부록 — AI에게 원리를 물어 배운 기록

「ChatGPT Edu 활용 상시 공모전」참가신청서 첨부자료 · 박세환

이 앱과 그 위에 세운 로봇 연구는 제가 처음 다루는 것들 위에 있었습니다. 맥 앱도, 기기에서 도는 언어 모델도, 강화학습도 배운 적이 없었습니다. 그래서 코드를 먼저 받지 않고 원리를 먼저 묻고, 답을 받으면 곧바로 돌려서 확인하는 방식으로 진행했습니다. 아래는 실제 대화에서 그대로 가져온 것입니다.

> 도구를 밝힙니다. 앱의 구조를 잡는 단계는 학교 ChatGPT Edu 계정의 Codex로 했고, 로봇 연구 단계의 이론 학습과 구현에는 Claude를 함께 썼습니다. 이 부록은 후자의 기록입니다.

1. 무엇을 배우려는지부터 물었다

제가 물은 것 (2026-09-05)

"지금 내 맥북에서 ssh 접속으로 spark에 접속해서 isaac sim, lab 좀 해보려고 하는데 내가 잘 모르거든. 그래서 일단은 기초적인 거부터 자세히 설명을 해주면 좋을 것 같아. (...) sim, lab 차이가 뭐고, RL, IL이 뭐고 자세히 설명해줘."

받은 답에서 배운 것

시뮬레이터(Isaac Sim)와 그 위의 학습 프레임워크(Isaac Lab)가 다른 층이라는 것, 강화학습(RL)은 보상을 스스로 최대화하며 배우고 모방학습(IL)은 사람의 시연을 따라 배운다는 것. 이 구분이 이후 "시뮬에서 RL로 먼저 감을 잡고, 실물 데이터로는 IL을 한다"는 계획의 뼈대가 되었습니다.

2. 수식이 필요한 곳은 수식으로 물었다

제가 물은 것 (2026-09-05)

"MDP(마르코프 결정 과정) 라 이거좀 이론적으로 좀 더 설명해줘."

받은 답에서 배운 것

강화학습 문제가 상태·행동·전이확률·보상·할인율의 다섯 요소로 표현된다는 것, 그리고 그것이 제가 돌리고 있던 학습의 로그(Episode_Reward, success_rate)와 어떻게 대응되는지. 이론을 물은 이유는 로그를 읽기 위해서였고, 실제로 그다음부터 학습이 실패하는 이유를 로그에서 읽을 수 있게 되었습니다.

3. "지금 하는 것과 저것이 뭐가 다르냐"를 물었다

제가 물은 것 (2026-09-06)

"음 LORA 파인튜닝이랑 다른거냐? 그거 관련해서 좀 설명해줄래?"

받은 답에서 배운 것

당시 돌리던 SmolVLA 학습이 비전·언어 부분은 얼리고 액션 전문가 98M만 통째로 업데이트하는 부분 전체 미세조정이고, LoRA는 가중치를 바꾸는 방식 자체가 다른 기법이라는 것. 이 구분을 알고 나서 학습 범위(전체·전문가만·비전 동결)를 이 기계의 실측 속도에 맞춰 고를 수 있게 되었습니다.

4. 막혔을 때 "연구자들은 보통 어떻게 하나"를 물었다

제가 물은 것 (2026-09-06)

"학습 개선시키고 큐에 넣자. 비슷한 경우의 레퍼런스랑 우리 경우의 차이점을 더 분석해볼래? 내가 RL은 잘 몰라서. 이럴때는 보통 연구자들은 어떻게 해?"

받은 답에서 배운 것

세 번의 실패가 전부 "문턱 옆에 서기"라는 같은 패턴이었다는 진단과, 보상 설계에서 흔히 쓰는 접근들. 그 뒤 제어 방식을 관절 각도에서 손 끝 위치(역기구학) 로 바꾸었고, 그때부터 `lifting_object` 항이 3,932만 스텝 만에 처음으로 0을 벗어났습니다.

5. 하려는 연구가 의미가 있는지도 물었다

제가 물은 것 (2026-09-05)

"내가 나중에 모터의 전류값을 이용해서 간접적인 텍타일등을 공부해 보고 싶은데, 지금 텔레옵하면서 모터에서 전류나 이런거를 간접적으로 읽어서 데이터 수집에 저장할 수 있나? 그리고 궁금한거는, 이런 방향의 연구나 그런게 객관적으로 의미가 있는 연구인가?"

받은 답에서 배운 것

서보의 레지스터에서 실제로 읽히는 값이 무엇인지(`Present_Load`는 채워지고 `Present_Current`는 이 펌웨어에서 사실상 비어 있다는 것)를 먼저 측정으로 확인한 뒤, 그 위에서 연구로서의 의미를 논의했습니다. 하고 싶은 것을 곧바로 시작하지 않고, 그 전제가 실제로 성립하는지부터 재는 습관이 여기서 생겼습니다.

6. 답을 그대로 믿지 않았다 — AI를 교정한 기록

제가 지적한 것 (2026-09-06)

"근데 sparkq를 할때 spark 충돌 문제랑 이거를 하나에서 모아서 관리하면 편하겠다 라는 생각으로 만든건데 너가 그걸 그렇게 우회해 버려도 되나? 물론 이론적으로 우회해도 상관 없었을 수 있지만, 이게 나중에 맥락을 모르는 다른 agent가 와서 다른거 실행시켜 버리고 하면 큰 문제가 될수도 있을것 같은데"

무슨 일이었나

GPU가 하나뿐인 학습 서버 앞에 제가 만들어 둔 작업 큐가 있었는데, AI가 그것을 우회해 학습을 직접 띄웠습니다. 편의로는 문제가 없어 보였지만 실제로는 제 잡을 세 번 깨뜨렸고, 무엇보다 맥락을 모르는 다음 사람이 같은 일을 반복하게 만드는 구조였습니다. 지적한 뒤 그 규칙을 저장소의 문서로 못 박았습니다.

이 기록을 굳이 넣는 이유는, AI를 쓰는 일에서 제일 중요한 것이 답을 받는 능력이 아니라 받은 답이 내 전제를 깨고 있지 않은지 보는 능력이라고 생각하기 때문입니다.

돌아보며

배운 것은 개별 지식보다 순서였습니다. 모르는 분야를 만나면 ① 무엇을 배워야 하는지부터 묻고 ② 필요한 만큼만 이론으로 내려갔다가 ③ 곧바로 돌려서 확인하고 ④ 답이 내 전제와 어긋나면 그것부터 따지는 것. 이 순서는 로봇에서만 쓰이는 것이 아니라 다음에 처음 만나는 분야에서도 그대로 쓸 수 있고, 그것이 이 사례에서 다른 구성원과 나누고 싶은 부분입니다.