

PREFAIL:

Identifying Precursors to Failures in Robotic Lift-and-Place Tasks to Improve Task Execution Performance

2026.10.01

Zeyu Shangguan, Rajas Chitale, Rutvik Patel, Satyandra K. Gupta, Daniel Seita



Proactive failure prediction before intervention becomes too late

Overview



- 01 **Author & Paper**
- 02 **Introduction**
- 03 **Background : FINO-Net**
- 04 **PREFAIL**
- 05 **Conclusion**



Zeyu Shangguan

University of Southern California

alumni.usc.edu의 이메일 확인됨 - 홈페이지

Artificial Intelligence Robotics

팔로우

내 프로필 만들기

제목

인용

연도

Cross-Domain Few-Shot Object Detection via Enhanced Open-Set Object Detector

Y Fu, Y Wang, Y Pan, L Huai, X Qiu, Z Shangguan, T Liu, L Kong, Y Fu, ...

In European Conference on Computer Vision, 247-264

107

2024

Test-time linear out-of-distribution detection

K Fan, T Liu, X Qiu, Y Wang, L Huai, Z Shangguan, S Gou, F Liu, Y Fu, ...

2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR ...

37

2024

Identification of novel classes for improving few-shot object detection

Z Shangguan, M Rostami

2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW ...

37

2023

Manipbench: Benchmarking vision-language models for low-level robot manipulation

E Zhao, V Raval, H Zhang, J Mao, Z Shangguan, S Nikolaidis, Y Wang, ...

arXiv preprint arXiv:2505.09698

35

2025

Improved region proposal network for enhanced few-shot object detection

Z Shangguan, M Rostami

arXiv preprint arXiv:2308.07535

23

2023

Cross-domain multi-modal few-shot object detection via rich text

Z Shangguan, D Seita, M Rostami

2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV ...

19 *

2025

Sequential multi-object grasping with one dexterous hand

S He, Z Shangguan, K Wang, Y Gu, Y Fu, Y Fu, D Seita

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems ...

15

2025

PREFAIL: Identifying Precursors to Failures in Robotic Lift-and-Place Tasks to Improve Task Execution Performance

Z Shangguan, R Chitale, R Patel, SK Gupta, D Seita

arXiv preprint arXiv:2607.16921

1 *

2026

인용

전체

2021년 이후

서지정보

342

340

h-index

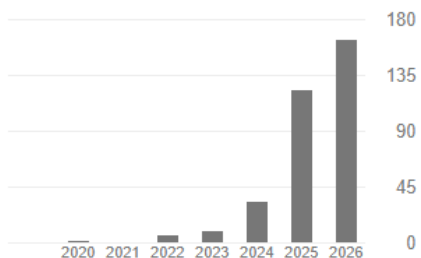
10

10

i10-index

10

10



공개 액세스

모두 보기

자료 0개

자료 1개

비공개

공개

재정 지원 요구사항 기준

Introduction



Why speed and reliability conflict

- Non-prehensile lift-and-place: object를 carrier 위에 접촉력과 마찰로 지지한다.
- 빠른 방향 전환
-> 관성력이 마찰을 이겨 object shift / drop이 발생
- 항상 낮은 속도
-> failure risk는 줄지만 cycle time이 증가
- throughput을 유지하면서 manipulation reliability를 확보하는 것 중요



Core trade-off **Speed / Cycle Time ↔ Reliability / Failure Risk**

Fast manipulation changes the failure-prediction problem itself.

- | | | |
|----|----------------------------|--|
| 01 | Scarce failure data | Dynamic high-speed failure cases are rare, making failure-oriented training data difficult to collect. |
| 02 | OOD ambiguity | Existing OOD methods can struggle to separate failure and non-failure trajectories during rapid manipulation. |
| 03 | Policy dependence | Approaches that predict future actions are often coupled to a specific learned policy, limiting cross-policy generalization. |

This motivates failure prediction that reacts directly to scene dynamics rather than relying on a fixed policy structure.

Background : FINO-Net



Reactive execution monitoring: detect and identify unexpected outcomes from sensory evidence.



FINO-Net

RGB + Depth + Audio로 manipulation / post-manipulation failure를
end-to-end로 감지하고 유형까지 분류

Raw multimodal sensory sequences → learned representation → failure detection / identification

Failure labels

Collision · Miss · Overflow · Spill · Overturn (+ Success)

Manipulation actions

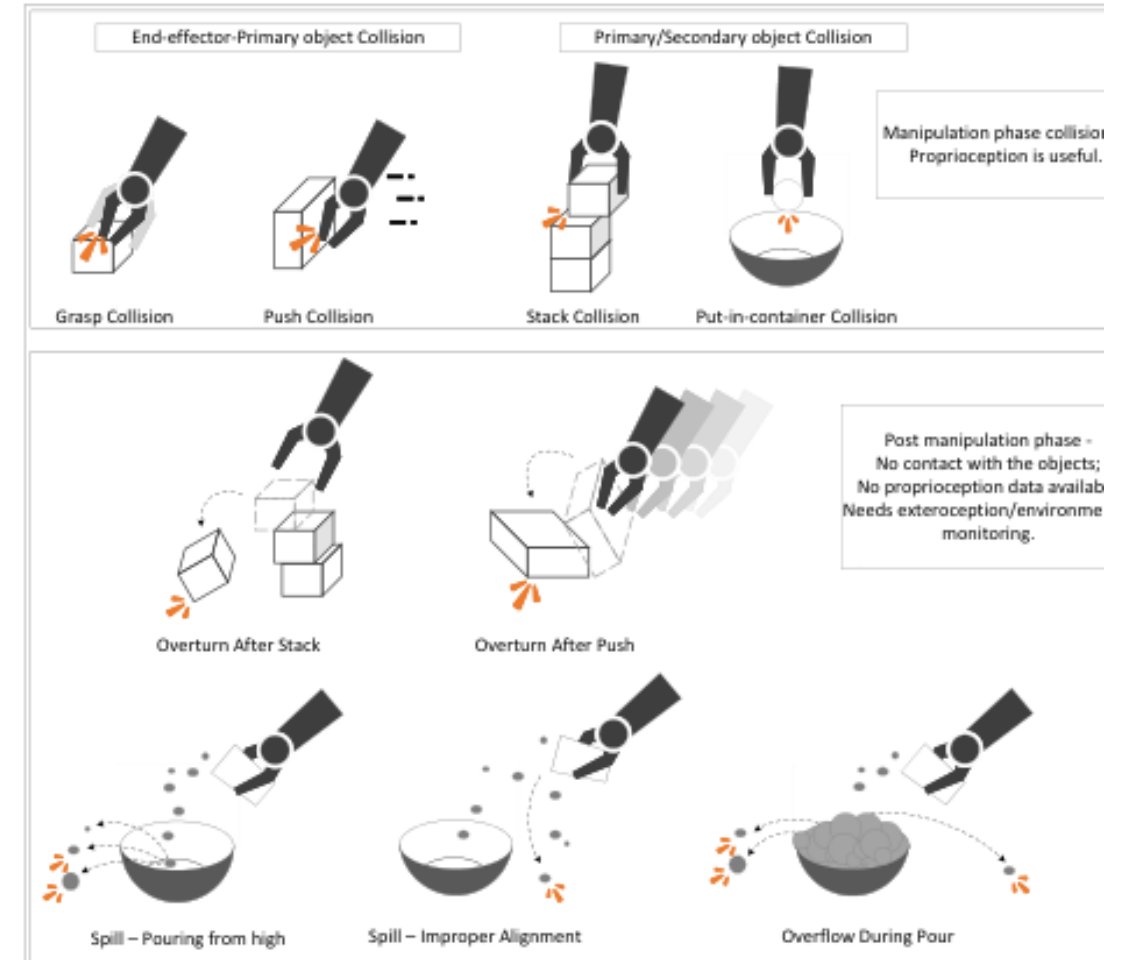
Push · Pick&Place · Pour · Put-in-container · Stack-a-tower

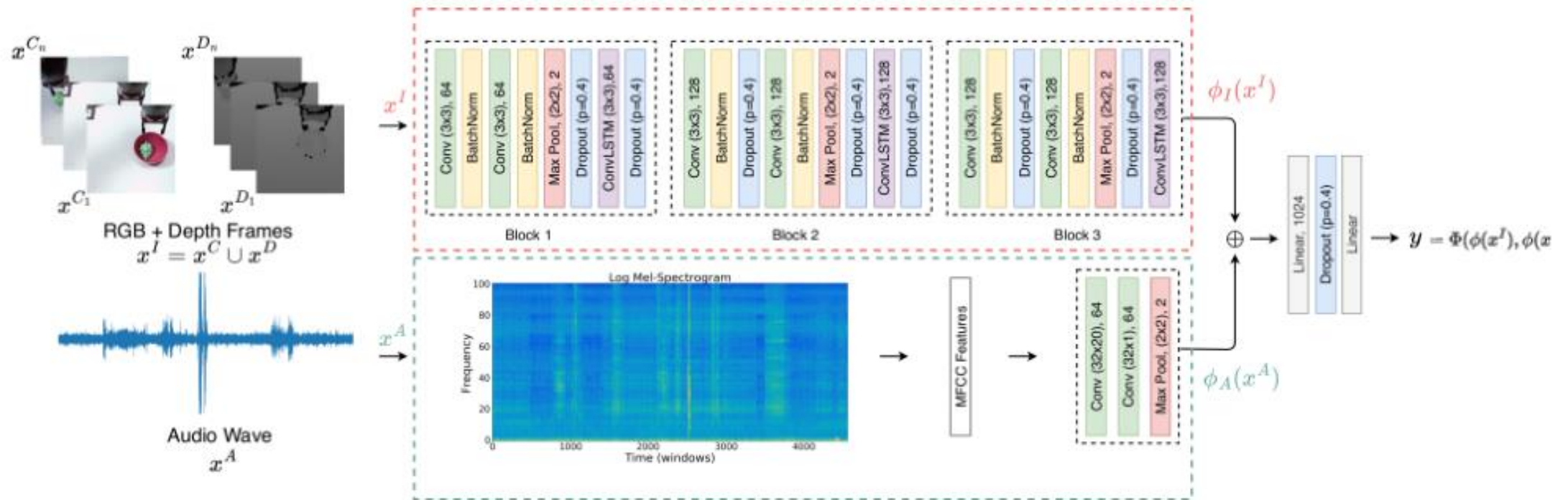
Sensors

RGB · Depth · Audio

Exteroceptive monitoring also covers post-manipulation states

Dataset extension: +99 new multimodal manipulation recordings





RGB + Depth

early fusion

Conv + ConvLSTM

spatio-temporal visual feature

Vision + Audio

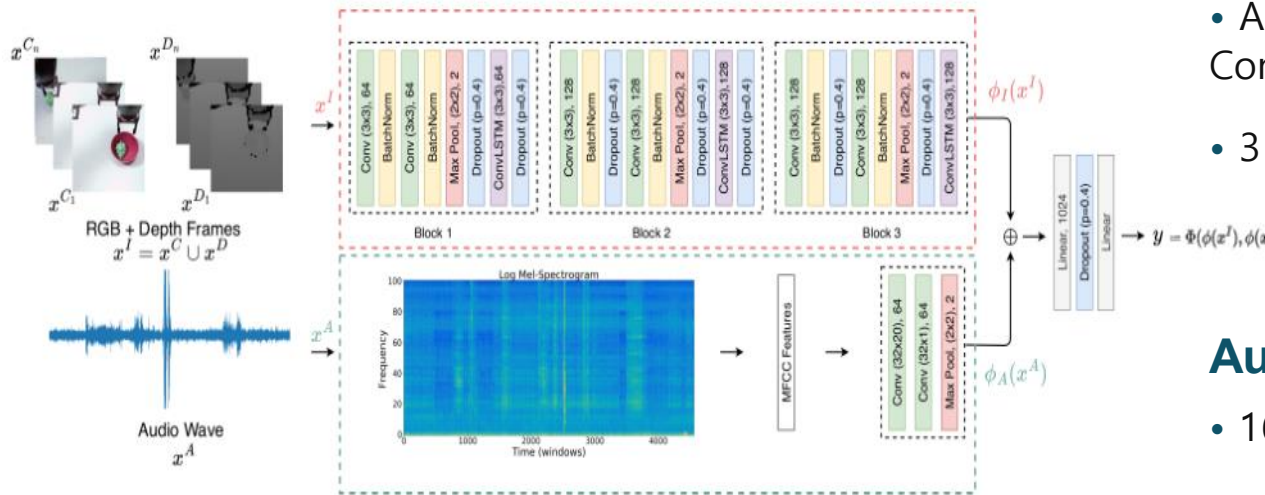
late fusion → classifier

Vision branch

- Depth thresholding으로 self-occluded frame 제거
- 전체 execution에서 RGB-D pair을 sampling
- Approach / Manipulate / Retreat temporal pattern을 ConvLSTM으로 표현
- 3 visual blocks: convolution + ConvLSTM + pooling

Audio branch

- 16 kHz audio → 32 ms window → STFT / Mel filterbank
- 20 MFCC features를 CNN으로 처리
- Final visual / auditory features를 concatenate 후 FC layers에서 예측



Detection: success vs failure | Identification: success + 5 failure classes

TABLE III: Quantitative Results

	Detection			Classification		
	Pr	Re	F1	Pr	Re	F1
FINO-Net-RGB	0.7667	0.7567	0.7617	0.6924	0.6216	0.6551
FINO-Net-D	0.6898	0.6756	0.6826	0.7302	0.6486	0.6869
FINO-Net-RGB-D	0.7817	0.7567	0.7690	0.6846	0.6756	0.6801
FINO-Net-A	0.6801	0.6486	0.6640	0.7475	0.6216	0.6788
FINO-Net-RGB-D-A	0.8665	0.8648	0.8656	0.8085	0.7837	0.7959

RGB-D-A Detection F1 = 0.8656

Classification F1 = 0.7959

Visual + auditory cues are complementary.

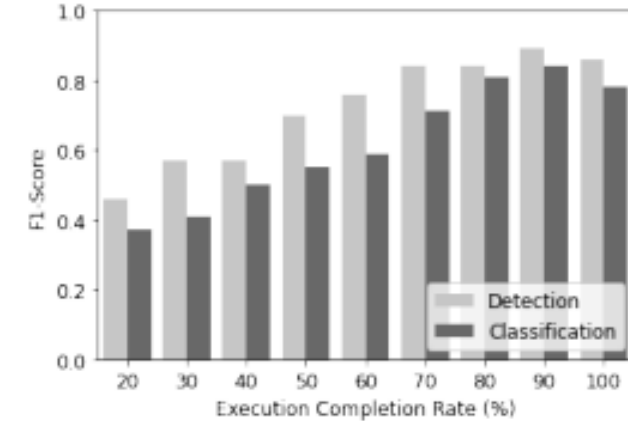


Fig. 6: The analysis on real-time demands for failure detection and classification. Different rates of the observation sequence of the execution are provided to FINO-Net-RGB-D-A model

Real-time analysis

Failure indicators are observed mainly in the second half of execution; predictions become more reliable as more of the trajectory is observed.

FINO-Net observation

more reliable predictions towards the end of manipulation



When detection becomes reliable, is there still enough time to intervene?

FINO-Net detects and identifies failures, but it does not define the latest point at which an intervention can still prevent failure. Its own future work explicitly points to anticipation before failure occurs.

PREFAIL



Task

6-DOF robot transports an object on a flat carrier at maximum joint velocity.

Failure

Object가 robot motion 중 carrier에서 떨어지면 failure.
Carrier 위에서 reorientation만 발생하면 success.

Input window

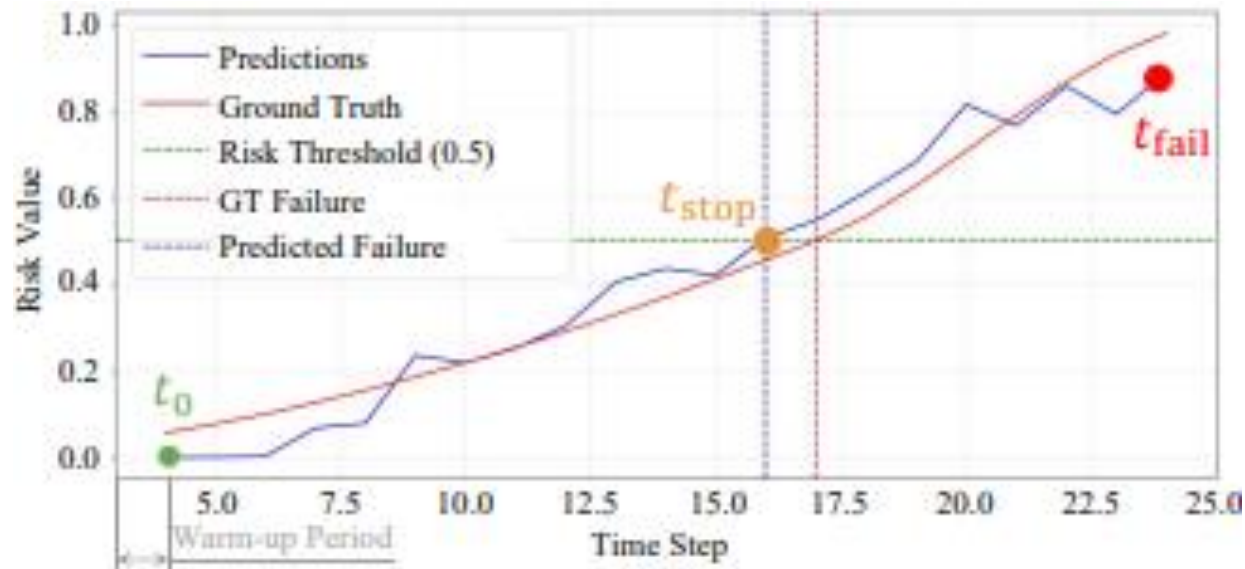
최근 $m = 5$ time steps의 multi-view image + robot state

Output

각 time step마다 continuous risk $R_i \in [0,1]$



Goal: formulate proactive prediction as per-step continuous risk estimation.



t_{stop}

Emergency stop을 적용했을 때 failure를 막을 수 있는 마지막 time step

t_{fail}

실제로 object가 carrier에서 떨어져 failure가 발생하는 시점

Interpretation

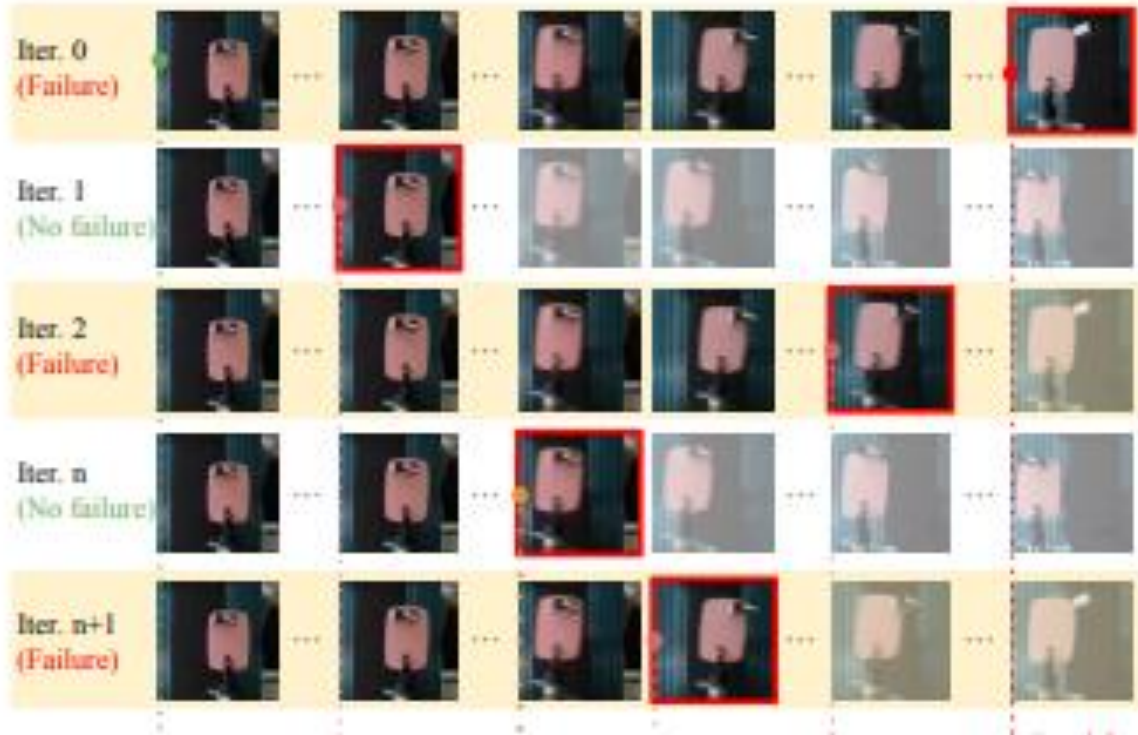
t_{stop} is the decision deadline: after this point, even an emergency stop can no longer prevent failure.

Prediction is actionable only if the boundary is crossed before t_{stop} .

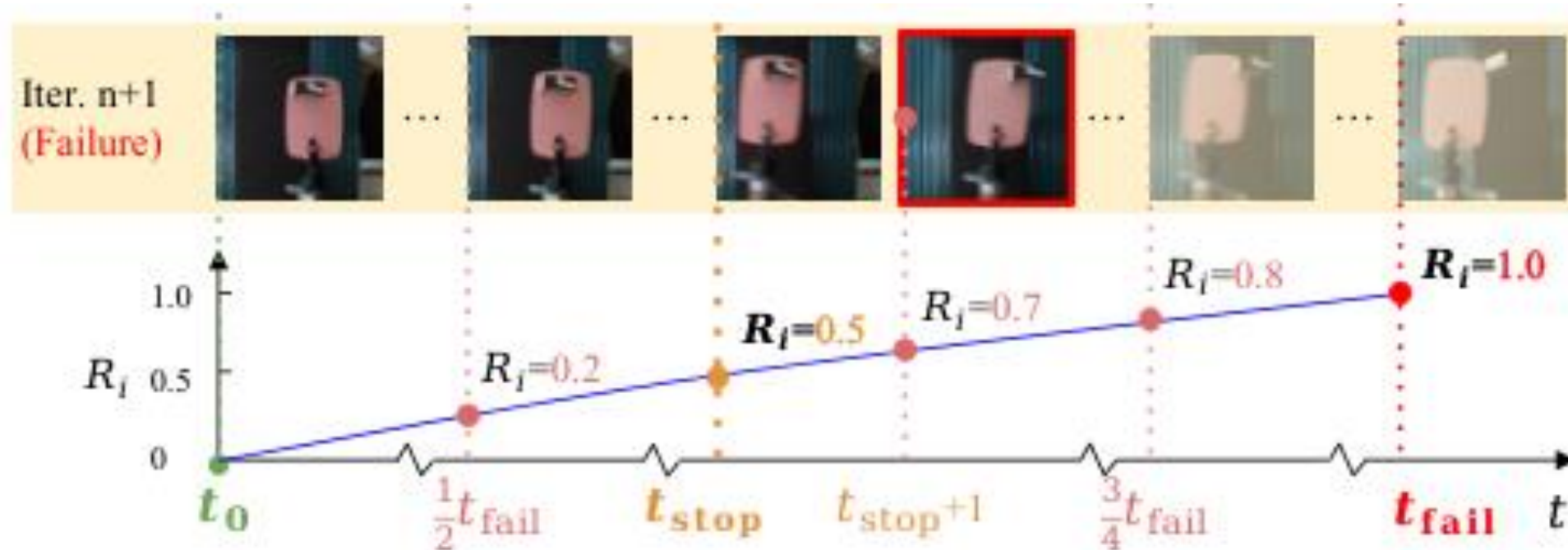
Binary search over intervention time

- 1 실패 trajectory에서 실제 failure 시점 t_{fail} 을 먼저 확인
- 2 동일 trajectory를 replay하고 search interval의 중간 시점에 e-stop 적용 (binary search)
- 3 그래도 실패하면 → 더 이른 구간 탐색
- 4 실패하지 않으면 → 더 늦은 구간 탐색
- 5 경계가 수렴하면 latest safe intervention time을 t_{stop} 으로 정의

Simulation: counterfactual replay + binary search
Real world: repeated physical rollout / reset with e-stop



Backtracking gives the critical boundary — not a full risk curve.



Only t_0 , t_{stop} , and t_{fail} are physically grounded by the rollout.

Intermediate R_t values are **constructed supervisory labels**, not measured failure probabilities.

Next → Hermite spline interpolation

The model still needs a target risk value for every timestep.

PREFAIL uses Hermite spline interpolation between the labeled anchors.

Modules	Precision $\uparrow$	Accuracy $\uparrow$	Recall $\uparrow$	η $\uparrow$
Hermite (used)	0.995	0.997	0.995	0.923
Bézier	0.973	0.988	0.995	0.887
Linear	0.958	0.980	1.0000	0.825
Step	0.314	0.314	1.0000	0.1373

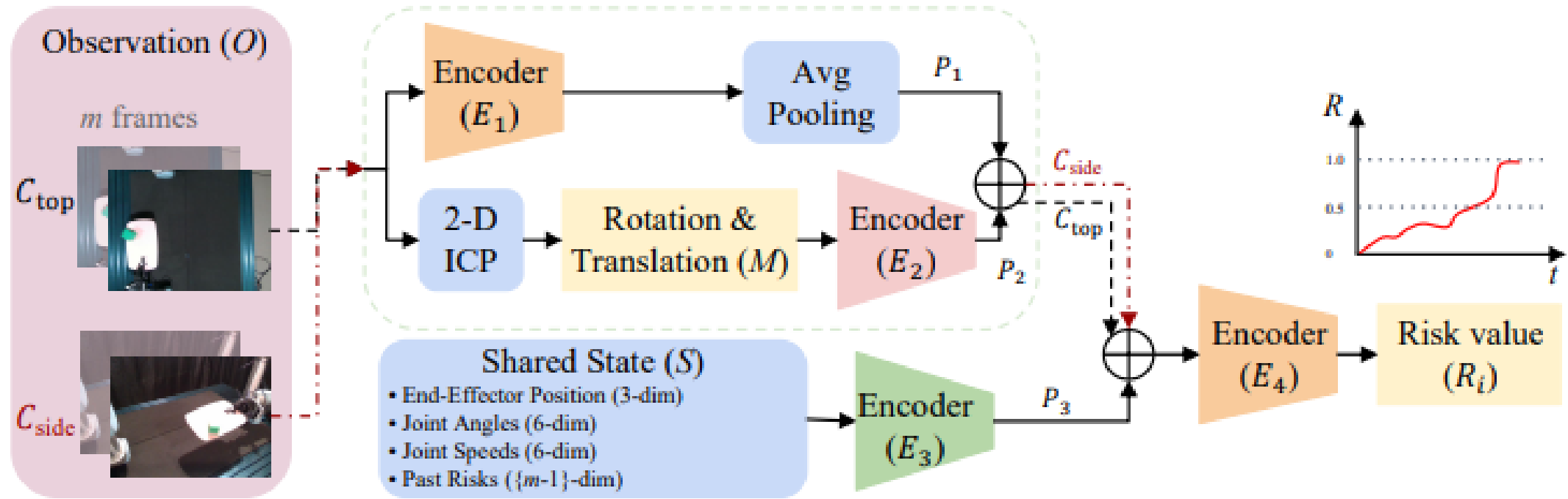
TABLE III: Comparison of different interpolation techniques.

Interpretation

- Step labels are too abrupt and provide weak temporal supervision.
- Continuous interpolation preserves the ordering of recoverability.
- Hermite gives the best overall precision / accuracy / η among compared interpolations.

Important: spline shape is a training target, not a physically measured “true” risk curve.

Classification loss anchors the t_{stop} boundary; regression loss learns the smooth temporal trend.



Observation branch

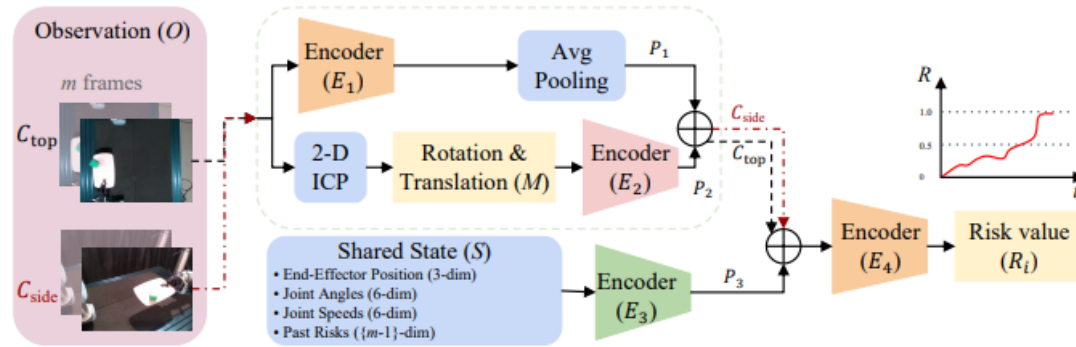
Visual P_1 + Relative motion P_2

State branch

Robot state + Past risk $\rightarrow P_3$

Fusion

$P_1 \oplus P_2 \oplus P_3 \rightarrow E_4 \rightarrow R_i$



Why multi-view?

- Object motion이 camera line-of-sight와 겹치면 한 view에서 ambiguous할 수 있다.
- Top + side view가 서로 보완적인 motion cue를 제공한다.
- 최근 $m = 5$ frame sequence를 사용해 instantaneous appearance가 아니라 temporal change를 본다.

Image encoder E1

CNN layer로 multi-view image sequence를 encode

Avg. pooling

camera / time features를 aggregate

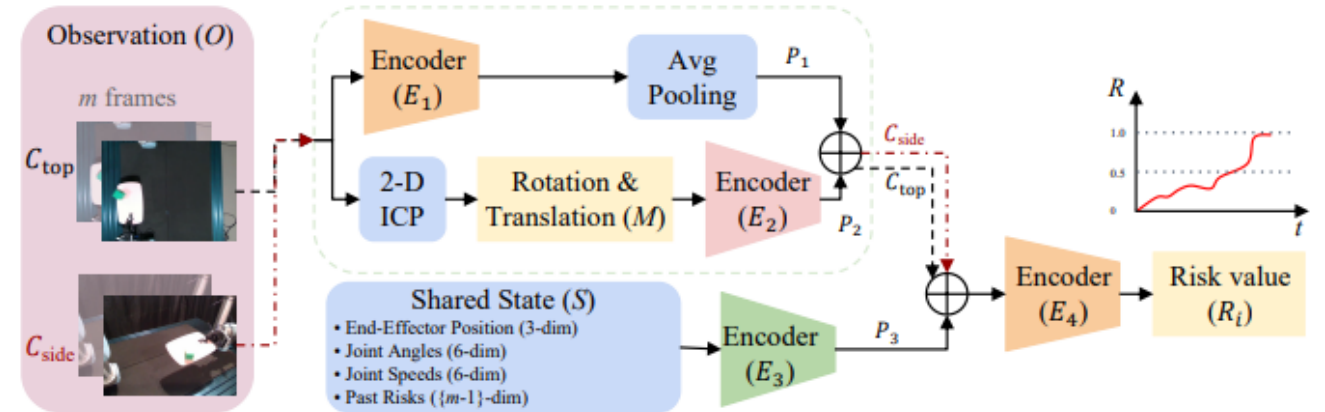
Output: visual feature P1

Key intuition

Carrier 자체가 빠르게 움직이는 것은 정상

위험 신호는 object가 carrier에 대해 미끄러지거나 회전하는 상대 운동

2-D ICP 초기 contour와 현재 contour의 rigid displacement를 추정



$$R_{o-c} = R_{o-o} - R_{c-c}$$

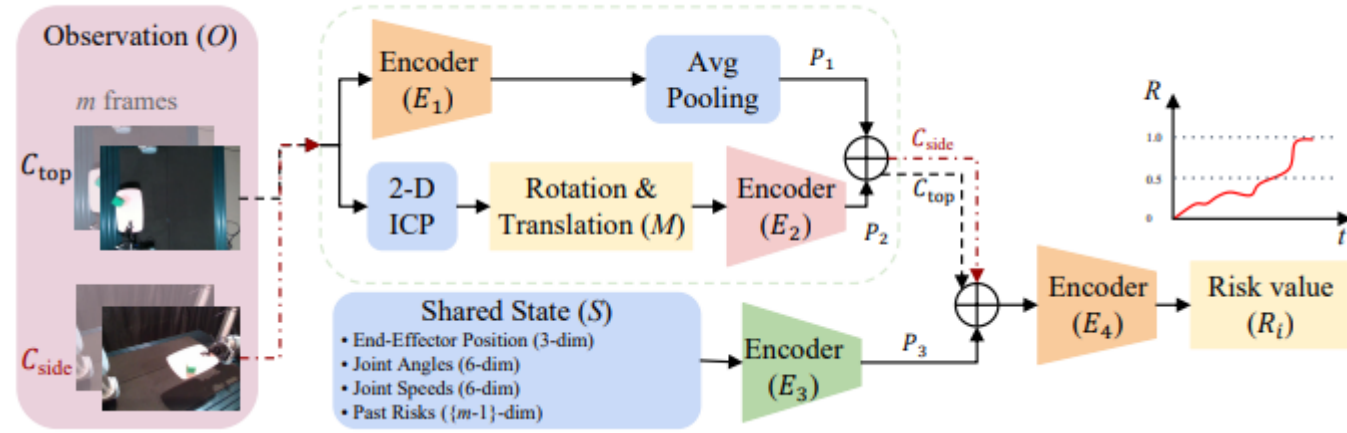
$$T_{o-c} = T_{o-o} - T_{c-c}$$

Real world: SAM2로 object / carrier contour segmentation

Simulation: projected shape contour 직접 사용

Relative motion $M \rightarrow$ MLP encoder $E2 \rightarrow$ feature $P2$

The model explicitly separates robot motion from object-carrier instability.



End-effector position

3-D position
현재 workspace 상태

Joint angles

6-D
robot configuration

Joint speeds

6-D
현재 동역학 / 빠른 motion

Past risks

$m-1$ values
risk의 temporal trend

State vector $S \rightarrow$ MLP encoder $E_3 \rightarrow P_3$

P_1, P_2, P_3 를 concatenate $\rightarrow$ ResNet encoder $E_4 \rightarrow$ continuous risk R_i

$$L_{\text{total}} = \alpha L_{\text{cls}} + (1 - \alpha) L_{\text{reg}}$$

Classification loss L_{cls}

- Binary cross entropy
- t_{stop} 이전 = low-risk / 이후 = high-risk boundary를 분리
- Emergency-stop trigger에 필요한 safety boundary를 명확히 학습

Regression loss L_{reg}

- Mean squared error
- R_{pred} 와 continuous ground-truth risk curve의 차이를 최소화
- Risk가 t_{stop} / t_{fail} 로 진행되는 temporal evolution을 학습

Why both? Classification은 경계를 선명하게, regression은 위험도의 시간적 진행을 유지한다.

Simulation

- MuJoCo + UR5 + flat tray carrier
- 539 box-shaped trajectories
- 5,236 diverse-object trajectories
- Total 5,775 trajectories

Real world

- UR5 + Robotiq 2F-85 + flat tray
- 2× Intel RealSense D435, top / side, 30 fps
- 254 trajectories, five box-shaped object types



Train / Validation / Test = 8 : 1 : 1

Standard metrics

Precision · Accuracy · Recall

FP-2 실제 failure를 맞혔더라도 t_{stop} 이후에 detection 하면 too late로 처리

Intervention timeliness ratio

$$\eta = t_{\text{stop}}^{\text{pred}} / t_{\text{stop}}$$

$\eta \rightarrow 1$: 가능한 늦게 개입하면서도 deadline 이전에 멈춤

$\rightarrow$ task progress를 최대한 유지하고 cycle time 증가를 줄임

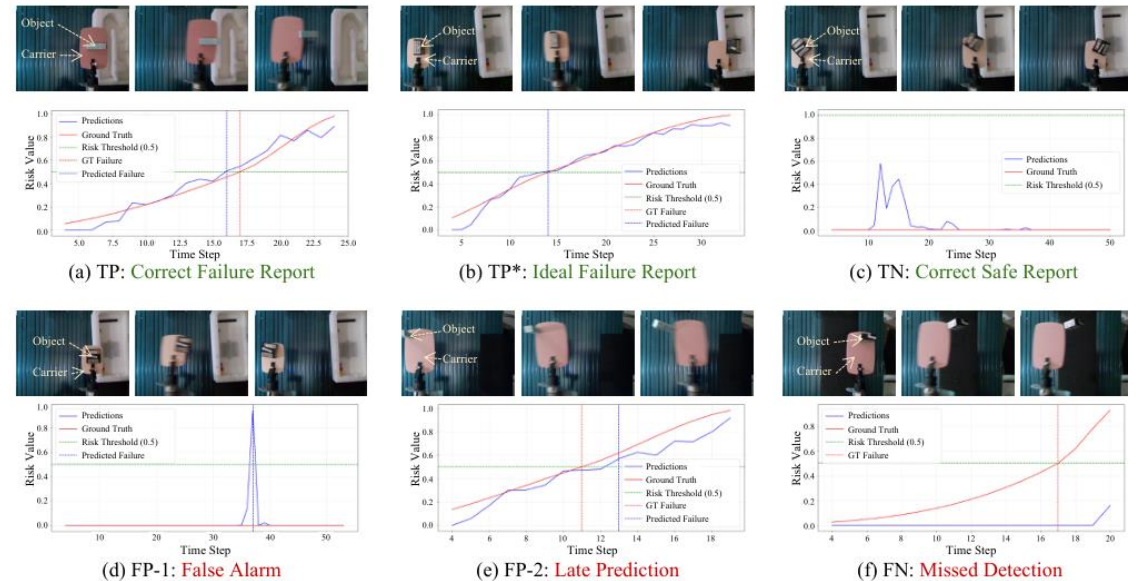


Fig. 5: Visualization of all evaluation results on real-world test cases, with the predicted risk value curve (solid red line) vs the ground truth risk value curve (solid blue line) during evaluation. The dotted red line refers to the moment of t_{stop} , while the dotted blue line is $t_{\text{stop}}^{\text{pred}}$. The threshold of risk value ($R_i = 0.5$) is indicated by dotted green line. TP* is a perfect case that demonstrates $\eta = 1.0$. See Sec. IV-C for details.

Methods / Variants		Simulation				Real-world			
		Precision $\uparrow$	Accuracy $\uparrow$	Recall $\uparrow$	η $\uparrow$	Precision $\uparrow$	Accuracy $\uparrow$	Recall $\uparrow$	η $\uparrow$
Fail-Detect [7] (ResNet18)	(CFM [28], FM)	0.389	0.389	1.000	0.179	0.478	0.478	1.000	0.297
	(CFM [28], DP)	0.386	0.386	1.000	0.202	0.500	0.500	1.000	0.303
	(NatPN [29], FM)	0.374	0.374	1.000	0.198	0.493	0.493	1.000	0.275
	(NatPN [29], DP)	0.384	0.384	1.000	0.185	0.493	0.493	1.000	0.275
	(PCA-kmeans [26], FM)	0.389	0.389	1.000	0.104	0.493	0.493	1.000	0.226
	(PCA-kmeans [26], DP)	0.388	0.388	1.000	0.082	0.489	0.489	1.000	0.221
	(RND [27], FM)	0.369	0.369	1.000	0.267	0.504	0.504	1.000	0.289
	(RND [27], DP)	0.384	0.384	1.000	0.170	0.507	0.507	1.000	0.302
	(logpZO [7], FM)	0.389	0.389	1.000	0.118	0.504	0.504	1.000	0.195
	(logpZO [7], DP)	0.389	0.389	1.000	0.083	0.496	0.496	1.000	0.219
PREFAIL (ours) (ResNet18)		0.991	0.995	0.995	0.933	0.944	0.963	0.985	0.932
PREFAIL (ours) (ResNet50)		0.967	0.986	0.995	0.964	0.916	0.948	0.984	0.910
PREFAIL (ours) (ResNet152)		0.995	0.997	0.995	0.923	0.900	0.933	0.978	0.925

TABLE I: Performance on simulation and real-world data. For simulation, the model is both trained and tested on simulation data. For real-world, the model is trained and tested on real-world data, with results reported as the average over 10-fold cross-validation across 10 experiments. The baseline method has multiple variants, each evaluated with different policy generation methods (FM for flow-matching [38], DP for diffusion policy [10]). The best results are highlighted in bold. Our PREFAIL consistently and largely outperforms all baseline variants. Although we do not achieve 1.000 recall, our high precision

Simulation — ResNet18

Precision: **0.991**
Accuracy: **0.995**
Recall: **0.995**
H: **0.933**

Real world — ResNet18

Precision: **0.944**
Accuracy: **0.963**
Recall: **0.985**
H: **0.932**

Fail-Detect baselines: recall ≈ 1.0 but low precision / accuracy $\rightarrow$ almost everything is flagged as failure.

Modules	Precision $\uparrow$	Accuracy $\uparrow$	Recall $\uparrow$	η $\uparrow$
Complete	0.995	0.997	0.995	0.923
Classification loss only	0.574	0.850	1.000	0.990
Regression loss only	0.980	0.991	0.995	0.855
Without images	0.973	0.988	0.995	0.904
Without relative motion	0.812	0.870	0.816	0.596
Without past risk	0.314	0.315	1.000	0.832
Without end-effector position	0.958	0.983	0.995	0.910
Without joint angles	0.927	0.972	1.000	0.860
Without joint speeds	0.384	0.417	0.995	0.839

Without relative motion

Precision 0.812 · Accuracy 0.870

Recall 0.816 · η 0.596

Without past risk

Precision 0.314 · Accuracy 0.315

Recall 1.000 · η 0.832

Without joint speeds

Precision 0.384 · Accuracy 0.417

Recall 0.995 · η 0.839

Relative motion and past-risk history are the most critical signals.

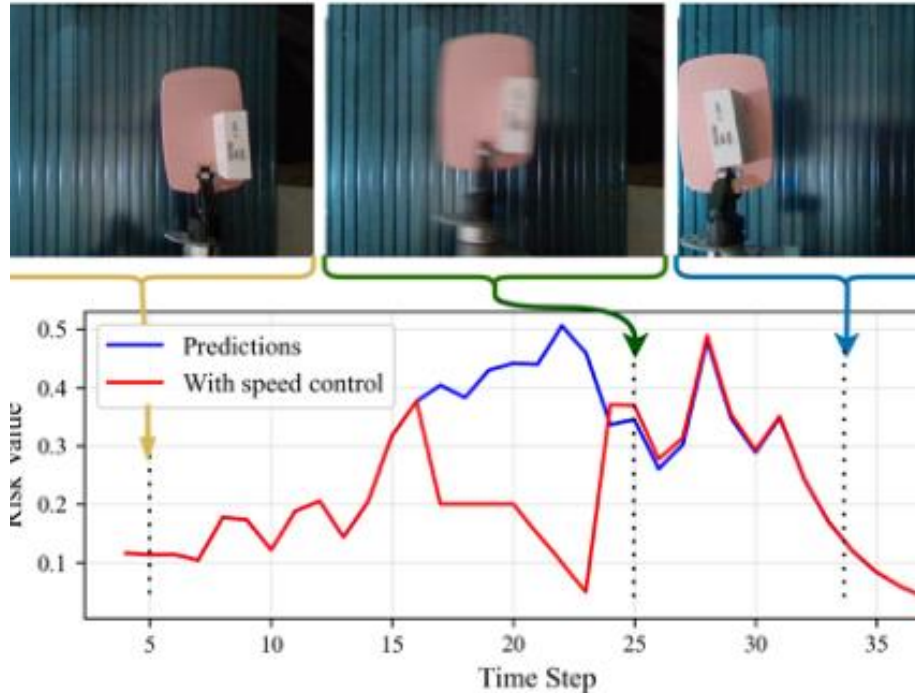


Fig. 6: Trajectory showing a real-world deployment scenario where the risk is set based on the predicted risk value which in turn minimizes the subsequent risk after time step 20.

Velocity control

- Risk = 0 $\rightarrow$ maximum joint speed
- Risk가 증가할수록 speed를 linearly 감소
- Risk = 0.5 $\rightarrow$ minimum joint speed

Example rollout

Original: failure ≈ 0.86 s

Expected slow-from-start: ≥ 1.8 s

Risk-aware control: success ≈ 1.3 s

필요한 순간에만 감속해 reliability를 높이면서 cycle-time penalty를 제한한다.

Conclusion



Key contributions

1. Actionable temporal supervision

historical backtracking을 통해 실제로 intervention 가능한 마지막 시점을 supervision에 넣었다

2. Relative-motion precursor

object와 carrier 사이의 relative motion을 failure precursor로 활용

3. Continuous risk for adaptive execution

image, robot state, recent risk를 이용해 매 timestep continuous risk를 예측하고, 이를 실제 속도 제어까지 연결

Evidence

Simulation and real-world experiments show high accuracy and timeliness; deployment prevents failure with limited cycle-time increase.

PREFAIL asks not only “will it fail?” but whether the prediction arrives while intervention is still useful.

Limitations stated in the paper

1. Single-object setting

Validated with a single object on the carrier; multi-object interactions are not modeled.

2. 2-D ICP under difficult visual conditions

Contour-based 2-D/ ICP can degrade under occlusion, poor lighting, or visual ambiguity. -> 3D

3. Cost of real-world historical backtracking

Real-world t_{stop} labeling requires repeated physical rollouts and resets, making supervision expensive.

4. No general recovery strategy

PREFAIL predicts when to intervene, but does not provide a task-general recovery policy.

Paper suggestion: depth sensing + 3-D motion estimation could improve robustness at additional computational cost.

Thank You

