well, underscoring InFeR's generalisability.

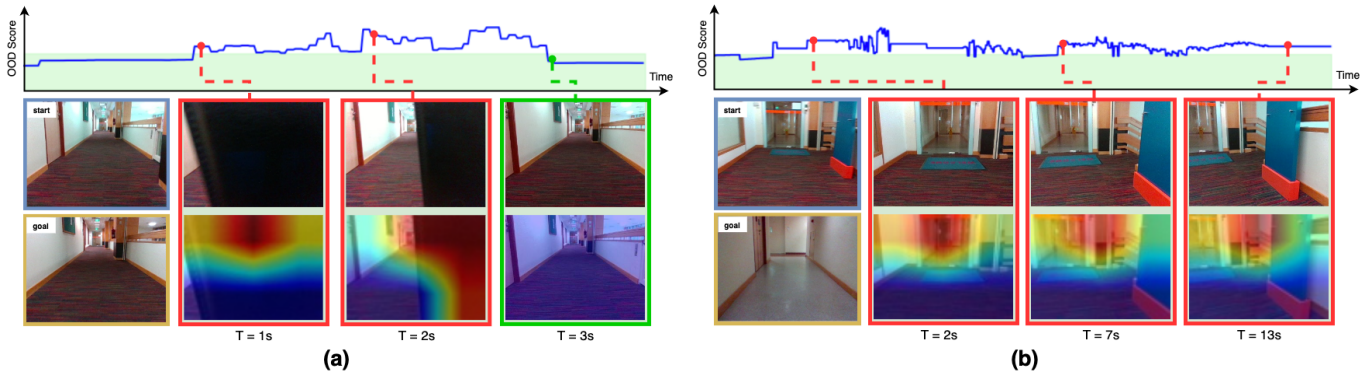

Fig. 5. **Informed failure recovery with InFeR-GNM.** (*a*): Robot is blocked by a pedestrian passing close by in front. It first backtracks (T=1s) to avoid the pedestrian, recognises the pedestrian on its right (T=2s) and pivots in the opposite direction to recover (T=3s). (*b*): Robot is issued an infeasible goal behind the glass door. It perturbs locally to various directions, but finds itself always blocked by obstacles, thus terminating and seeking help.

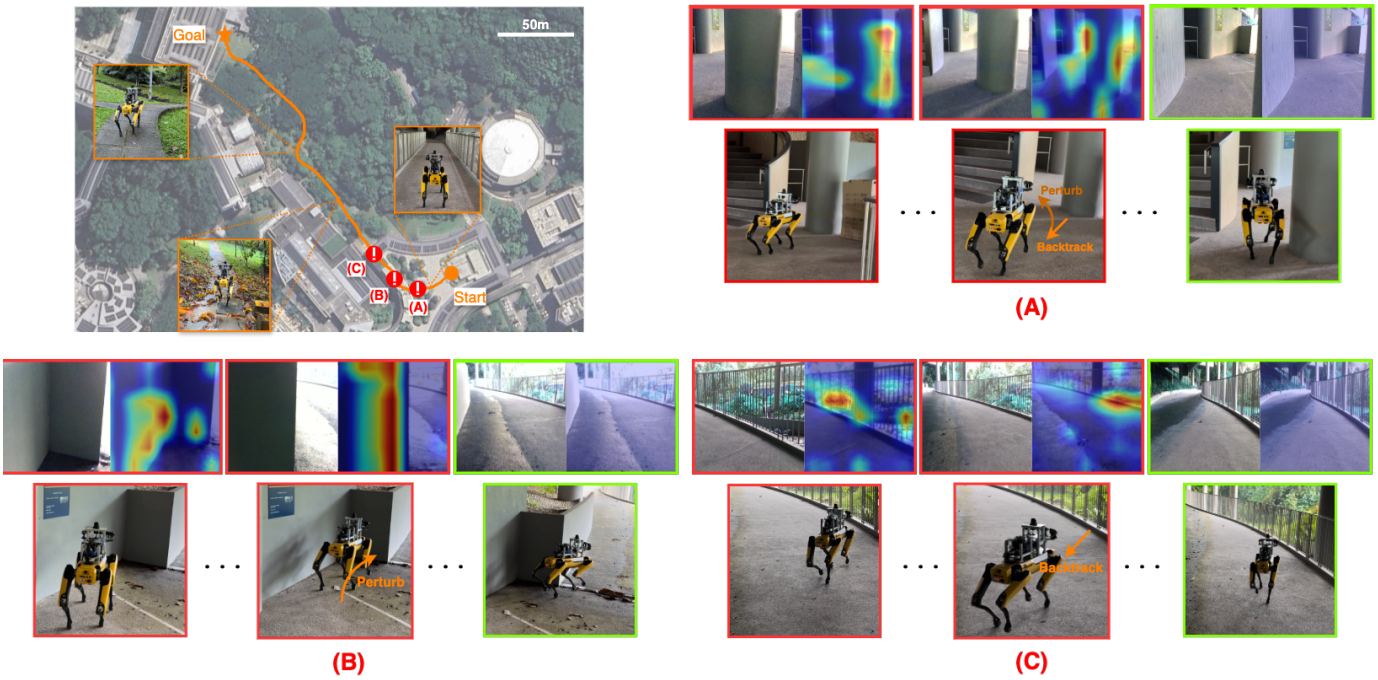

Fig. 6. **Informed recovery enables long-range navigation with InFeR-DEC.** The robot navigates a 300m route starting indoors then crossing a park. It recovers from the three failures shown: **(A)**: stuck in local minima due to clutter, has to backtrack and perturb to find a path; **(B)**: stuck in a corner, has to perturb to the right to continue on the path; **(C)** heading towards a railing, unseen during training, has to backtrack and align itself along the path.

A key factor influencing performance is the resolution of the visual encoder's final feature map. InFeR-DEC (Figure 6) has higher-resolution feature maps than the GNM (Figure 5), enabling finer recognition. This contributes to the GNM's poorer performance with close-range dynamic obstacles (**NF3**). Though InFeR-GNM shows slower recovery times due to gradual KL score decay, it still quickly responds with suitable behaviours like backtracking or perturbing away from obstacles (Figure 5).

C. Informed recovery enables long-range navigation

We show that InFeR allows the DECISION policy to autonomously complete a 300m route with varied terrain (cluttered indoor plaza, outdoor park with forested paths) despite critical failures in Figure 6.