SAFER: Incorporating Visual Context for Multitask Failure Detection in VLA Models

Püren Tap
Dept. of AI and Data Eng.
Istanbul Technical University
Istanbul, Turkey
0009-0004-3185-2430

Rafi Putra Abdurrahman
Dept. of AI and Data Eng.
Istanbul Technical University
Istanbul, Turkey
0009-0005-8067-9512

Sanem Sarıel
Dept. of AI and Data Eng.
Istanbul Technical University
Istanbul, Turkey
0000-0003-2993-6681

Sinan Kalkan
Dept. of Comp. Eng. & ROMER
Middle East Technical University
Ankara, Turkey
0000-0003-0915-5917

***Abstract*—As Vision-Language-Action (VLA) models are increasingly used in complex, multi-task robotic setups, automatic detection of execution failures has become critical for safety and reliability. Recent studies have demonstrated that latent action embeddings of a VLA can be leveraged to predict task success. However, relying solely on action embeddings neglects the environmental context that often precipitates failure.**

In this paper, we address this gap by first demonstrating that the distributions of image embeddings correlate with failure scenarios. Motivated by this finding, we propose SAFER, a fusion approach that integrates both action and image embeddings into a dedicated failure prediction module. Through extensive experiments, we show that image embeddings provide complementary information to action embeddings, providing up to 8% point improvement in seen and 2% point improvement on unseen tasks with 5.7x times faster inference over the state-of-the-art for detecting failures. Overall, SAFER underscores the necessity of visual context in monitoring the internal reasoning of VLA models and provides a more robust framework for autonomous failure detection. Code and trained models are available at: https://github.com/purentap/safer

***Index Terms*—manipulation failure detection, VLA, failure detection in VLAs**

I. Introduction

Recent advancements in large-scale robotic datasets [1], [2] have paved the way for generalist foundation models in robotics, referred to as Vision-Language-Action (VLA) models [3]–[7]. These models take as input one or more visual observations together with a natural language task description, and generate robot actions.

Despite their generalization promises, VLAs struggle in real-world scenarios. Real-world robotic manipulation is inherently stochastic, and unexpected situations may arise during task execution. Therefore, reliable and safe deployment in real-world settings requires robust failure detection mechanisms that can identify erroneous or unsafe states, see Fig. 1, across diverse tasks and execution conditions. In this paper, we primarily focus on detecting table-top manipulation failures on VLA models.

There are three main challenges in detecting failures in robotic manipulations: First, the failure detection methods must be lightweight and computationally efficient. This is critical since failure states must be detected in a timely manner [8], [9] as they occur, enabling immediate corrective actions. Second, failures should be detected early as possible. A predominant failure mode in learning based policies is error accumulation [10], making recovery difficult once the policy enters a faulty state. Early detection of failures is therefore essential for both preventing compounding errors and environmental safety. Third, to enable real-world deployment of VLAs, the failure detectors must generalize across diverse tasks and failure scenarios.

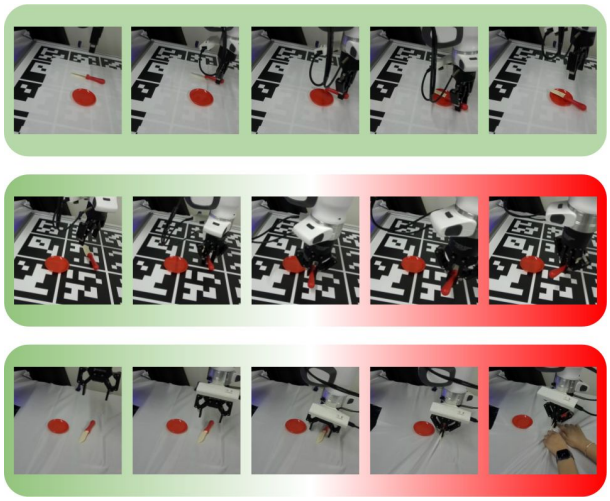


Fig. 1: Example successful (first row) and failed (second and third rows) rollouts from the SAFE Real World Dataset [8]. **Task:** Pick up the knife and put it on the plate. **Erroneous state:** In the second row, the robot gets stuck trying to lift the knife, and the maximum amount of timesteps is reached. **Unsafe state:** In the third row, the robot picks up the knife but also pulls the tablecloth, resulting in a failure.

Failure detection in robotic scenarios has received significant attention in the literature and has been addressed with different approaches. Prior work on failure detection focusing on generative policies leverages temporal consistency between the generated actions [11], learnable out-of-distribution (OOD) models via flow matching [9], supervised failure detection [8] or a large Vision-Language-Model (VLM) for both failure detection and reasoning [12], [13].

The most prominent, recent work is the SAFE model by Gu et al. [8]. Gu et al. proposed a lightweight, supervised multitask failure detection framework specifically focusing on VLAs. They demonstrated that a VLA's internal features from the final layer, before the action is decoded to tokens or a velocity field, carry informative signals about a task's success

or failure.

In this work, we introduce **SAFER**, as a lightweight framework that uses visual context in addition to action representations for failure detection. Specifically, we take inspiration from Gu et al.'s study on action embeddings and show that visual context captured by the image embeddings of a VLA's image encoder carry signals about a task's success or failure. Motivated by this result, we propose a learnable module that dynamically fuses image and action embeddings of a VLA to predict success or failure of a task. We evaluate our approach in both simulated [14], [15] and real world environments [8].

Contributions. The main contributions of our work are:

- We provide a qualitative analysis to show that image embeddings of a VLA's image encoder carry useful signals about whether the executed task results in success or failure. To be specific, we use the OpenVLA model's image embeddings in the LIBERO environment and display that failure and success cases are clustered in the image embedding space.
- Motivated by the results of our analysis, we propose SAFER, a lightweight architecture that performs early fusion of image embeddings and VLA latent features using an adaptive gating mechanism which is capable of failure detection at each timestep. SAFER extends prior model of SAFE [8], which only used action embeddings for failure detection.
- Our extensive experiments on several VLAs, environments and embodiments demonstrate significant gains in both performance and latency compared to the state-of-the-art model of SAFE as well as other baseline methods.

II. Related Work

A. Vision Language Action (VLA) Models

Early approaches on VLAs discretized continuous actions into bins by reusing the least frequent tokens from the large language model (LLM) vocabulary [3], [4]. Subsequent works [5], [16] introduced an additional action head module, which decodes the LLM's latent embeddings into continuous actions by diffusion or flow matching. More recent studies have proposed alternative decoding strategies, with an action tokenization scheme using discrete cosine transform [7] or a parallel decoding scheme [6]. Despite these significant advances, VLA models still exhibit performance limitations on unseen tasks and environments where task success rates may drop between 30% and 60% [5] which highlight the need for robust failure detection frameworks tailored to VLA policies [8].

B. Failure Detection in Robotic Manipulation

Failure detection in robotics has been a longstanding research topic [17]. Recent approaches to failure detection can be broadly categorized into two detection methods: Out-of-distribution (OOD) detection [9], [18]–[20] and supervised failure detection [8], [18], [19], [21], [22]. OOD-based methods treat failures as deviations from in-distribution data, whereas supervised approaches frame failure detection as a binary classification problem using episode level success/failure labels. A significant drawback of OOD methods is that they treat unseen scenarios as out-of-distribution, which conflicts with the idea of a generalist policy. In addition to detection, some studies also focus on reasoning about failures. For example, Liu et al. [23] constructs hierarchical summaries from robot past experiences and utilizes large language models (LLMs) for failure explanation and correction. Similarly, Duan et al. [12] fine-tunes a vision language model (VLM) on curated simulated failure data to jointly perform detection and reasoning. Lin et al. [13] proposes a VLM which can both detect and reason about the failures on VLAs, along with the corresponding recovery actions. While these methods demonstrate strong generalization and interpretability, they are limited by long inference times. As a result, they are not suitable for manipulation failures that require immediate corrective actions. Consequently, in such scenarios, a lightweight failure detection method is preferred.

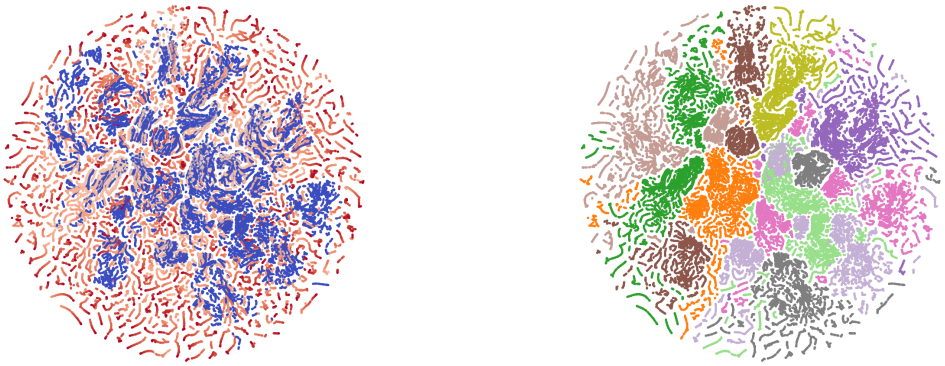
Recent work on OOD approaches [9] for failure detection on generative policies, conducts an extensive evaluation of learned and post-hoc score candidates, identifying logPZO [9] and RND [24] as the most effective, which we adopt as baselines in this work.

The most recent work in the supervised setting focuses on multi-task failures on VLAs [8], and proposes lightweight modules such as MLP or LSTM that leverage VLA latent embeddings. We argue that performant failure detection systems should leverage both the latent action embeddings and image embeddings, which together carry useful signals about both the environment and the model's behavior. Building on [8], our work incorporates the learned image embeddings from VLA's image encoder with adaptive gating into the failure detection framework.

III. Problem Formulation

In this work, our goal is to detect when a robot policy fails during task execution. At timestep *t*, a VLA takes in an observation $o_t = [I_t^1, .., I_t^k, l_t, q_t]$, where I_t^k is the k^{th} camera view in RGB format which can either be a head, wrist or an external camera, l_t is the language instruction at that timestep, q_t is the robot state, and the model outputs an action chunk $A_t = [a_t, a_{t+1}, ..., a_{t+H-1}]$ for the next H timestep where the first $H'(H' < H)$ steps are taken, and the model outputs a new action chunk $A_{t+H'}$, at timestep $t + H'$. We notate the intermediate action feature vector as z_{a_t}, which refers to the final LLM hidden state, and the encoded patch embeddings obtained from the camera as z_{I_t}. Moreover, some VLA policies decode a series of m tokens before converting them to an action vector, which we denote as $W_t = [w_t^1, .., w_t^m]$.

To train the failure detection module, we perform inference using the already fine-tuned VLA policy and collect rollout trajectories $\tau_i = (z_{a_t}, W_t, A_t, z_{I_t}^k)$ where each trajectory corresponds to a single episode, and is annotated with a binary label $y_i \in \{0, 1\}$ indicating success or failure. If the task is completed successfully according to the required steps, it is labeled as 0; otherwise a failure occurs, it is labeled as 1.



(a) Projections of Failed and Successful Rollouts.

(b) Projections by Task ID.

Fig. 2: T-SNE projections of image embeddings from OpenVLA in the LIBERO environment. (a) Successful rollouts are shown in blue. Failed executions start from light red and shift towards a darker color through the end of execution. (b) Embeddings by task ID, which shows tasks form clusters, and when examined together with (a), we conclude that failed rollouts can be differentiated from the clusters.

All labels are assigned at the episode level. Following [8], we split the data into *seen* and *unseen* tasks. We use D_{train_seen} for training, D_{eval_seen} for validation, and perform tests on D_{eval_unseen}.

IV. Method

Our framework consists of two stages: First, we collect the trajectories, the VLA action and the image embeddings, by running inference on a VLA model, with no additional training. We perform early fusion of the collected embeddings by utilizing a Gated Multimodal Unit (GMU), see Fig. 3. The fused embeddings then passed as input to the LSTM for temporal modelling.

A. Motivation for Using Image Embeddings: An Analysis

We take inspiration from the study by Gu et al. [8], who hypothesized that the internal features of VLA models provide high-level and abstract knowledge about whether if a task execution is success or failure by observing a "failure zone" from the projected latent VLA embeddings. We conjecture that failure cases must manifest themselves with visual signals that can be captured with image embeddings.

To evaluate our hypothesis, we investigate whether image representations are informative about separation between successful and failed tasks and whether they can bring information about the model's uncertainty. We visualize the internal features by employing t-SNE [25] to reduce dimensionality to two components. As illustrated in Fig. 2a, we observe that using image embeddings from a VLA model's vision encoder backbone provides a clear distinction between failure and successful executions. When examined together with Fig. 2b, we observe that the successful task executions form clusters and failed rollouts are scattered around. Furthermore, the projected image embeddings of the earlier timesteps of failure executions cluster with successful embeddings and temporally diverge into the aforementioned failure regions.

Our analysis further reveals that, beyond raw action embeddings, image embeddings also contribute to separate success and failure cases. Moreover, these features increasingly reflect this distinction over time.

B. Gating Mechanism

SAFER performs an early fusion of visual and action modalities prior to a temporal modeling with an LSTM. As a fusion strategy, we employ a Gated Multimodal Unit (GMU) [26], which adaptively balances the contribution of each modality with a learnable gating mechanism.

Let $z_I \in R^{d_I}$ and $z_A \in R^{d_A}$ denote the image and VLA latent embeddings, respectively. The GMU works by first projecting each modality into a lower dimensional representation:

$$h_I = \tanh(W_I z_I^T), \tag{1}$$

$$h_A = \tanh(W_A z_A^T), \tag{2}$$

where W_I and W_A are learnable projection matrices and $h_I, h_A \in \mathbb{R}^{d_k}$. Then, the gating vector and $g \in \mathbb{R}^{d_k}$, $g_i \in [0, 1]$ is computed as:

$$g = \sigma(W_g \cdot [z_I, z_A]), \tag{3}$$

where $[\cdot, \cdot]$ is the concatenation operator. Finally, the fused multimodal representation is obtained by element-wise multiplication ($\odot$):

$$h = g \odot h_I + (1 - g) \odot h_A. \tag{4}$$

The resulting fused representation h is then fed into the LSTM for temporal modeling. All parameters $\Theta = \{W_I, W_A, W_g\}$ are learned jointly with the LSTM.

C. Failure Detection Framework

Following Gu et al. [8], we use a single-layer LSTM to model temporal dependencies over VLA representations. The input at each timestep consists of the fused feature embedding

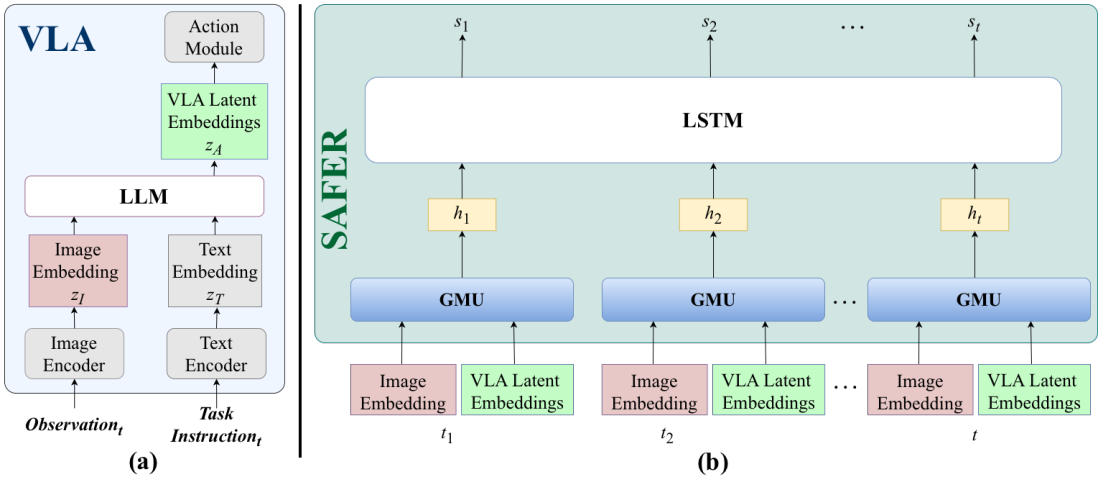


Fig. 3: **(a) Embedding Extraction:** We extract the image embeddings from the image encoder and the output latent embeddings from the LLM. **(b) SAFER:** Image embeddings and VLA latent embeddings are early fused within the GMU module, and then passed to the LSTM.

obtained from GMU by incorporating both the hidden representations and the mean pooled patch embeddings. Different aggregation strategies for VLA hidden representations were experimented by Gu et al. [8]. We have utilized the aggregation that worked best for each simulation and the embodiment.

The model takes the trajectory $T = [h_{t_1}, h_{t_2}...h_T]$ as input where h_t is the fused feature output of the GMU for timestep t. LSTM outputs a *failure probability* s_t at each timestep t through a final linear layer with the Sigmoid activation.

D. Loss

The model is trained using binary cross-entropy loss on all timesteps (t) of each episode i:

$$\mathcal{L} = -\sum_i \sum_t \left[y_i \log\left(s_t^{(i)}\right) + (1 - y_i) \log\left(1 - s_t^{(i)}\right) \right]. \quad (5)$$

Following prior work [8], we used an episode level label y_i to supervise the training for each timestep of the rollout.

V. Experiments

We experimented on four different VLA models in simulation, namely OpenVLA [4], π_0 [5], π_0-Fast [7] on LIBERO [14], open-pi-zero π_0^* [27] on SimplerEnv [15] and two distinct real world datasets [8] collected from a Franka Robot running π_0-Fast and WidowX manipulator running OpenVLA.

A. Experiment Environments

Our evaluation environments are replicated from [8], ensuring the comparability of our results against theirs by strictly adhering to the test cases and variables used in their work.

1) LIBERO: The LIBERO benchmark [14] provides a comprehensive collection of task suites, making it a suitable environment for evaluating VLA models [4]–[7]. We use the LIBERO-10 task suite in our experiments to test OpenVLA [4], π_0 [5], and π_0-FAST [7]. Similar to the experiments performed in [8], we randomly reserve 3 out of 10 tasks as unseen tasks.

2) SimplerEnv: In adherence with [8], we further evaluate our approach using SimplerEnv [15], which provides a high fidelity simulation environment modeling the RT-series [2], [3], [28] and BridgeData V2 [29] datasets. We utilize the pretrained π_0 models from a reproduction [27], denoted here as π_0*. We evaluate 4 tasks on each embodiment, with 3 tasks being seen and 1 unseen.

3) Real-Robot Experiments: In order to evaluate our proposed models on real world cases, we utilize the two datasets published by [8]. These datasets are comprised of distinct setups: one featuring the π_0-FAST model (finetuned on the DROID dataset [2]) deployed on a Franka Emika Panda Robot, and another utilizing the OpenVLA model (pretrained on the "Open-X Magic Soup++" dataset [1]) on a WidowX robot manipulator. For every rollout, the datasets include the rollout trajectories accompanied with a video of the rollout. Accordingly, we extract the image observation for each rollout trajectory from the videos, and take the corresponding visual embeddings by passing the observations through the vision encoder backbone of the respective VLA model. We reserve 3 out of 13 tasks from the Franka dataset and 2 out of 8 tasks from the WidowX dataset for unseen evaluation, treating the remaining tasks as seen during training.

B. Baseline Methods for Comparison

We compare our method with three families of failure detection approaches: (i) Embedding distance based, (ii) supervised learning based, and (iii) OOD based.

As a first step, we reproduced the results from [8]. Unlike the original work, which reports results from the last training epoch, we evaluate models based on the checkpoint with the best seen-task performance. We omit the token uncertainty and action consistency baselines, as the former consistently exhibits low performance in its findings, and the latter due to the high computational requirement during inference.

TABLE I: Simulation results, averaged AUROC score over 3 seeds. **Best** and second-best results are highlighted.

Model	**OpenVLA**		π_0		π_0**-Fast**		π_0^*		**Average**	
Simulation	**LIBERO**		**LIBERO**		**LIBERO**		**SimplerEnv**			
Split	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
Mahalanobis dist.	56.85±0.75	62.52±6.55	80.07± 6.31	79.39± 8.59	**92.59±2.54**	82.37± 4.26	89.37±2.72	73.92±14.94	79.72	74.55
Euclidean dist. k-NN	59.84±2.19	53.34±3.37	84.33± 3.20	77.41±18.09	89.34±2.31	81.46± 8.81	91.76±2.22	78.56± 9.18	81.31	72.69
Cosine dist. k-NN	60.32±2.04	67.03±7.31	84.49± 3.14	77.28±18.08	90.59±2.54	84.53± 7.11	91.81±1.78	80.32± 9.82	81.80	77.29
PCA-KMeans	61.32±3.43	52.25±2.79	69.15±10.42	45.28±24.76	72.55±5.76	70.01± 8.29	73.61±4.48	64.47±16.26	69.15	58.00
RND	59.32±5.44	47.17±7.94	85.18± 2.90	80.08±13.71	87.90±1.73	76.37± 6.30	91.01±1.55	65.84± 7.28	80.85	67.36
LogpZO	63.07±4.80	57.35±4.21	85.68± 2.45	76.78± 9.13	89.75±3.67	82.13± 6.22	92.12±1.73	72.95±12.78	82.65	72.30
SAFE-MLP	73.38±1.66	71.64±4.23	82.14± 3.12	76.65±11.64	84.63±2.48	69.95±22.32	95.03±2.77	80.10± 6.63	83.79	74.58
SAFE-LSTM	75.64±2.64	**71.71±5.58**	88.82± 1.15	78.75±15.19	92.08±1.84	83.92± 6.11	**96.23±1.92**	79.51± 7.40	88.19	78.47
SAFER	**77.52±4.14**	70.14±5.38	**89.68± 2.04**	**85.41± 6.42**	92.12±1.85	**85.37± 6.68**	96.15±0.95	**82.23± 9.55**	**88.86**	**80.78**

1) Embedding Distance Based Baselines: Methods based on embedding distance compute a failure score measuring distances of VLA latent embeddings without any additional learning. The training samples are partitioned into two sets, E_{succ} and E_{fail}, corresponding to successful and failed rollouts respectively. Then, for each VLA latent embedding $z_{a_t} \in \{D_{val\_seen}, D_{val\_unseen}\}$, a failure score is computed as follows: $s_t = d(z_{a_t}, E_{succ}) - d(z_{a_t}, E_{fail})$, where $d(\cdot, \cdot)$ measures the mean distance between a single vector and set of vectors.

We follow the baselines given in [8], and employ the three distance metrics: Mahalanobis, Euclidean and cosine distances averaged over the k nearest neighbors and PCA-kmeans [19].

2) Learned OOD Scores: Following [8], we employ RND [24] and LogpZO [9] as they are the best performing OOD scores shown in [9].

3) Supervised Failure Detection Baselines: We adopt [8] as a baseline for supervised failure detection where two distinct architectures, SAFE-MLP and SAFE-LSTM, are presented. Models are trained on both successful and failed rollouts, and evaluated on seen and unseen tasks.

C. Evaluations

We split the data into three sets, D_{train}, D_{val_seen} and D_{val_unseen}, for all methods. D_{val_seen} consists of tasks present in the training set, whereas D_{val_unseen} consists of tasks not observed during training. For each set of experiments, we perform a grid search over hyperparameters using three random seeds, and select the run with best performance on seen tasks. We report the mean are under the ROC curve (AUROC) for both seen and unseen tasks. Following the evaluation criteria of Gu et al. [8], we determine the minimum trajectory length across all rollouts for each task. To ensure consistent evaluation across trajectories of varying lengths, the rollout level failure probability is obtained by $s = \max(s_1, ..., s_{min\_task\_step})$ where s_t is the failure probability for each timestep and $s_{min\_task\_step}$ is the minimum trajectory length taken by the VLA model to complete the task.

TABLE II: Real World Experiments: Averaged AUROC scores, π_0-FAST on Franka over 5 seeds and OpenVLA on WidowX over 3 seeds. **Best** and second-best results are highlighted.

Model	π_0**-Fast**		**OpenVLA**	
Embodiment	**Franka**		**WidowX**	
Split	Seen	Unseen	Seen	Unseen
Mahalanobis dist.	67.45±5.26	48.84±2.14	90.09±4.21	88.12±3.07
Euclidean dist. k-NN	77.24±4.15	57.89±3.56	74.18±3.82	59.80±8.21
Cosine dist. k-NN	76.58±4.54	57.77±4.92	76.24±3.88	73.10±6.51
PCA-KMeans	50.02±3.86	47.15±2.55	78.94±4.39	65.26±7.33
RND	65.39±3.46	48.72±4.64	66.48±5.67	49.70±3.93
LogpZO	67.54±6.01	49.92±3.57	71.29±1.71	52.63±5.84
SAFE-LSTM	81.76±3.36	**62.91±5.16**	88.94±0.69	78.38±9.14
SAFE-MLP	78.91±3.33	58.72±6.31	92.81±2.48	**89.72±3.86**
SAFER	**86.39±2.29**	62.28±1.00	**93.88±1.62**	84.89±6.04

VI. Results

We report the mean AUROC scores for simulation and real-world experiments.

A. Simulation Results

The simulation results are listed in Table I. They indicate that incorporating image embeddings and the gating unit improves model performance on unseen tasks across all environments except OpenVLA on the LIBERO environment, resulting in $\sim$2% gain on average. While our approach improves failure detection on unseen tasks, it also enables the model to improve performance on seen tasks by $\sim$1% on average.

B. Real-world Results

To evaluate the real-world generalization of SAFER, we use the real-world environments described in Section V-A. The results are given in Table II. Results clearly show that a prominent performance gain is observed on both models and robot environments in seen cases. Notably, with π_0-Fast on Franka Robot, we observe improvements over the closest baseline achieving $\sim$ 8% increase on seen tasks while performing on par on unseen tasks.

C. Ablation Analysis

We conduct additional experiments to evaluate the effectiveness of the GMU. In Table III, we report the ablation in simulation environments, where LSTM(I) is trained solely on the extracted image embeddings, and LSTM(I+A) on concatenated image and VLA latent embeddings. Compared to LSTM(I+A), using the gating unit enables SAFER to achieve higher performance on both seen and unseen tasks by adaptively leveraging each modality.

TABLE III: Modality ablation analysis on simulation environments. **Best** and second-best results are highlighted.

Model	**OpenVLA**		π_0		π_0**-Fast**		π_0^*	
Simulation	**LIBERO**		**LIBERO**		**LIBERO**		**SimplerEnv**	
Split	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
LSTM (I)	70.74±4.70	61.62±11.13	83.43±4.20	69.72±16.32	89.49±4.14	79.31±3.82	95.07±2.22	73.95±21.56
LSTM (I+A)	73.24±2.62	62.36± 9.69	88.72±2.93	81.74± 7.65	90.14±2.89	82.31±6.35	95.78±1.41	80.73±11.26
I+A with GMU (**SAFER**)	**77.52±4.14**	**70.14± 5.38**	**89.68±2.04**	**85.41± 6.42**	**92.12±1.85**	**85.37±6.68**	**96.15±0.95**	**82.23± 9.55**

TABLE IV: Inference latency comparison. We report the mean and standard deviation of the measured inference times.

Model	**Mean Latency (ms)**
SAFE-LSTM	2.58±0.024
SAFER	0.45±0.005

D. Inference Latency

We evaluate the inference latency of SAFE-LSTM and SAFER using a single rollout trajectory per inference on an NVIDIA RTX 4060 Ti. Each model is executed 1000 times across 64 samples, following a warm-up phase, from the OpenVLA-LIBERO trajectories. The average latency is reported in Table IV. By leveraging lower-dimensional embeddings, SAFER achieves a 5.7× reduction in inference time.

VII. Conclusion

In this work, we first provide a qualitative analysis to show that visual context captured by image embeddings carry visual signals indicative of a VLA's success or failure on a task. Motivated by this finding, we introduce SAFER as a new lightweight model on VLAs, which fuses image embeddings and VLA latent features for timestep-level failure detection.

Our extensive evaluations in both simulation and real-robot experiments show that this fusion leads to improved performance. The significant gains we report confirm that visual context together with action embeddings can be used together for detecting failures in real-world scenarios.

Acknowledgment

This work was supported by Scientific Research Projects Department of Istanbul Technical University. Project Number: MYL-2025-47164. Computing resources used in this work were provided by the National Center for High Performance Computing of Turkey (UHeM). We acknowledge the use of proprietary LLMs in rephrasing our sentences in Abstract, Introduction, Related Work and Method sections.

References

[1] A. O'Neill *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0," in *Int. Conf. on Robotics and Automation (ICRA)*, 2024.

[2] A. Khazatsky *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," 2025.

[3] B. Zitkovich *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *Conf. on Robot Learning (CoRL)*, 2023.

[4] M. J. Kim *et al.*, "Openvla: An open-source vision-language-action model," *The 8th Conference on Robot Learning*, 2025.

[5] K. Black *et al.*, "π0: A vision-language-action flow model for general robot control," *ArXiv*, vol. abs/2410.24164, 2024.

[6] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *Robotics: Science and Systems*, 2025.

[7] K. Pertsch *et al.*, "Fast: Efficient action tokenization for vision-language-action models," 2025.

[8] Q. Gu *et al.*, "Safe: Multitask failure detection for vision-language-action models," *Neural Information Processing Systems (NeurIPS)*, 2025.

[9] C. Xu *et al.*, "Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies," *Robotics: Science and Systems*, 2025.

[10] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," *Int. Conference on Artificial Intelligence and Statistics*, 2011.

[11] C. Agia *et al.*, "Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress," *Conf. on Robot Learning (CoRL)*, 2025.

[12] J. Duan *et al.*, "Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation," *Int. Conf. on Learning Representations*.

[13] Z. Lin *et al.*, "Failsafe: Reasoning and recovery from failures in vision-language-action models," 2025.

[14] B. Liu *et al.*, "LIBERO: Benchmarking knowledge transfer for lifelong robot learning," in *Advances in Neural Inf. Processing Systems*, 2023.

[15] X. Li *et al.*, "Evaluating real-world robot manipulation policies in simulation," in *Annual Conf. on Robot Learning (CoRL)*, 2024.

[16] J. Wen *et al.*, "Tinyvla: Toward fast, data-efficient vision-language-action models for robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 10, no. 4, pp. 3988–3995, 2025.

[17] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.

[18] H. Liu, S. Dass, R. Martín-Martín, and Y. Zhu, "Model-based runtime monitoring with interactive imitation learning," *Int. Conf. on Robotics and Automation (ICRA)*.

[19] H. Liu *et al.*, "Multi-task interactive robot fleet learning with visual world models," *Conf. on Robot Learning (CoRL)*, 2024.

[20] R. Sinha, A. Elhafsi, C. Agia, M. Foutter, E. Schmerling, and M. Pavone, "Real-time anomaly detection and reactive planning with large language models," *Robotics: Science and Systems*, 2024.

[21] A. Inceoglu, E. E. Aksoy, A. Cihan Ak, and S. Sariel, "Fino-net: A deep multimodal sensor fusion framework for manipulation failure detection," *Int. Conference on Intelligent Robots and Systems (IROS)*, 2021.

[22] A. Inceoglu, E. E. Aksoy, and S. Sariel, "Multimodal detection and classification of robot manipulation failures," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1396–1403, 2024.

[23] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," *Conf. on Robot Learning (CoRL)*, 2023.

[24] N. He, S. Li, Z. Li, Y. Liu, and Y. He, "ReDiffuser: Reliable decision-making using a diffuser with confidence estimation," *Int. Conf. on Machine Learning*, 2024.

[25] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[26] J. Arevalo, T. Solorio, M. M. y Gómez, and F. A. González, "Gated multimodal networks," *Neural Computing and Applications*, vol. 32, pp. 10 209 – 10 228, 2020.

[27] A. Z. Ren, "open-pi-zero: Re-implementation of π_0 vision-language-action (VLA) model from physical intelligence," https://github.com/allenzren/open-pi-zero, 2025, version 0.1.1, released January 27, 2025. Accessed April 6, 2025.

[28] A. Brohan *et al.*, "RT-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[29] H. R. Walke *et al.*, "BridgeData V2: A dataset for robot learning at scale," *Conf. on Robot Learning (CoRL)*, 2023.