FailureGen: Scalable Robot Failure Data Generation with World Models

Akul Santhosh akuls@nvidia.com
Chaitra Sathe csathe@nvidia.com
Aastha Jhunjhunwala aasthaj@nvidia.com
NVIDIA

Project website: https://akul-nv.github.io/failuregen/

Abstract: Robot failures are difficult to collect at scale, limiting the range of failure scenarios represented in robot learning datasets. We present FailureGen, a method for generating synthetic videos with temporal event annotations by reusing action trajectories from recorded failures. The method comprises four stages: fine-tuning an action-conditioned Cosmos world model on successful and failed real executions; annotating source recordings to identify failure-associated donor segments and compatible successful hosts; transferring donor motion into host contexts, with optional spatial offsets, to generate continuous video rollouts; and annotating and filtering the generated samples according to observed events and visual quality. We study two implementations with different tasks and robot configurations, with visual failure detection serving as one downstream application. In a preliminary comparison, a detector trained on real and generated data achieves an average precision of 0.51, compared with 0.41 using real data alone, and detects 33 of 36 failure intervals, compared with 28 of 36. By reusing observed failures in additional contexts, FailureGen expands the data available for studying robot failures and developing downstream learning methods.

Keywords: Synthetic data generation, Robot failure generation, World models, Failure detection

1 Introduction

Collecting diverse examples of robot failures at scale is difficult. Failures can depend on specific combinations of robot motion, object configuration, and contact conditions. Capturing this variation requires repeated physical executions, and less frequent failure modes may still remain underrepresented.

We present FailureGen, a method for expanding failure datasets by reusing action trajectories from recorded failures. A failure-associated donor supplies the motion, while a successful host supplies the target scene and nominal trajectory. An adapted Cosmos world model [1] generates video rollouts conditioned on the host scene and transferred motion. Different hosts and optional spatial offsets vary the context and location of the intervention, producing additional examples without requiring a new physical execution for each video.

Prior work has explored synthetic failure construction through procedural perturbations in simulation, as in AHA [2], and through action-conditioned world models with multiple verifiers, as in Dream2Fix [3]. FailureGen offers a complementary approach by using observed failures as the source of motion interventions and expanding the contexts in which these behaviors are represented. This approach makes further use of failures that have already been captured, including events that would be difficult to collect repeatedly.

Accepted to the "Oops, I erred" Workshop at CoRL 2026.

Transferred motion may produce a different failure or a successful continuation in a new scene. Generated videos may also contain visual artifacts. We therefore label and filter rollouts according to their observed events and visual quality, producing data for downstream learning.

Section 4 examines two implementations with different tasks and robot configurations. We evaluate visual failure detection as one downstream application through a preliminary comparison of training on real-only and augmented data.

This paper makes three contributions:

- FailureGen, a method for expanding robot failure datasets by transferring recorded failure trajectories into compatible host scenes and generating video rollouts with an action-conditioned world model.
- Two experimental implementations applying the method to different tasks and robot configurations.
- A preliminary evaluation of visual failure detection as a downstream application, comparing detectors trained on real-only and augmented datasets.

2 Related Work

Simulation and procedural failure generation. Simulation provides controlled environments for generating robot failures and studying their consequences. AHA constructs failure examples through procedural perturbations for detection and reasoning [2]. FailSafe pairs generated failures with executable recovery actions, using simulator execution to validate proposed corrections [4]. DenseReward synthesizes failures such as missed grasps, collisions, and object drops to learn dense rewards for downstream planning and policy learning [5]. Guardian combines simulated action perturbations with supervision constructed from real recordings for planning and execution verification [6]. These methods demonstrate how targeted changes to robot execution can produce useful examples of unsuccessful behavior.

World-model-based data generation. Action-conditioned world models generate visual continuations from scene observations and robot motion. Ctrl-World and Cosmos provide models and adaptation tools for this form of generation [7, 8, 1], while RoboDream composes robot motion with scene conditioning for data synthesis [9]. DreamGen, VLAW, and WorldSample use generated experience to support policy learning and improvement [10, 11, 12]. For failure generation, Dream2Fix perturbs actions, generates candidate outcomes with a world model, and applies multiple verifiers to construct recovery supervision [3]. StressDream steers generated futures toward specified outcomes, including failures, by optimizing the initial diffusion noise [13]. FailureGen builds on conditional video generation by supplying motion derived from recorded failures and transferring it into compatible host contexts.

The reliability of these generated interactions is also an active research topic. MiraBench evaluates physical consistency, action following, and optimism bias [14]. WorldSimProbe examines action realization and interaction dynamics, including a donor–receiver setup that replays actions from one task in another scene [15]. This provides a related use of motion–scene composition for evaluating world models. Together, these studies motivate assessing generated motion and interaction outcomes alongside visual quality.

Trajectory reuse and data augmentation. Recorded demonstrations provide another basis for expanding robot datasets. MimicGen transforms demonstrated motion for new object configurations [16], while DemoGen adapts trajectories and synthesizes corresponding point-cloud observations through spatial transformations [17]. Real2Render2Real reconstructs object motion and appearance from recordings to render robot demonstrations without object-dynamics simulation [18]. GenAug and ROSIE expand visual diversity through generative edits to recorded observa-

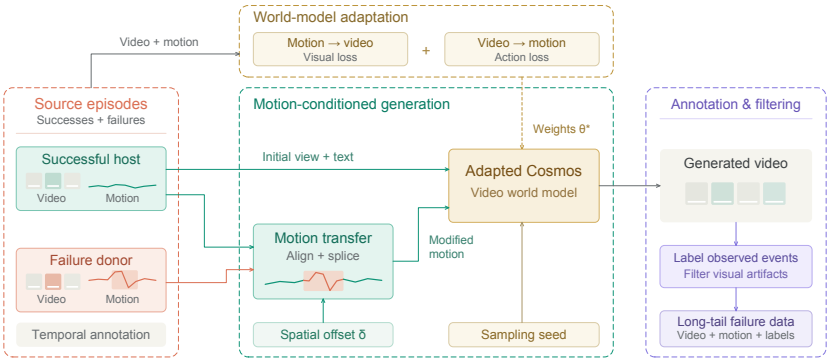

Figure 1: **FailureGen architecture.** Successful and failed source executions support world-model adaptation through forward- and inverse-dynamics supervision. At generation time, a successful host supplies visual context and nominal motion, while a failure donor supplies motion for a selected intervention. Motion transfer and optional spatial offsets construct the conditioning sequence for the adapted world model. Generated videos are labeled according to their observed events, with a cleanup filter for unreliable examples. Each retained example includes the video, its associated motion sequence, and temporal event labels.

tions [19, 20]. These approaches illustrate how existing experience can support additional training examples with reduced physical data collection.

These lines of work offer complementary ways to expand robot experience. Capturing the diversity of failures, particularly the long tail of events across changing scenes and interaction conditions, remains challenging. FailureGen contributes to this effort by reusing observed failure motion in additional compatible contexts. Such motion provides an additional source of interventions that could be combined with procedural perturbations, visual augmentation, and guided world-model generation to broaden failure coverage. The resulting data may support several learning objectives: prior work studies failure detection [21], diagnosis and corrective guidance [22], and reward learning [5]. This paper evaluates visual failure detection as one downstream application.

3 Method

Problem setup. We are given a source dataset $\mathcal{D}_{\text{src}}$ of successful and failed robot executions. Each episode contains synchronized visual observations $\mathbf{o}_{0:N}$ and measured robot states $\mathbf{s}_{0:N}$, where t indexes sampling instants and N is the number of transitions. The subscript range includes both endpoints. Our goal is to construct an additional dataset $\mathcal{D}_{\text{gen}}$ of generated video–motion pairs with temporal event annotations, expanding the contexts in which rare robot failures are represented. We call a recording that supplies failure-associated motion a *donor*, and a successful recording that supplies a target scene and nominal motion a *host*. The generated data can support multiple downstream learning tasks; in this paper, we evaluate its utility for visual failure detection.

As illustrated in Fig. 1, FailureGen comprises four components: world-model adaptation, source annotation, motion-conditioned generation, and annotation and filtering of generated data. We adapt the model using successful and failed executions, identify compatible donor–host segments, and transfer failure-associated motion into host scenes, optionally varying the intervention location. Finally, we label the generated events and filter unreliable examples.

3.1 World-model adaptation

We first encode recorded motion into the conditioning representation used by the world model. For a motion representation r, an encoder ψ_r maps consecutive measured states to a conditioning vector $\mathbf{a}_t = \psi_r(\mathbf{s}_t, \mathbf{s}_{t+1})$. Joint-space inputs can use $\Delta\mathbf{q}_t = \mathbf{q}_{t+1} - \mathbf{q}_t$, where $\mathbf{q}_t$ is the measured joint-

position vector. Relative end-effector inputs can use $\Delta T_t = T_t^{-1} T_{t+1}$, where T_t is a homogeneous end-effector pose matrix expressed in a fixed reference frame, measured directly or obtained through forward kinematics. Gripper or other tool states are included according to the embodiment. The resulting vectors describe observed motion; measured state transitions need not equal the commands sent to the robot.

We adapt a pretrained Cosmos world model [1] using aligned video–motion windows from successful and failed executions. We combine forward- and inverse-dynamics supervision with the aim of improving spatial dynamics: the visual objective supervises scene evolution under robot motion, while the action objective supervises motion associated with observed visual changes.

In forward-dynamics examples, the model generates a visual continuation conditioned on initial observations, the recorded motion sequence, and a text description. In inverse-dynamics examples, it predicts motion conditioned on the observed video. Each training sample is randomly assigned to one of these two directions, using the model's rectified-flow objective in video-latent or action space, respectively. Text descriptions provide task or execution context; each experimental study specifies whether task, phase, or event-specific captions are used.

Motion enters the model through domain-specific input and output projections, with a dedicated domain for each embodiment and motion representation. Fine-tuning updates the generation expert and its associated video, action, and conditioning modules, while keeping the remaining parameters frozen. Implementation details and study-specific adaptation settings are provided in Appendix A. Motion encoding, coordinate conventions, visual preprocessing, and normalization remain consistent between adaptation and generation. We denote the adapted parameters by θ^* and the forward model's conditional distribution over visual continuations by p_{θ^*}.

3.2 Temporal annotation of source data

We annotate the source recordings to locate motion segments that can be transferred. An agentic labeling pipeline using vision-capable LLMs (Appendix D) identifies task steps, failures, and recoveries, assigning each event a category and start and end times. We distinguish the failure event itself from its visible aftermath, and label these intervals independently of whether the episode eventually succeeds. Proposed labels are checked for consistent categories and valid timestamps, and can be corrected during review.

These annotations guide donor and host selection. We match failure-associated donor segments with successful host segments at corresponding task steps, checking compatibility of their robot and object configurations. Recordings are partitioned by episode before selection, with only training recordings serving as donors or hosts. Evaluation recordings are also excluded from world-model adaptation and normalization.

3.3 Motion-conditioned generation

For each selected donor–host pair, we align a failure-associated donor segment with an intervention window in the host. We index the host window from zero, let τ denote the intervention's starting transition within it, and let u denote the donor segment's starting transition. In addition to transferring donor motion, we can apply spatial offsets to vary where the intervention occurs. For a generation horizon of H motion transitions, the transfer operation constructs the conditioning sequence

$$\widetilde{\mathbf{a}}_{0:H-1} = \mathcal{T}_r\big(\mathbf{a}^h, \mathbf{a}^d, \mathbf{s}^h_\tau; \tau, u, \boldsymbol{\delta}, \eta\big)\,. \tag{1}$$

Here, superscripts h and d identify host and donor quantities, $\mathbf{a}^h$ and $\mathbf{a}^d$ are their encoded motion sequences, and $\mathbf{s}^h_\tau$ is the measured host state at intervention onset. The representation-specific operator $\mathcal{T}_r$ produces the modified sequence $\widetilde{\mathbf{a}}_{0:H-1}$. The parameters $\boldsymbol{\delta}$ specify optional spatial offsets or motion-excursion variations, while η specifies the transferred channels, intervention duration, and transition settings.

The operation anchors donor motion to the host configuration at intervention onset and applies the requested variation through representation r. Each embodiment specifies the offset coordinate frame and its conversion into conditioning inputs. Transition settings control how the modified segment joins the nominal host motion. Unselected channels retain host conditioning throughout; selected channels retain it outside the intervention and its transitions. Donor and host recordings must use compatible motion representations and kinematic conventions.

We then condition the adapted world model on the host's initial observation $\mathbf{o}_0^h$, the modified motion sequence, and a text description c:

$$\widehat{\mathbf{o}}_{1:H} \sim p_{\theta^*}\left(\cdot \mid \mathbf{o}_0^h, \widetilde{\mathbf{a}}_{0:H-1}, c\right). \tag{2}$$

The sequence $\widehat{\mathbf{o}}_{1:H}$ contains the H generated future observations, which follow the host's initial observation to form the generated clip. Spatial offsets can change where the transferred motion encounters objects and surfaces. The world model predicts the resulting visual continuation, allowing the same failure-associated behavior to unfold at different locations with different contact and object-motion dynamics. The observed outcome may therefore change even when the donor segment remains fixed.

We generate candidates by varying compatible donor–host pairs, motion parameters, and sampling seeds. Each candidate retains its source identifiers, conditioning sequence, transfer parameters, checkpoint, and sampling settings. The transferred motion specifies the intended intervention; the generated video determines the observed event and its timing.

3.4 Annotation and filtering

We annotate generated videos using the same labeling pipeline as the source data. Labels describe the events visible in each video, which may differ from the intended intervention. Because world models can produce visual artifacts, implausible interactions, or unclear outcomes, a cleanup filter can exclude clips that cannot provide reliable training examples. The annotation and filtering policy used for each study is specified in the experiments.

Each retained example contains the generated video, its associated motion sequence, temporal event labels, and source identifiers. These video–motion pairs form $\mathcal{D}_{\text{gen}}$ and provide data for downstream learning tasks.

4 Experimental Studies

Our experiments examine two aspects of FailureGen: its ability to expand the available failure data across host contexts, and the usefulness of these generated examples for downstream learning. We study an in-house industrial bimanual manipulation task and a focused three-task ViFailback follow-up. The industrial setting emphasizes multistage execution and interactions with surrounding equipment, while ViFailback examines additional manipulation scenarios and downstream model sizes.

Both settings use an adapted Cosmos3-Nano world model with forward- and inverse-dynamics supervision. Downstream evaluation compares fine-tuning on real data alone with fine-tuning on the same real data augmented with FailureGen examples. The industrial experiment additionally includes a pretrained reference without task-specific fine-tuning.

Visual verification and filtering. Generated videos are assessed using the agentic annotation pipeline and visually checked and verified by human reviewers. Review considers visual coherence, preservation of the host scene, and the events depicted in each generation. Unsuitable examples are removed before the data are considered for downstream use.

Across the accumulated generation experiments, filtering removes **9.8% of the industrial data** and **11.2% of the ViFailback data**, leaving **90.2% and 88.8%**, respectively, usable. These cumulative usability rates are reported separately from the experiment-specific selection of failure clips for downstream training. Review procedures and filtering details are provided in the appendix.

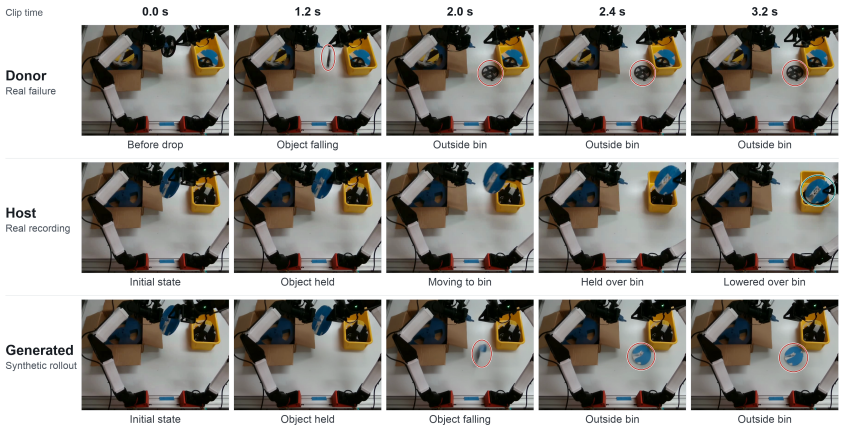

Figure 2: **Failure transfer in industrial manipulation.** Top: a real donor execution in which a reel falls outside the bin. Middle: the real host recording, which supplies the initial scene and object appearance. Bottom: the FailureGen continuation, which preserves the host context while reproducing the donor's drop behavior with different timing and object motion. Columns show elapsed clip time; red outlines highlight the falling or dropped object.

4.1 Industrial bimanual manipulation

Task and data. Our in-house dataset captures a bimanual YAM robot performing an industrial material-handling task: picking up an SMT reel, transferring it to the other arm, and placing it onto a stack. Successful execution requires coordinated grasping, handover, and placement. Failures include pickup misses, pickup slips, and handover drops, which can produce further interactions with the table, tote, and surrounding objects.

The source collection contains 80 real episodes: 45 clean executions, 12 pickup misses, 13 pickup slips, and 10 handover drops. World-model adaptation and downstream detector training use the same 66 training episodes. The remaining 14 recordings provide downstream validation data. Generator-specific validation and test allocations are detailed in Appendix B.

World-model conditioning uses measured end-effector motion and gripper state for both arms. Each arm contributes translation, a six-dimensional rotation representation, and gripper state, yielding a 20-dimensional conditioning representation.

Failure-data expansion. FailureGen transfers failure-associated motion into successful host recordings, reusing their scene appearance and initial robot–object configurations. This allows a failure mechanism to unfold in additional contexts: for example, the consequences of releasing a reel depend on its position relative to the table and tote.

For the downstream experiment reported here, the annotation pipeline identifies **75 synthetic failure clips from 108 donor-transfer candidates**. These clips span **26 distinct host recordings and 23 failure donors**. Of the selected clips, 69 reproduce the intended failure mechanism and six exhibit another observed failure, which determines their assigned label.

The resulting clips expand the available examples of pickup misses, pickup slips, and handover drops. Source–generation comparisons examine how the host scene is preserved while the transferred motion produces different object trajectories and interactions with surrounding surfaces.

Figure 2 illustrates a donor–host transfer in the industrial task. The donor shows a reel falling outside the bin, while the host maintains its grasp during transport. The generated continuation preserves the host's blue reel and scene arrangement while depicting a drop onto the surrounding table, transferring the failure behavior to a different object and initial configuration.

Table 1: Industrial failure recognition on the shared real evaluation subset. Events are detected failure intervals out of 36; clean alarms are clean episodes containing an alert out of 41. Real-only reports the stronger of the evaluated real-only configurations. The full configuration comparison is provided in the appendix.

Training condition	Events	Avg Precision	Clean alarms
No fine-tuning	25/36	0.0412	41/41
Real-only	28/36	0.4085	5/41
Real + FailureGen	**33/36**	**0.5119**	**0/41**

Downstream evaluation. We evaluate Cosmos3-Edge-based failure recognition under three conditions: no task-specific fine-tuning, real-only fine-tuning, and fine-tuning with FailureGen augmentation. Both fine-tuning conditions use the same 66 real training episodes; the augmented condition adds the 75 generated failure clips. Inputs contain video without robot telemetry, and the prediction target is active failure during execution.

Checkpoints and decision thresholds are selected using real validation recordings. Evaluation uses a separate real recording session containing 74 eligible episodes: 41 clean episodes and 33 episodes containing 36 annotated failure events. An event is detected if an alert falls within its annotated interval. We also measure average precision and the number of clean episodes containing an alert. Model implementation, training budgets, and evaluation-label review are detailed in Appendix B.

Results. The model trained with FailureGen augmentation detects **33 of 36 failure events (91.67%)**, compared with 28 of 36 (77.78%) for the strongest real-only configuration and 25 of 36 (69.44%) without fine-tuning (Table 1). This corresponds to a **13.9-percentage-point increase in event recall** over the strongest real-only baseline.

The augmented model detects all 12 pickup-slip events and all ten handover-drop events, together with 11 of 14 pickup misses. It achieves the highest average precision, 0.51, and raises **no alarms in the 41 clean episodes**. The augmented configuration therefore provides greater failure-event coverage while maintaining reliable behavior on clean executions.

4.2 Focused ViFailback evaluation

Tasks and setup. Following an initial task-level screening, we conduct a dedicated follow-up on three ViFailback tasks [22]: placing an object into a black box, placing a vial and test tube into rack slots, and opening a drawer, inserting a can, and closing the drawer. These tasks cover object placement, precise rack placement, and articulated-object manipulation. Additional conditioning and evaluation details are provided in Appendix C.

The available host recordings vary in object identity, arrangement, tabletop appearance, clutter, and background configuration. FailureGen uses joint-motion and gripper-state conditioning to produce additional examples within these recorded scene contexts. The generation mixture and source-selection details are provided in the appendix.

For downstream evaluation, Qwen3-VL-2B and Qwen3-VL-8B are fine-tuned on the focused three-task dataset. For each model, the initial condition uses real training recordings, while the augmented condition uses the same real recordings together with task-matched FailureGen examples. Both conditions are evaluated on the same real evaluation episodes, which are excluded from training and donor–host selection.

We report episode-level accuracy and failure recall. Accuracy measures correct success/failure classification across all evaluation episodes; failure recall measures the fraction of failure episodes correctly identified.

Figure 3 illustrates transfer of premature-release behavior from a can to a bottle while preserving the host scene. An additional drawer-placement example appears in Appendix C (Fig. 4).

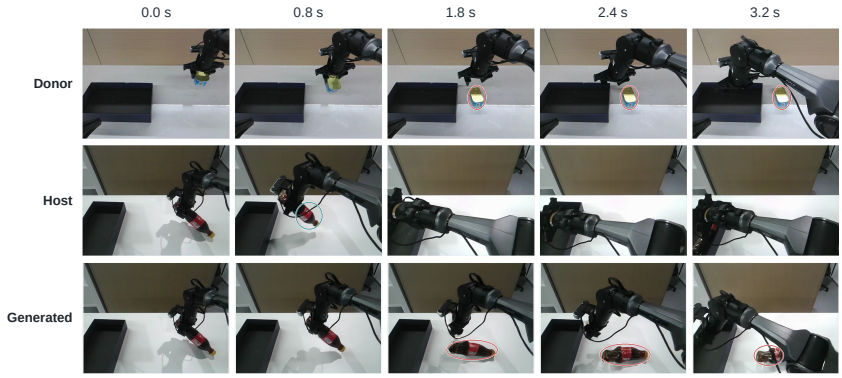

Figure 3: **Failure transfer for object placement.** Top: the donor releases a can outside the box. Middle: the host carries a bottle toward the box. Bottom: the generated continuation preserves the host's bottle and scene while depicting premature release onto the table. Times are relative to each clip; red outlines highlight the released object.

Table 2: Failure detection on the three-task ViFailback subset. Values show real-only → FailureGen-augmented performance (%). Bold values indicate improvements.

	Qwen3-VL-2B		Qwen3-VL-8B	
Task	Accuracy	Failure recall	Accuracy	Failure recall
Object into black box	94.1 → **100.0**	93.3 → **100.0**	98.5 → **100.0**	98.3 → **100.0**
Vial/test-tube rack	58.5 → 58.5	53.3 → 53.3	49.6 → **54.1**	43.3 → **48.3**
Drawer/can	84.4 → **97.0**	82.5 → **96.7**	94.8 → **97.0**	94.2 → **96.7**

Results. FailureGen augmentation improves both accuracy and failure recall in **five of the six model–task combinations**, with the remaining combination unchanged (Table 2).

The largest improvement occurs in drawer manipulation. Qwen3-VL-2B accuracy increases from **84.4% to 97.0%**, while failure recall increases from **82.5% to 96.7%**. Qwen3-VL-8B also improves, reaching 97.0% accuracy and 96.7% failure recall.

For object-into-black-box placement, both models reach **100% accuracy and failure recall** with augmentation. Qwen3-VL-2B improves from 94.1% accuracy and 93.3% recall, while Qwen3-VL-8B improves from 98.5% accuracy and 98.3% recall.

For vial/test-tube rack placement, Qwen3-VL-8B improves from 49.6% to 54.1% accuracy and from 43.3% to 48.3% failure recall. Qwen3-VL-2B remains unchanged. These results extend the observed utility of FailureGen-augmented training to additional manipulation tasks and two downstream model sizes.

5 Conclusion

We presented FailureGen, a method for expanding long-tail robot failure data through motion-conditioned world-model generation. By transferring failure-associated motion into compatible host contexts, FailureGen produces additional video–motion examples with varied visual contexts and interactions, annotated according to their observed outcomes.

Experiments on industrial bimanual manipulation and a focused ViFailback subset show that these examples can support improved failure recognition on real recordings. Together, the results support FailureGen as a way to extend the learning value of limited failure demonstrations and motivate its evaluation in broader downstream robotics tasks.

References

[1] NVIDIA. Cosmos 3: Omnimodal world models for physical AI. Technical report, NVIDIA, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/.

[2] J. Duan, W. Pumacay, N. Kumar, Y. R. Wang, S. Tian, W. Yuan, R. Krishna, D. Fox, A. Mandlekar, and Y. Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/70a06501001e1820fd1eb9ee821302d2-Abstract-Conference.html.

[3] D. Li, J. Lei, H. Wang, L. Liu, Y. Yang, Z. Wang, B. Liu, M. Zheng, and Z. Fan. Learning actionable manipulation recovery via counterfactual failure synthesis. Unpublished manuscript; under review, 2026. URL https://dream2fix.github.io/.

[4] Z. Lin, J. Duan, H. Fang, D. Fox, R. Krishna, C. Tan, and B. Wen. FailSafe: Reasoning and recovery from failures in vision-language-action models. In *2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. URL https://2026.ieee-iros.org/program/contributed_talks/#a53e3b:725.

[5] Y. Fang, W. Dong, J. Liu, Y. Yang, M. Huo, Y. Mu, H. Yao, L. E. Li, D. Szafir, and M. Ding. DenseReward: Dense reward learning via failure synthesis for robotic manipulation. Unpublished manuscript; under review, 2026. URL https://dense-reward.github.io/.

[6] P. Pacaud, R. Garcia Pinel, S. Chen, and C. Schmid. Guardian: Detecting robotic planning and execution errors with vision-language models. In *Workshop on Making Sense of Data in Robotics: Composition, Curation, and Interpretability at Scale at CoRL 2025*, 2025. URL https://openreview.net/forum?id=wps46mtC9B.

[7] Y. Guo, L. X. Shi, J. Chen, and C. Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0ae94013da7cd459402fd77874e09ee3-Abstract-Conference.html.

[8] NVIDIA, N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, et al. Cosmos world foundation model platform for physical AI. Technical report, NVIDIA, 2025. URL https://research.nvidia.com/publication/2025-01_cosmos-world-foundation-model-platform-physical-ai.

[9] J. Ye, R. Xue, B. Van Hoorick, R. Li, H. Rajaprakash, P. Tokmakov, M. Z. Irshad, V. Guizilini, and Y. Wang. RoboDream: Compositional world models for scalable robot data synthesis. In *2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. URL https://harshithabr.github.io/publication/robodream/. Accepted at IROS 2026.

[10] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin, L. Magne, A. Mandlekar, A. Narayan, Y. L. Tan, G. Wang, J. Wang, Q. Wang, Y. Xu, X. Zeng, K. Zheng, R. Zheng, M.-Y. Liu, L. Zettlemoyer, D. Fox, J. Kautz, S. Reed, Y. Zhu, and L. Fan. DreamGen: Unlocking generalization in robot learning through video world models. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 5170–5194. PMLR, 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

[11] Y. Guo, T. Lee, L. X. Shi, J. Chen, P. Liang, and C. Finn. VLAW: Iterative co-improvement of vision-language-action policy and world model. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 37998–38011. PMLR, 2026. URL https://proceedings.mlr.press/v306/guo26i.html.

[12] Y. Xue, L. Xu, Z. Liu, Z. Wu, Z. Gu, X. Song, B. Jia, and Z. Wang. WorldSample: Closed-loop real-robot RL with world modelling. Unpublished manuscript, 2026. URL https://xxreinsno.github.io/worldsample/.

[13] J. Seo, S. Veer, R. Tian, W. Ding, A. Sharma, K. Leung, E. Schmerling, M. Pavone, and A. Bajcsy. StressDream: Steering video world models for robust policy evaluation and improvement. In *Conference on Robot Learning (CoRL)*, 2026. URL https://stressdream.github.io/. Accepted; proceedings forthcoming.

[14] T. Yang, Z. Shen, Z. Mi, Z. Zhang, J. Zhou, J. Ji, J. Dai, J. Chen, B. Chen, and Y. Yang. MiraBench: Evaluating action-conditioned reliability in robotic world models. Unpublished manuscript, 2026. URL https://arxiv.org/abs/2605.29360.

[15] P. Co, S. Hu, C. Jiao, H. Cheng, Y. Luo, Y. Xu, S. Chen, Z. Zhao, Z. Wang, D. Chi, P. Liu, Y. Chen, H. Liu, Z. Yuan, H. Jia, Y. Zhuang, T. Zhang, L. Lin, H. Tan, and S. Zhang. WorldSimProbe: Diagnosing simulator faithfulness in action-conditioned world models for embodied manipulation. Unpublished manuscript, 2026. URL https://evophys.com/WorldSimProbe/.

[16] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1820–1864. PMLR, 2023. URL https://proceedings.mlr.press/v229/mandlekar23a.html.

[17] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu. DemoGen: Synthetic demonstration generation for data-efficient visuomotor policy learning. In *Proceedings of Robotics: Science and Systems*, 2025. doi:10.15607/RSS.2025.XXI.157. URL https://www.roboticsproceedings.org/rss21/p157.html.

[18] J. Yu, L. Fu, H. Huang, K. El-Refai, R. A. Ambrus, R. Cheng, M. Z. Irshad, and K. Goldberg. Real2Render2Real: Scaling robot data without dynamics simulation or robot hardware. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 547–577. PMLR, 2025. URL https://proceedings.mlr.press/v305/yu25a.html.

[19] Z. Chen, S. Kiami, A. Gupta, and V. Kumar. GenAug: Retargeting behaviors to unseen situations via generative augmentation. In *Proceedings of Robotics: Science and Systems*, 2023. doi:10.15607/RSS.2023.XIX.010. URL https://www.roboticsproceedings.org/rss19/p010.html.

[20] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, D. M, J. Peralta, B. Ichter, K. Hausman, and F. Xia. Scaling robot learning with semantic data augmentation through diffusion models. In *Proceedings of Robotics: Science and Systems*, 2023. doi:10.15607/RSS.2023.XIX.027. URL https://www.roboticsproceedings.org/rss19/p027.html.

[21] C. Grislain, H. Rahimi, O. Sigaud, and M. Chetouani. I-FailSense: Towards general robotic failure detection with vision-language models. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_5.html#thi2i_75.

[22] X. Zeng, X. Zhou, Y. Li, J. Shi, T. Li, L. Chen, L. Ren, and Y.-L. Li. Diagnose, correct, and learn from manipulation failures via visual symbols. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 42386–42395, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zeng_Diagnose_Correct_and_Learn_from_Manipulation_Failures_via_Visual_Symbols_CVPR_2026_paper.html.

A Implementation Details

World-model adaptation. We adapt Cosmos3-Nano through forward- and inverse-dynamics supervision. Fine-tuning updates the generation expert, time embedding, video and action projections, and action-modality embedding; the remaining parameters are frozen. The corresponding implementation modules are `moe_gen`, `time_embedder`, `vae2llm`, `llm2vae`, `action2llm`, `llm2action`, and `action_modality_embed`.

Generation inputs. Both studies use windows containing 17 observations at 5 Hz and 16 motion transitions, spanning 3.2 s between the first and last observations. Motion is derived from measured robot states at the corresponding sampling instants. Training-derived normalization is reused during generation. Neutral task descriptions provide context without specifying a future failure.

Motion variation. Donor motion is aligned with the selected host segment, with transitions connecting the intervention to the host motion. For ViFailback, variation scales joint-motion excursions around the host anchor by factors $\{0.95, 0.975, 1.025, 1.05\}$ while preserving gripper values and timing. Each generated example retains its donor and host identifiers, conditioning sequence, transfer settings, and inference seed.

B Industrial Experiment Details

Data and generator. The 80 source recordings are partitioned into 66 generator-training, seven validation, and seven test episodes. The training partition contains 37 clean executions, ten pickup misses, eleven pickup slips, and eight handover drops. Downstream detectors use the same 66 training episodes and combine the remaining 14 recordings for validation. All synthetic donors and hosts come from training.

Generator observations are subsampled from 30 to 5 Hz and assembled into a 960×960 canvas, with the overhead view above the two wrist views. The 20-dimensional conditioning contains relative end-effector translation, six-dimensional rotation, and absolute gripper state for each arm. Poses are obtained through forward kinematics, with relative motion computed as $T_t^{-1} T_{t+1}$. Normalization uses training-only first and 99th percentiles. Equal-probability forward- and inverse-dynamics training uses batch size 16 for 1,243 updates.

Synthetic supervision. The generation cohort pairs each of 27 failure donors with two hosts and renders each pairing with seeds 0 and 1. The 75 selected failure clips contain 18 pickup misses, 27 pickup slips, and 30 handover drops. They contribute 128 positive decision targets. Only active-failure timestamps in synthetic clips receive supervision; other frames remain available as temporal context.

Detector implementation. Fine-tuned detectors initialize independently from Cosmos3-Edge revision `2a00e87e` and train its visual tower, projector, and causal binary prediction heads. Each prediction receives seven RGB observations at 5 Hz, spanning 1.2 s, on a 640×320 three-view canvas. The target indicates active failure at the final timestamp; unknown labels are excluded. The pretrained reference retains the language decoder and scores *yes*/*no* responses under a fixed prompt.

Training and model selection. Table 3 summarizes the fine-tuned configurations. The stronger real-only baseline in the main paper uses a 5:1 negative-to-positive target ratio and positive-loss weight 5.0. Each run is configured for five epochs; validation average precision selects the epoch-4 checkpoints. Decision thresholds maximize validation frame F1 and remain fixed during evaluation. The configurations share their real source episodes but differ in sampled negatives and optimization budgets.

Evaluation protocol. Testing uses a separate real recording session. The eligible subset excludes ten clean recordings with an arm outside the field of view and one transport-drop recording outside

Table 3: Industrial fine-tuning configurations. Update counts describe the configured five-epoch runs. Real 1:1 and Real 5:1 denote negative-to-positive sampling ratios; Aug. denotes real plus FailureGen data.

Quantity	Real 1:1	Real 5:1	Aug.
Real positive targets	105	105	105
Synthetic positive targets	0	0	128
Real negative targets	105	525	233
Configured updates	135	400	295
Events detected	21/36	28/36	33/36
Average precision	0.3090	0.4085	0.5119
Clean episodes with alarms	4/41	5/41	0/41

the failure taxonomy. Real evaluation annotations received partial human review; label corrections and exclusions followed inspection of false positives. All systems are scored on the same final subset with fixed weights and validation thresholds.

An event is detected when an alert falls within its labeled interval. Average precision is computed over decision timestamps, and a clean-episode alarm occurs when any eligible timestamp in a clean episode exceeds the threshold. These definitions distinguish event coverage from frame-level classification.

C ViFailback Experiment Details

Tasks and conditioning. The focused evaluation covers object-into-black-box placement (task 49), vial/test-tube rack placement (task 55), and drawer/can manipulation (task 80). Conditioning contains twelve measured joint-motion channels and two absolute gripper-position channels. Overhead and wrist observations are subsampled from 25 to 5 Hz. Generation uses the adapted Cosmos3-Nano exponential-moving-average checkpoint at update 1,500, selected using validation replay fidelity.

Host variation and generation. Hosts provide variation in object identity for black-box placement, tabletop appearance and setup for rack placement, and background objects and cabinet pose for drawer manipulation. Generated examples include donor-motion transfers, host-motion controls, and original-context failure replays, with labels assigned from their observed outcomes. Repeated inference seeds provide additional renders of a source pairing rather than additional donor–host contexts.

Downstream comparison. Qwen3-VL-2B and Qwen3-VL-8B are fine-tuned on the three selected tasks. Initial models use real training recordings; augmented models use the same real recordings with task-matched synthetic examples. Both conditions are evaluated on the same real recordings, excluded from training and donor–host selection.

Accuracy measures correct episode-level success/failure classification. Failure recall measures the fraction of failure episodes identified correctly. These episode-level metrics are reported separately from the industrial study's active-event detection metrics.

Additional qualitative example. Figure 4 shows failure transfer in drawer placement. The host places the can inside the drawer, while the generated continuation depicts the can falling onto the adjacent tabletop.

Cleanup removes unsuitable generations. Across the accumulated experiments, removal rates are 9.8% for industrial manipulation and 11.2% for ViFailback, leaving 90.2% and 88.8%, respectively, usable. These usability rates are separate from selecting clips that contain a particular failure: a coherent successful continuation can remain usable without becoming a positive failure-training example.

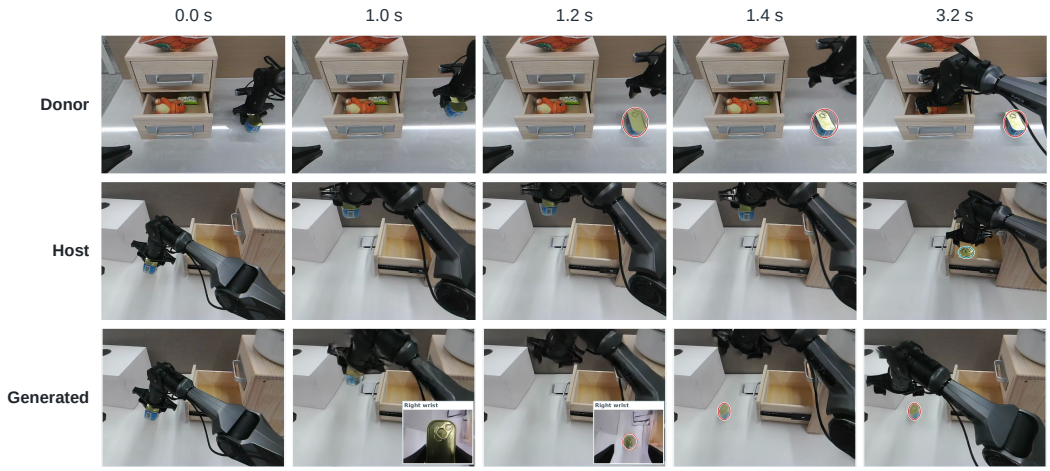

Figure 4: **Failure transfer for drawer placement.** Rows show the real donor, real host, and generated continuation. The generated can is released outside the drawer while the host scene is retained. Wrist-camera insets show the release obscured in the main view. Times are relative to each clip; red outlines highlight the released can.

D Agentic temporal annotation

The labeling pipeline coordinates vision-capable LLM workers through evidence preparation, label proposal, and review. Given episode videos and a task-specific label vocabulary, it prepares chronological frame grids with synchronized camera views and source-frame indices. Workers inspect overlapping windows to propose task phases, failure events, and recovery intervals. Independent review passes check these proposals and look for missed events.

Candidate events and temporal boundaries receive closer inspection using denser frame sequences. Disagreements trigger additional visual adjudication, while unresolved cases are flagged for human review. When available, robot telemetry can nominate intervals for inspection; event labels are determined from visual evidence.

Structural validation checks label categories, timestamp bounds, interval overlaps, and required timeline coverage. The output records event categories, temporal boundaries, source references, and review status. A review interface lets human annotators inspect the videos and correct labels or boundaries. Episode processing runs in parallel, with intermediate results retained for resuming interrupted annotation.