Supplementary Material for Latent-Space Prediction Error and Fuzzy Risk-Scaled Recovery for Robotic Manipulation

Thanh-Hai Tran [1,2] and Xuan-Bach Le [1,2⋆]

[1] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
[2] Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam
{tthai.sdh241,lexuanbach}@hcmut.edu.vn

This document holds the material the main paper refers to but has no room for: the code release, the full module-coverage matrix and how it was scored, the RSSM parameter breakdown, significance tests alongside the Wilson intervals, and a few implementation details. Reference numbers are those of the main paper, whose list is reproduced in full.

S1 Code, models and data

All controllers, model checkpoints, training scripts and the Webots worlds used to produce the reported runs are available at

https://github.com/haitran-13/acivs2026-supplementary

The repository README maps each reported table and figure to the script that computed it and states the repository's scope: method transparency rather than turnkey reproduction, since the per-episode logs the analysis scripts parse are not hosted there. Two details are not visible from the main text. First, the deployed risk coefficient is $\alpha_t = \max(\alpha_{\text{ANFIS}}, \alpha_{\text{multistep}})$: besides the ANFIS path described in Section 4.4, the controller also rolls the RSSM prior forward 20 steps and raises α_t to a fixed level when the predicted and realised embeddings diverge beyond a threshold. Second, three ANFIS checkpoints are shipped, but only `anfis_transit_v4.pt` (2 inputs, 6 rules) is the risk-control mapper of Section 4.4; the other two serve episode-level logging and are not the source of any reported value.

S2 Module coverage and scoring criteria

Table 1 is the full version of the comparison summarised in Section 2. A system is marked present on a column when its published description meets the corresponding criterion.

M1 Perception: it takes raw images as input, not a pre-extracted state vector.

⋆ Corresponding author

M2 World Model: it learns a predictive model of future states or latents and uses that model at run time.
M3 Risk Assessment: it derives an anomaly or risk quantity and consumes that quantity itself.
M4 Control: it emits a robot action, rather than an alarm or a report for a human operator.
Prop.: it scales the size of that action continuously with the estimated risk, instead of switching at a threshold.

M3 is the broadest of the five. A raw anomaly score satisfies it, so detector-only systems qualify; requiring a graded, tiered risk level instead would leave only the fuzzy-based entries and this work. In Table 1, ✓ marks a criterion as met and – as not met.

Table 1: Module coverage of ten representative works (2023–2026).

Method	**M1**	**M2**	**M3**	**M4**	**Prop.**
SafeDreamer [5]	✓	✓	✓	✓	–
FARL [6]	✓	✓	✓	✓	–
WM-AD [7]	✓	✓	✓	✓	–
RAPT [8]	✓	✓	✓	✓	–
DreamerV3 [2]	✓	✓	–	✓	–
FIPER [12]	✓	–	✓	–	–
V-JEPA 2 [1]	✓	✓	–	–	–
Fuz-RL [10]	–	–	✓	✓	–
EfficientAD [11]	✓	–	✓	–	–
AHA [13]	✓	–	✓	–	–
Ours	✓	✓	✓	✓	✓

Two entries need explanation. DreamerV3 is marked absent on M3 because its critic estimates return, not anomaly or risk. EfficientAD is marked present on M3 under the broad criterion above, although its output is a raw score rather than a risk level; supplying that risk level is what Module 3 of the present system adds.

S3 RSSM parameter breakdown

Section 4.3 gives the RSSM parameter count as 336,384. The main text specifies the encoder and the recurrent core but not the decoder, so the total is not reconstructible from it. Table 2 lists every tensor in the released checkpoint. The input is the 384-d DINOv2 CLS token, the deterministic state h_t is 128-d and the stochastic state w_t is 32-d, so the prior and posterior heads each emit 64 values, a mean and a log-variance per dimension.

Table 2: Parameters of the released RSSM checkpoint.

Block	**Tensor**	**Shape**	**Params**
Encoder	`net.0.weight`	256×384	98,304
	`net.0.bias`	256	256
	`net.1.weight/bias`	256 each	512
Dynamics	`rnn.weight_ih`	384×38	14,592
	`rnn.weight_hh`	384×128	49,152
	`rnn.bias_ih/hh`	384 each	768
	`prior_head.weight`	64×128	8,192
	`prior_head.bias`	64	64
	`post_head.weight`	64×384	24,576
	`post_head.bias`	64	64
Decoder	`net.0.weight`	256×160	40,960
	`net.0.bias`	256	256
	`net.2.weight`	384×256	98,304
	`net.2.bias`	384	384
Total			**336,384**

The GRU input width of 38 is the 32-d stochastic state concatenated with the 6-d action. The decoder maps $[h_t, w_t]$ $(128 + 32 = 160)$ back to the 384-d embedding, and is the largest of the three blocks.

S4 Significance tests

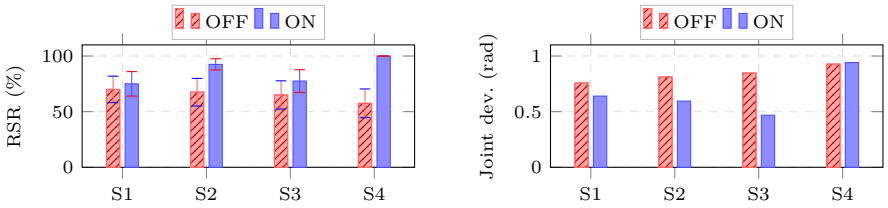
The main paper reports Wilson 95% intervals and treats non-overlap as the comparison criterion. Non-overlap is conservative: two intervals can overlap while the underlying proportions still differ significantly. Table 3 gives Fisher's exact test (two-sided) on the same success counts, $N = 40$ per mode per scenario; these are the counts behind the RSR values in Table 4.

Table 3: Fisher's exact test on OFF vs. ON recovery success.

Scenario	**OFF**	**ON**	p
S1 Vibration	28/40	30/40	0.80
S2 Jerky	27/40	37/40	0.010
S3 Pause	26/40	31/40	0.32
S4 Speed	23/40	40/40	1.7×10^{-6}
Pooled	104/160	138/160	1.4×10^{-5}

The conclusions are unchanged: S2 and S4 separate, S1 and S3 do not. The p-values are the firmer statement: the S2 intervals clear each other by only 0.2 pp, so one episode either way would flip the non-overlap test.

Fig. 1 plots the same Table 4 numbers: per-scenario OFF/ON RSR (a) and mean joint deviation (b).



(a) RSR (Wilson 95% CI).

(b) Mean joint deviation.

Fig. 1: Per-scenario recovery (a) and trajectory quality (b); $N = 40$ per mode, OFF (red, hatched) vs. ON (blue, solid). S5/S6 omitted (0% RSR in both modes; L1 in the main paper).

S5 Implementation details

Confidence ramp (Section 5.3). The ramp and the KL gate are one condition, not two: α_t is multiplied by min(1, transit step/100) *only while* $\mathrm{kl}_{\mathrm{latest}} < 0.65$; once KL exceeds that value the ramp is bypassed and α_t passes through undamped, so a strong early signal is not suppressed. The ramp is applied identically in OFF and ON, so the two modes see the same α_t and the comparison stays matched.

What a "session" is (Section 6.4). The latency figures aggregate 9,969 step-level samples from 19 deployed controller sessions. A session is one launch of the Webots world running a sequence of episodes under a fixed scenario; the 19 sessions span seven scenario types, which is broader than the four recoverable scenarios (S1–S4, 4×40 episodes) used for RSR, so the latency sample and the recovery sample are not in one-to-one correspondence. Six of the seven are S1–S6; the seventh is an obstacle disturbance that was dropped from the evaluation because it is not recoverable by a joint residual (L1), and its 668 samples do not carry the reported figures: excluding them leaves P95 = 34.21 ms and mean 25.63 ms.

The $x_{3,std}$-only ablation (Section 5.4). This configuration thresholds the raw pixel-motion feature at a fixed score and was not calibrated to a target FPR, as Section 5.4 states. A post-hoc check supports the reported 0% TPR: running the same controller on 80 nominal episodes and setting the threshold just above the largest nominal score, which is the 0%-FPR rule used for B1/B2, yields 1/160 = 0.6% TPR, with an AUROC of 0.526 against the nominal set. The pixel-motion signal is therefore uninformative at any operating point, not only at the one used.

Residual magnitude. EWR in Table 4 is the mean of $\|\alpha_t \delta a_t\|$ over TRANSIT steps, not a bound. The per-step bound is the 0.30 rad clip on δa_t; the mean staying below 0.20 rad is a statement about typical intervention size, not a guarantee about any single step.

References

1. Assran, M., et al.: V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. arXiv:2506.09985 (2025)
2. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse domains through world models. arXiv:2301.04104 (2023)
3. Zadeh, L.A.: Fuzzy sets. Information and Control **8**(3), 338–353 (1965)
4. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **SMC-15**(1), 116–132 (1985)
5. Huang, W., Ji, J., Xia, C., Zhang, B., Yang, Y.: SafeDreamer: Safe reinforcement learning with world models. In: ICLR (2024)
6. Li, H., et al.: Failure-Aware RL: Reliable offline-to-online reinforcement learning with self-recovery for real-world manipulation. arXiv:2601.07821 (2026)
7. Domberg, F., Schildbach, G.: World models for anomaly detection during model-based reinforcement learning inference. arXiv:2503.02552 (2025)
8. Munn, H., Tidd, B., Bohm, P., Gallagher, M., Howard, D.: RAPT: Model-predictive out-of-distribution detection and failure diagnosis for sim-to-real humanoid deployment. arXiv:2602.01515 (2026)
9. Johannink, T., et al.: Residual reinforcement learning for robot control. In: ICRA, pp. 6023–6029 (2019)
10. Wan, X., Yang, C., Yang, C., Song, J., Sun, M.: Fuz-RL: A fuzzy-guided robust framework for safe reinforcement learning under uncertainty. In: NeurIPS, vol. 38 (2025)
11. Batzner, K., Heckler, L., König, R.: EfficientAD: Accurate visual anomaly detection at millisecond-level latencies. In: WACV, pp. 127–137 (2024)
12. Römer, R., Kobras, A., Worbis, L., Schoellig, A.P.: Failure prediction at runtime for generative robot policies. In: NeurIPS, vol. 38 (2025)
13. Duan, J., et al.: AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. arXiv:2410.00371 (2024)
14. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. MIT Press (2018)
15. Michel, O.: Webots™: Professional mobile robot simulation. Int. J. Adv. Robotic Syst. **1**(1), 39–42 (2004)
16. Oquab, M., et al.: DINOv2: Learning robust visual features without supervision. TMLR (2024)
17. Jang, J.-S.R.: ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**(3), 665–685 (1993)
18. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: ICML, pp. 1861–1870 (2018)
19. Tran, T.-H., Le, X.-B.: Industrial visual anomaly detection in robotics: methods, datasets, and deployable system architectures. In: Advances and Trends in Artificial Intelligence, LNCS, vol. 16616, pp. 105–119. Springer, Singapore (2027)