DongGeon (Lloyd) Kwak

B.E. Computer Science and Engineering, SeoulTech

kwaklloyd@gmail.com • lloydkwak.github.io • github.com/lloydkwak



PROFILE

Fourth-year undergraduate in Computer Science and Engineering (minor in Applied AI), graduating in August 2027; GPA 4.13/4.50, 8th of 114 in major. I study how pretrained robot policies can keep working when the robot's body changes. Before that I worked as a software developer at two companies for eight months.

RESEARCH INTERESTS

Generalist robot learning • cross-embodiment learning • lifelong robot learning (adaptation, failure detection and recovery) • long-horizon manipulation with task and motion planning

OVERVIEW

Period	Work	Context
2025 – now	FAVOR	Robot learning research; first-author manuscript in preparation
2026 – now	MTREA	Capstone Design project, two-person team
2026	MakeYourBrick	SeoulTech ST startup club
2025	RL-Arena	Open Source Software course, two-person team; outstanding project
2025	AI-generated image detection	Team project, two people
2025	Logisim MLP	Digital Logic and Systems course; outstanding project
2022 – 2023	Laver, CHASER	Software developer

PUBLICATIONS AND PATENTS

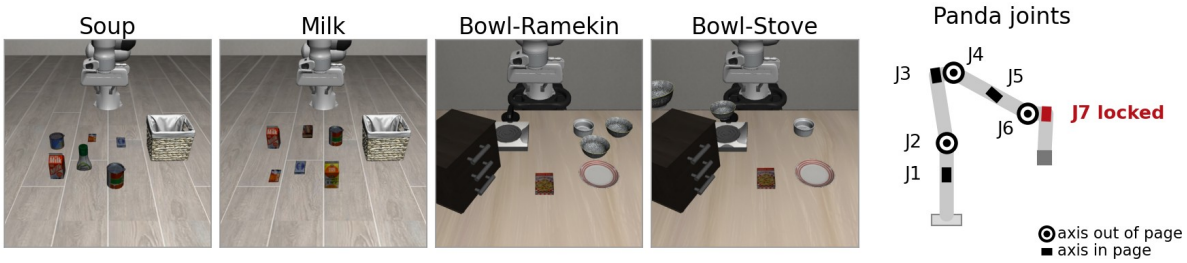
- **D. Kwak**. FAVOR: Fault-Aware Visuomotor Policy Adaptation via Ordered Retargeting. Manuscript in preparation.
- K. Lee, G. Lee, **D. Kwak**. Real-time crack detection system for construction sites. Korean Patent 10-2647135, registered March 2024.

FAVOR

Jul. 2025 – present • Research feedback: Prof. Ji-Hyeong Han; weekly lab meetings

QUESTION

When one joint of a robot arm locks or loses part of its range, can a pretrained diffusion policy still complete its task without retraining, and where should the correction be applied?



The four simulated tasks (LIBERO) and the Franka Panda joints.

WHAT I DID

- Trained a joint-space diffusion policy on a healthy Franka Panda (4 LIBERO tasks) and injected locked and range-limited joint faults at four levels of severity, including faults that appear mid-task.
- Compared seven corrections applied at three stages of inference: choosing among sampled actions, guiding the denoising process, and inverse kinematics after sampling. All methods ran on the same seeds and diffusion samples, and the main result was replicated on new seeds.
- Found that restoring the end-effector position first, and orientation only with the remaining joints (priority IK), worked best from moderate faults on. Showed that weighted IK without a posture term is equivalent to this to first order, which explains when the weighted baselines fail.
- Showed that the end-effector error left after correction predicts which faults are recoverable (AUC 0.94 on the same conditions).

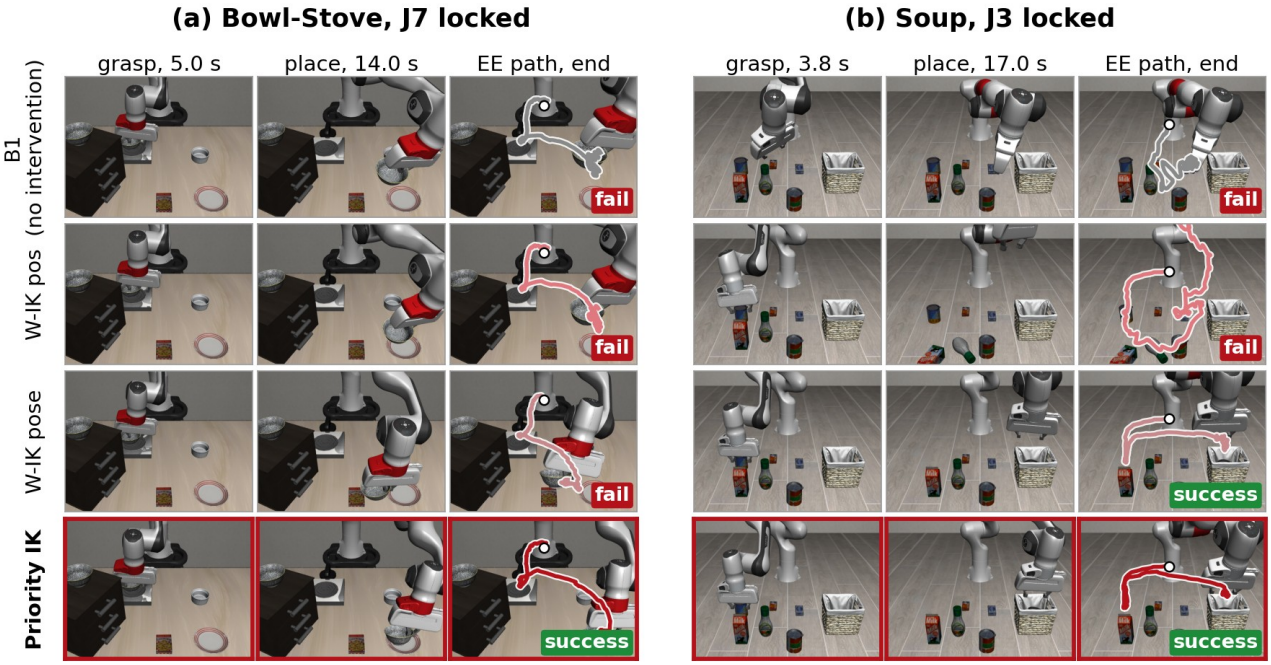
RESULTS

Fault level	No correction	Denoising guidance	Weighted IK (pos.)	Weighted IK (pose)	Priority IK
Mild	0.66	0.70	0.75	0.70	0.70
Moderate	0.42	0.44	0.59	0.51	0.64
Severe	0.28	0.31	0.45	0.39	0.58
Locked	0.11	0.19	0.30	0.32	0.48
All 112 conditions	0.37	0.41	0.52	0.48	0.60

Task success rate over 20 paired seeds per condition. Denoising guidance here is the E-C-I projection method.

LIMITATIONS

- The method needs an accurate lock angle; a 0.01 rad error lowers success from 0.67 to 0.39, so the angle should come from the joint encoder.
- Some faults (locked joints 2 and 4) cannot be compensated by any of the methods.
- All results so far are in simulation. I am now testing on a real xArm7.



FAVOR: the same episode under four methods for two faults. The bottom row (Priority IK) succeeds in both.

MTREA

Sep. 2026 – present • Capstone Design, SeoulTech • two-person team

QUESTION

When the $\pi_{0.5}$ model is fine-tuned on one task with a new low-cost robot arm (SO-101), does it lose performance on other tasks it could do before?

WHAT I DID

- Wrote the evaluation protocol with my teammate: fine-tune on pick-and-place only, then test on stacking, pushing, pouring and opening a drawer. Zero-shot performance is not a usable baseline because the new body alone keeps it near zero, so naive full fine-tuning is the baseline and a model fine-tuned on each held-out task is the ceiling.
- Built arm calibration and leader–follower teleoperation with a safety lock for data collection, a training wrapper with dataset sync and remote job control for the GPU server, and an asynchronous server that runs the trained policy on the arm.
- Measured memory use on the shared GPU (an A100 slice with 40 GB) and chose full fine-tuning with batch size 16, the largest that fits.

STATUS

Data collection, training and deployment work end to end. The evaluation on the held-out tasks is next.

Other Projects

MakeYourBrick May 2026, startup club

A web app that turns a single photo into a LEGO model. The user selects an object, SAM2 segments it, SAM 3D Objects reconstructs it in 3D, and the shape is converted into colored bricks in the LDraw format. Shadows baked into the reconstructed texture produced black bricks, which I fixed by softening shadows before color quantization. The example shown uses 10,263 bricks.



RL-Arena Nov. – Dec. 2025, Open Source Software course

A platform where users train reinforcement-learning agents, submit them, and see them ranked by ELO. Two-person team. I built the environment package (published on PyPI as *rl-arena*), the service that runs each match in an isolated container, and most of the backend, including automatic matchmaking; the web interface was shared. Selected as an outstanding project in the course.

AI-generated image detection Dec. 2025, team of two

A classifier that separates real photos from images made by diffusion models (115k images). Generated images had a narrower spread of pixel intensities, and preprocessing based on this raised test accuracy from 72.6% to 96.3% for ResNet-50 and from 60.7% to 88.9% for EfficientNet-B4.

Logisim MLP Spring 2025, Digital Logic and Systems course

A 256-128-10 MNIST classifier built as a digital logic circuit in Logisim, with about 34,000 trained weights stored in ROM, to understand how a neural network computes at the hardware level. Selected as an outstanding project in the course.

Work Experience

Laver, Jun. – Oct. 2022 • CHASER, Nov. 2022 – Jan. 2023 • software developer

Developed an object-detection system from model development to product integration, built backend services and APIs, and wrote an IFC-to-XML converter for building information models. I am also a co-inventor (third of three) on a Korean patent for a crack detection system in which mobile robots record video at construction sites and cracks are detected and tracked across frames.