PlantPulse® **AURA Models**

TECHNICAL WHITEPAPER

기술 백서 · REV. 2026.09

공장의 말로 답하는 산업 AI 모델.

KOPENS 가 직접 학습시킨 산업 AI 모델 셋 — 진단 · 검색 · 정밀 정렬. 전문가 혼합 구조로 **GPU 한 장**에 들어가고, 계산은 수식으로 풀며, 재지 않은 것은 «측정 안 됨» 이라 말합니다. 채점은 심판 모델이 아니라 **수식 · 스키마 · 운영 기록**이 합니다. 이 판은 **구성과 검증 방법**을 적습니다 — 성능 수치는 재측정 뒤 개정판에 싣습니다.

3	65,954	0	1
모델 — 35.9B · 4B · 0.6B	학습 샘플(최신 조립본)	심판 모델 — 기계 채점	GPU 로 서빙

2026년 9월

AURA · 온프레미스 / 폐쇄망 · 기술 검토자용

kopens®

00 · 요약

요약 — 범용 모델이 공장에서 위험해지는 지점

범용 모델은 세상에 대해 많이 알지만 **당신 공장의 압축기에 대해서는 모릅니다.** 베어링이 무엇인지는 기꺼이 설명하고, **재지도 않은 진동 값을 기꺼이 지어냅니다.** 공장에서 위험한 것은 뒤쪽입니다 — 값이 없다는 사실이 「정상」으로 둔갑하면 아무도 들여다보지 않은 설비를 운영자가 멀쩡하다고 믿습니다. **AURA** 는 그 반대로 훈련한 모델 계열이고, **KOPENS 가 데이터 설계부터 학습 · 평가 · 양자화 · 서빙까지 직접** 만듭니다 — 산업 데이터를 20년 다뤄 온 회사가 **그 데이터로 모델을 학습시킬 수 있게 된 것**이 이 제품의 출발점입니다.

3
언어 트랙 모델 — 진단(35.9B MoE) · 검색(4B) · 정밀 정렬(0.6B)

5
벤치마크 — 다섯 **전부 심판 모델 없이** 기계 채점

158
현장 진단 문항 — 정답이 **우리 밖**(운영 기록)에서 옵니다

1
GPU 로 언어모델 서빙(양자화 21 GB)

지금 어디까지 왔나 — «도는 것» 과 «만드는 중» 을 가릅니다

단계	무엇	뜻
제품으로 돕니다	언어모델 35.9B · 검색(4B) · 정밀 정렬(0.6B)	사내 추론 서버가 **학습본을 실제로 서빙** 중입니다 — 베이스가 아니라 우리가 학습시킨 판입니다
다음 단계	시계열 트랙	파형을 보는 **별도 계열** — 백만 파라미터급이라 **GPU 없이** 돕니다. 제품 적용은 다음 판에서 다룹니다(15장)
다음 단계	비전	**잣대를 먼저 만들었습니다** — 용접 결함 판정 벤치와 산업 이미지. 다음 판에서 다룹니다

목차

01 강점 — 수치 이전에 «구조» 가 만드는 차이 3
02 구성 — 언어 트랙의 세 모델과 파형을 보는 계열 4
03 학습과 데이터 — 정답이 어디서 오나 5
04 레시피 — 고장 메커니즘 · 파인튜닝 · 양자화 6
05 무엇을 보는가 — 고장 메커니즘 지도 7
06 가르는 법 — 순서가 다르면 다른 고장이다 8
07 데이터를 고르는 법 — 정답이 밖에서 오는가 9
08 확보한 데이터 — 무엇을 덮고 있나 10
09 시뮬레이터 — 물리를 먼저, 라벨은 유도 11
10 모른다고 말하기 — 판단 보류도 정답이다 12
11 정답의 출처 — 계산 · 규격 · 절차 · 도구 13
12 학습 비중 — 무엇을 얼마나 넣는가 14
13 정체성과 범위 — 프롬프트가 아니라 학습으로 15
14 검색 — 답변 품질의 상한 16
15 시계열 트랙 — 파형은 작은 모델로 17
16 평가 — 심판 모델 없이 재는 법 18
17 현장 진단 — 거짓 경보를 따로 세는 이유 19
18 온프레미스 배치 · 다음 판 · 문의 20

이 판에 성능 수치가 없는 이유. 학습본이 **제품 경로에 막 올랐습니다.** 수치는 **그 경로에서 다시 잰 뒤**에 싣습니다 — 예전 경로에서 잰 값을 미리 박아 두면 **나중에 고쳐도 그 숫자가 근거로 돌아다닙니다.** 그래서 이 판은 **구성과 검증 방법만** 확정해 적습니다. 무엇을 어떻게 재는지는 이미 정해져 있으므로, 검토자는 **같은 방식으로 자기 환경에서 직접 잴 수 있습니다**(16장).

독자. 기술 검토자 · 데이터 과학자 · 설비 엔지니어. 영업 자료가 아닙니다 — **무엇을 어떻게 재는지**를 검토자가 다시 잴 수 있을 만큼 적었습니다(16장).

01 · 강점

설계가 주는 것 — 수치 이전에 «구조» 가 만드는 차이

아래 여섯은 벤치마크 점수가 아니라 **설계에서 곧바로 따라 나오는 성질**입니다. 그래서 판올림으로 흔들리지 않고, 이 판에 성능 수치가 없어도 그대로 성립합니다.

배치

GPU 한 장에 들어간다

전체 35.9B 중 토큰당 **3B 만** 활성화되는 전문가 혼합 구조에, 양자화 체크포인트가 **21 GB** 입니다. 같은 품질을 노리는 dense 구성 대비 **연산이 9분의 1** 이고, 그래서 한 장으로 섭니다. **여덟 장이 필요한 모델은 성능과 무관하게 대부분의 공장에 들어가지 못합니다.**

운영

판올림에 색인을 다시 만들지 않는다

검색 모델의 **임베딩 차원(2,560)을 판 사이에 고정**했습니다. 새 판으로 갈아끼워도 벡터 색인이 그대로 쓰입니다 — 수천만 조각을 다시 임베딩해야 하는 판올림은 **결국 미뤄지고**, 미뤄진 판올림은 없는 것과 같습니다.

신뢰

0 과 «측정 안 됨» 을 가른다

「작업지시 0건」은 측정이고 「연결이 끊겨 못 읽었다」는 측정이 아닙니다. 뒤쪽을 앞쪽처럼 보고하면 **아무도 들여다보지 않은 설비를 정상으로 믿게 됩니다.** 학습 데이터와 평가 문항 양쪽에서 이 구분을 강제합니다.

검증

채점에 심판 모델을 쓰지 않는다

정답이 **수식 · 스키마 · 운영 기록**에서 나오므로 채점은 기계가 합니다. 같은 입력이면 **누가 언제 돌려도 같은 값**이고, 고객이 자기 환경에서 같은 방식으로 다시 잴 수 있습니다. 모델이 모델을 채점하면 그 둘의 편향이 함께 들어갑니다.

학습

지어낸 정답을 배우지 않는다

데이터의 정답이 **시뮬레이터 규칙 · 공식 · 도구 스키마**에서 나옵니다. 강한 모델로 만든 도구 궤적도 **스키마 검증을 통과한 것만** 씁니다(통과율 83%) — 떨어진 17%를 배웠다면 모델이 **있지도 않은 인자**를 자신 있게 넣었을 것입니다.

현장

양치기 소년을 만들지 않는다

현장 진단 평가에서 **거짓 경보를 미탐지와 따로** 셉니다. 둘을 «정확도» 하나로 합치면 **모든 구간에 경보를 울리는 모델**이 좋은 점수를 받습니다. 그런 조수는 일주일이면 꺼지고, 꺼진 조수는 아무것도 잡지 못합니다.

자체 기술

모델을 «쓰는» 회사가 아니라 «만드는» 회사

데이터 설계 · 학습 · 평가 계측기 · 양자화 · 서빙까지 **한 회사 안에서** 돕니다. 그래서 현장에서 틀린 것이 나오면 프롬프트로 덮는 것이 아니라 **데이터와 학습으로** 고칩니다 — 17장의 재측정 계획이 그 예입니다.

한 벌로 맞물림

플랫폼과 같은 손으로 만든다

도구 호출 학습·평가에 쓰는 스키마가 **가동 중인 PlantPulse 플랫폼의 실제 도구 정의**입니다(57종). 모델과 플랫폼이 따로 자란 조합에서는 얻기 어려운 맞물림입니다 — **남의 모델에 우리 도구를 설명하는** 대신 우리 도구로 우리 모델을 가르칩니다.

범위를 좁힌 것도 강점입니다. 설비 · 공정 · 품질 · 안전에 답하고 정치 · 외교는 범위 밖으로 둡니다. 한국어가 기본이고 영어로도 답합니다. **무엇에 답하지 «않을» 것인가를 정하는 일**에 학습의 5%를 썼습니다 — 범위 밖에서 그럴듯하게 답하는 조수는 범위 안에서도 그럴듯하게 답하기 때문입니다.

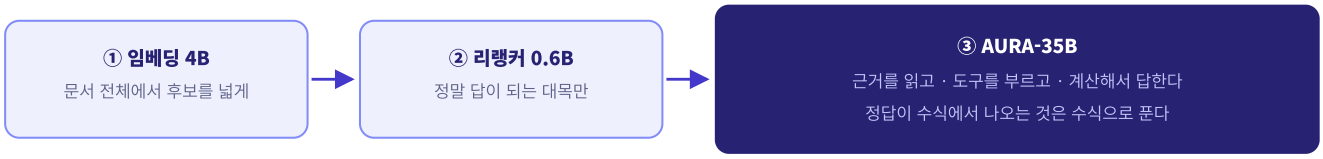
02 · 구성

언어 트랙의 세 모델 — 넓게 건지고, 정밀하게 고르고, 답한다

하나의 큰 모델이 전부를 하는 대신 **역할을 셋으로 나눴습니다.** 검색은 싸고 넓어야 하고, 정렬은 정확해야 하며, 답하기는 도구를 부를 줄 알아야 합니다. 셋을 한 모델에 밀어 넣으면 어느 하나가 나머지의 비용을 떠안습니다.

모델	크기	형식
AURA-35B	**35.9B** 토큰당 활성 **3B**	전문가 혼합(MoE). 진단 · 계산 · 도구 호출 · 보고서. 서빙 컨텍스트 **32,768** 토큰(학습 시퀀스 2,048)
AURA-Embedding-4B	4.0B	출력 **2,560차원**, 입력 최대 1,024토큰. 마지막 토큰 풀링, 질의에만 지시문을 붙이는 **비대칭 인코딩**(문서에는 붙이지 않습니다)
AURA-Reranker-0.6B	0.6B	질의와 문서를 **함께 읽는** 교차 인코더. 입력 최대 1,024토큰, 상위 20~100개에만 적용

셋이 맞물리는 길



임베딩 차원은 판올림 사이에 바뀌지 않습니다. 검색 모델을 새 판으로 갈아끼워도 **벡터 색인을 다시 만들 필요가 없습니다.** 수천만 조각을 다시 임베딩하는 일은 한 번 해 보면 다음부터 두려워지는 작업이라, 차원을 계약처럼 고정해 두었습니다.

왜 전문가 혼합(MoE)인가 — 현장에 «들어갈 수 있는가»의 문제

전체 35.9B 중 토큰당 **3B 만** 활성화됩니다. 같은 품질을 노리는 dense 구성과 견주면 **연산이 9분의 1**이고, 그래서 **GPU 한 장**으로 섭니다. 여덟 장이 필요한 모델은 성능과 무관하게 대부분의 공장에 들어가지 못합니다 — 이 선택이 곧 도입 가능 여부를 가릅니다. 같은 이유로 **전문가 혼합을 택했습니다** — dense 구성은 연산이 9배입니다.

검색 모델은 어떻게 학습했나

임베딩	대조학습(InfoNCE) — 같은 배치 안의 다른 항목을 부정 예로 쓰고 **헷갈리게 만든 오답**을 따로 넣습니다. 비슷해 보이지만 답이 아닌 문서를 가려내는 힘이 거기서 옵니다
리랭커	후보 목록에서 정답을 고르는 listwise 교차엔트로피 — **순위를 매기는 일 자체를** 학습합니다

파형은 «다른 물건» 으로 봅니다 — 시계열 트랙

언어모델은 **왜 그런지를 설명**하는 데 강하지만, 초 단위로 들어오는 파형에서 **무엇이 이상한지**를 찾는 일은 성질이 다릅니다. 그래서 파형 쪽은 **백만 파라미터급의 별도 계열**로 봅니다 — **GPU 없이 돌 만큼 작아서** 엣지에 얹을 수 있고, 현장마다 따로 맞춰도 비용이 거의 들지 않습니다. 제품 적용은 **다음 판에서 다룹니다** — 값을 확인한 뒤에 넣는 것이 이 제품의 순서입니다.

거기서 나온 것 하나는 이미 씁니다 — «모델을 부르지 않는 쪽이 나은 구간이 있다». 설비의 **평소 기준선 대비 편차**를 통계로 보는 것만으로, 모델을 한 번도 부르지 않고 이상 구간이 갈리는 경우가 있었습니다. 그래서 그 구간은 **모델로 풀지 않습니다.** 모델을 쓰는 자리와 통계로 충분한 자리를 가르는 것이 «큰 모델을 더 크게» 보다 먼저입니다.

03 · 학습과 데이터

한 단계, 어댑터 — 그리고 정답이 어디서 오는가

학습은 **지도 미세조정(SFT) 한 단계**입니다. 사전학습은 하지 않았고 선호학습(DPO·RLHF)도 **하지 않았습니다**. 가중치는 어댑터(LoRA)로 얹어 기반 가중치를 건드리지 않습니다. 그리고 산업 문답은 인터넷에 없으므로 **만들어야** 하는데, 핵심은 **만든 문답의 정답을 누가 보증하느냐**입니다 — 다른 모델에게 물어 만든 정답을 배우면 그 모델의 착각까지 함께 배웁니다.

최신 판의 규모

최신 조립본	**65,954**건 — 판마다 다시 조립합니다
직전 판	49,401건 · 에폭 2 · 약 **7,500만** 토큰
시퀀스 길이	2,048 (서빙 컨텍스트는 32,768)
언어모델 학습 시간	데이터센터급 GPU **1장 · 6시간** (직전 판 기준)
검색 모델 학습 시간	임베딩 30분 · 리랭커 1시간

구성 비율

정체성 · 가드레일	5%
도구 호출	5%
진단 · 계산 · 센서 · 규격 · 절차	나머지 90%

가드레일이 5% 인 것은 **「무엇에 답하지 않을 것인가」도 학습 대상**이기 때문입니다 — 범위 밖 질문에 그럴듯하게 답하는 조수는 범위 안에서도 그럴듯하게 답합니다.

데이터 다섯 갈래

갈래	정답이 어디서 오나
물리 시뮬레이터 합성	설비 **18종** × 고장모드 **9종** 의 센서 시계열. 정답이 **시뮬레이터 규칙**에서 나옵니다
수식 생성	산업 계산 문항. 정답이 **공식으로 확정**되므로 채점에 심판이 필요 없습니다
공개 원문	산업안전보건법 · OSHA 등 규격 조문과 현장 용어
절차 문서	SOP 에서 만든 **근거 기반** 문답 — 문서에 있는 것만으로 답하게
도구 스키마	가동 중인 플랫폼의 OpenAPI · MCP 정의에서 생성한 도구 호출 예제

최신 조립본의 구성 — 이벤트 체인 40,000 · 계산 8,546 · 센서 5,009, 그리고 도구 호출 · MCP · 플랫폼 · 규격 · 용어. **계산이 0건이던 판에서 계산 능력이 무너진 적이 있어** 조립할 때마다 갈래별 건수를 먼저 봅니다.

> **증류한 것은 «검증해서» 씁니다.** 도구 궤적 일부는 더 강한 모델로 멀티턴 예제를 만들었습니다. 다만 생성 결과를 **스키마로 전량 기계 검증**해 통과한 것만 학습에 넣습니다 — **통과율 83%**. 나머지 17%는 그럴듯하지만 스키마가 거부하는 호출이었고, 그것을 배웠다면 모델이 **있지도 않은 인자**를 자신 있게 넣었을 것입니다.

정답을 «규칙» 에서 뽑는다

① 시뮬레이터
고장은 «만들어» 본다

설비 18종에 고장모드 9종을 걸어 센서 시계열을 만듭니다. 베어링이 마모될 때 진동·온도·전류가 **어떤 순서로** 움직이는지가 시뮬레이터 규칙에 있으므로, 「이 파형에서 무엇이 일어나고 있나」의 정답이 확정됩니다.

② 수식
계산은 풀어서 낸다

베어링 결함 주파수 · 수명 · NPSH · 동체 두께 · 유량 · OEE · 공정능력 · 입열량. 난수로 문항을 만들고 **정답은 공식이 계산**합니다. 채점도 같은 공식으로 합니다.

③ 규격 · 절차
있는 것만 쓰게 한다

법령 조문과 SOP 로 **근거 기반** 문답을 만듭니다. 핵심은 「문서에 없는 것은 답에도 없어야 한다」를 데이터 단계에서부터 강제하는 것입니다.

④ 도구
스키마가 심판이다

운영 플랫폼의 실제 도구 정의에서 호출 예제를 만듭니다. 강한 모델로 만든 멀티턴 궤적도 **스키마 검증을 통과한 것만** 남깁니다(통과율 83%).

04 · 레시피

설비가 아니라 «고장 메커니즘» 을 가르친다

같은 베어링 마모라도 제조사와 모델마다 데이터를 따로 모으면 **조합이 수십만으로 불어납니다.** 그래서 가르치는 단위를 바꿨습니다 — **부품 · 고장 메커니즘 · 운전 조건**입니다. 메커니즘은 설비를 가리지 않으므로, **학습에 없던 설비에도 그대로 옮겨 갑니다.**

단위를 바꾼 이유

설비 중심 → 메커니즘 중심

설비 중심으로 모으면 **제조사 × 모델 × 자산** 만큼 데이터가 필요합니다. 메커니즘 중심은 **부품 + 메커니즘 + 운전 조건** 으로 접혀서, 같은 양의 데이터가 **훨씬 넓은 설비**를 덮습니다.

한 번 만들어 네 곳에

같은 고장 episode 를 재사용한다

물리 · 규칙 기반 생성기가 만든 **고장 episode 하나**를 학습 · 도구 학습 · 평가 · 벤치마크에 **공통으로** 씁니다. 네 곳이 같은 세계를 보므로 **학습과 채점이 어긋나지 않습니다.**

가르치는 것은 «답» 이 아니라 «추적 과정» 입니다

「베어링 온도가 높은 이유는?」에 답하는 것이 목표가 아닙니다. 목표는 이 흐름을 **스스로 밟는 것**입니다 —

```
알람 → 자산 확인 → 태그 탐색 → 추세 조회 → 이력 조회 → 가능한 메커니즘 나열
     → 후보별 증거 비교 → 부족하면 데이터 더 요청 → 원인 추론 → 조치 제안 → 정상화 검증
```

그래서 학습 데이터가 **「증상 → 정답」 짝**이 아니라 **증거를 모아 좁혀 가는 궤적**입니다. 도구 호출이 데이터에 섞여 있는 것도 같은 이유입니다 — 현장에서 원인을 좁히려면 **데이터를 더 가져와야** 하기 때문입니다.

파인튜닝 레시피

단계	**지도 미세조정(SFT) 한 단계** — 사전학습 없음
가중치	**어댑터(LoRA)** 로 얹습니다 — 기반 가중치를 건드리지 않습니다
시퀀스 · 에폭	2,048 · 2
학습 비용	데이터센터급 GPU **1장 · 6시간**
검색 모델	임베딩 30분 · 리랭커 1시간

최신 조립본	**65,954**건 — 판마다 다시 조립합니다
이벤트 체인	40,000 — 증거를 좁혀 가는 궤적
산업 계산	8,546 — 정답이 공식에서 나옵니다
센서	5,009
도구 · 규격 · 용어	도구 호출 · MCP · 플랫폼 · 규격 조문 · 현장 용어

조립할 때마다 갈래별 건수를 먼저 봅니다. 한쪽이 비면 그 능력이 조용히 약해집니다 — 실제로 계산 데이터가 빠진 판에서 계산이 흔들린 적이 있어, **건수 확인을 조립 절차의 첫 단계**로 두었습니다.

양자화 — 한 장에 올리면서 «흔들리지 않게»

① 캘리브레이션

도메인 데이터로 보정한다

4비트로 접을 때 쓰는 보정 표본을 **학습셋에서 뽑아** 씁니다. 분포가 다른 일반 문서로 보정하면 **접는 지점이 어긋나** 품질이 떨어집니다.

② 무엇을 남기나

라우터와 출력층은 접지 않는다

전문가 혼합에서 **어느 전문가에게 보낼지 정하는 로짓**은 예민합니다. 그래서 라우터와 출력층은 **양자화에서 제외**합니다 — 나머지를 접어 21 GB 를 만듭니다.

③ 배포 형태

어댑터를 베이스 위에 얹는다

베이스 21 GB 는 그대로 두고 **어댑터(수백 MB)만** 바꿔 끼웁니다. **다시 양자화할 필요가 없어** 배포도 되돌리기도 가볍고, 재양자화에서 품질이 흔들릴 여지도 없습니다.

④ 확인

전문가가 빠지지 않았는지 본다

4비트면 산출물이 원본의 **3분의 1 이하**로 떨어집니다. 비슷하게 남아 있으면 전문가 가중치가 접히지 않은 것이라, **크기만으로 1차 확인**이 됩니다.

05 · 무엇을 보는가

고장 메커니즘 지도 — 신호 하나로는 갈리지 않습니다

온도가 올랐다는 사실만으로는 후보가 **넷**입니다. 그래서 가르치는 것은 「온도 상승 → 베어링」 같은 짝이 아니라 **어느 신호가 함께 움직이고, 어느 것이 먼저 움직이는가**입니다. 아래가 그 지도입니다.

회전기계 계열 — 여덟

고장	온도	진동	소음	전력	보조 유체	징후 → 고장
베어링 마모	+28°C	**4.2배**	+14 dB	+8%	—	125일
정렬 불량	+12°C	3.0배	+8 dB	+15%	—	200일
언밸런스	+4°C	3.4배	+6 dB	+3%	—	250일
윤활 부족	+34°C	2.2배	+11 dB	+10%	윤활 −62%	**83일**(가장 빠름)
냉각 불량	**+42°C**	1.1배	+3 dB	+6%	냉각 −58%	100일
전기적 결함	+30°C	1.6배	+5 dB	**+32%**	—	143일
캐비테이션	+6°C	3.6배	**+18 dB**	**−12%**	—	111일
파울링	+36°C	1.1배	+2 dB	+22%	냉각 −30%	167일

유체 계열 — 셋

고장	토출압	하류압	차압	가동률	온도	징후 → 고장
누설	−18%	**−34%**	—	**+0.45**	+16°C	91일
막힘	**+16%**	−29%	**3.6배**	+0.12	+24°C	167일
밸브 고착	−5%	−7%	1.5배	—	+10°C	125일

고장 11종 × 설비 18종

원심펌프 · 왕복동/스크류 압축기 · 송풍기 · 냉동기 · 냉각탑팬 · 감속기 · 유도전동기 · 유압유닛 · 사출성형기 · 프레스 · CNC 스핀들 · 컨베이어 구동부 · 교반기 · 원심분리기 · 열교환기 순환펌프 · 보일러 급수펌프 · 진공펌프. **설비마다 가능한 고장이 다릅니다** — 회전 여섯은 18종 전부, 유체 셋은 11종, 캐비테이션과 파울링은 각각 4종입니다.

「밸브 고착」은 평균으로 안 잡힙니다

압력 평균이 **8.29 bar**(정상 8.49)로 거의 같은데 **표준편차가 0.309**(정상 0.104)입니다. 증상이 값이 아니라 **흔들림 자체**라, 평균만 보는 감시로는 지나갑니다 — **산포를 함께 보는 이유**입니다.

06 · 가르는 법

같은 두 신호라도 «순서» 가 다르면 다른 고장입니다

현장에서 원인을 좁히는 힘은 값의 크기가 아니라 **무엇이 먼저 움직였는가**에서 나옵니다. 그래서 학습 데이터가 스냅샷이 아니라 **시간 축을 가진 궤적**입니다.

고장	먼저 오는 신호	뒤따르는 신호
베어링 마모	**진동** 상승	온도가 따라 오름 · 결함 주파수 고조파
윤활 부족	**윤활유 유량** 저하	마찰열로 온도 급상승
냉각 불량	**냉각수 유량** 저하	온도만 오름(진동은 정상)
정렬 불량	2× 성분 우세 · **축방향** 진동	전력 증가
언밸런스	1× 성분 우세 · **반경방향**	온도 변화는 미미
전기적 결함	**전류 불평형**	소비전력 · 온도 동반 상승
캐비테이션	**광대역 소음** 급증	토출 저하 → **전력 감소**
파울링	전열면 오염 → **차압** 상승	온도 상승 동반

온도가 올랐을 때 — 넷을 가르는 규칙

진동이 «함께» 오르면

베어링 마모 · 정렬 불량 · 언밸런스 쪽입니다. 그다음은 1×인가 2×인가, 반경방향인가 축방향인가로 갈립니다.

온도만 오르면

냉각 불량 · 파울링 쪽입니다. 냉각수 유량이 먼저 떨어졌는지, 차압이 올랐는지를 봅니다.

윤활유량이 «먼저» 떨어졌으면

윤활 부족입니다. 온도 급상승은 결과이지 원인이 아닙니다.

전력이 «내려가는» 유일한 모드

캐비테이션입니다. 다른 모드는 전부 전력이 오릅니다 — 이 하나만으로 확정됩니다.

유체 셋을 가르는 법

누설	토출압과 하류압이 **함께** 떨어지되 하류가 더 떨어져 격차가 벌어집니다. 그리고 **가동률이 1.0 에 붙습니다** — 새는 만큼 채우느라 쉬지 못합니다. 차압은 그대로입니다
막힘	토출압이 오히려 **오릅니다**(상류에 쌓입니다). **차압 3.6배가 1차 신호**입니다 — 파울링과 닮았지만 온도 상승이 덜하고 차압이 더 가파릅니다
밸브 고착	평균은 정상 범위 안입니다. **산포**가 신호입니다

근거는 문장으로 나옵니다. 학습 데이터의 답이 판정 한 줄로 끝나지 않고, 그 판정을 뒷받침하는 문장을 함께 냅니다 — 「온도가 D+11에 먼저 오르고 진동은 D+19에 뒤따랐다(**8일 지연**) — 열적 원인이 선행」, 「윤활유량이 78.3% → 49.8% 로 **0.86%/일** 하락 — 지속적 소모 또는 누유」, 「최근 14일 진동 추세 **+0.309 mm/s·일** — 가속 진행」. 수치와 기간이 붙어 있어야 **사람이 검증할 수 있습니다.**

07 · 데이터를 고르는 법

«정답이 우리 밖에서 오는가» 를 먼저 묻습니다

데이터는 많이 모으는 것보다 **고르는 기준**이 먼저입니다. 우리가 정한 규칙으로 문제를 만들고 그 규칙으로 채점하면 **순환**입니다 — 그 고리를 끊는 데이터가 가장 값어치 있습니다. 그래서 세 가지를 묻습니다.

기준 ①

정답이 우리 밖에서 오는가

운영 기록 · 정비 이력처럼 **우리가 만들지 않은 라벨**이 붙어 있는가. 여기서 오는 데이터만 「우리 가정이 맞았는지」를 되물을 수 있습니다.

기준 ②

실험실 고장인가 현장 고장인가

인위로 만든 결함·가속 시험과, **운행 중 실제로 난 고장**은 다릅니다. 현장 데이터에는 결측 · 정비 이벤트 · 운전 조건 변화가 함께 들어 있습니다 — 그것까지가 현실입니다.

기준 ③

가정이 덜 끼는 형태인가

고주파 진동 원파형은 특징값으로 바꿔야 쓰는데 **그 변환에 우리 가정이 들어갑니다.** 압력 · 온도 · 전류처럼 **이미 공학 단위**인 신호는 그 가정이 덜 낍니다.

거르는 기준

센서와 라벨 사이에 «인과» 가 있는가

물리적으로 불가능한 값이 섞여 있거나, 센서는 멀쩡한데 라벨만 「곧 고장」인 데이터가 있습니다. **인과가 없는 데이터는 크기와 무관하게 쓰지 않습니다** — 학습하면 근거 없이 단정하는 습관이 붙습니다.

공개 데이터의 라벨도 그대로 믿지 않습니다

받아서 열어 보고 **직접 확인합니다.** 실제로 「정상」으로 분류된 구간 중 **72개**가 고장 구간과 통계적으로 구분되지 않아 평가에서 제외한 적이 있고, 정상 난이도 문항 33개 중 **11개가 두 시간 내내 값이 얼어붙은 창** 이면서 「정상」으로 라벨돼 있던 것도 찾아냈습니다. **센서 고착을 설비 고장으로 읽거나 그 반대로 넘기는 것**은 현장에서 큰 오류원이라, 이런 창은 평가에서 따로 다룹니다.

> **합성으로 가르치고, 실측으로 되묻습니다.** 시뮬레이터가 만든 데이터로 시뮬레이터가 만든 모델을 재면 같은 가정을 두 번 쓰는 것입니다. 그래서 합성 데이터는 **가르치는 데** 쓰고, **「맞았는지」를 묻는 자리에는 정답이 바깥에서 오는 데이터**를 놓습니다.

08 · 확보한 데이터

바깥에서 온 정답 — 무엇을 덮고 있나

위 기준을 통과해 **실제로 받아 열어 본** 공개 데이터입니다. 각각이 **커버리지의 어느 구멍을 메우는지**를 보고 골랐습니다 — 크기가 아니라 **덮는 자리**가 기준입니다.

데이터	성격	무엇을 덮나
지하철 공기압축기 MetroPT-3 · CC BY 4.0	**실측**	운행 중인 차량의 공기압축기 **6개월 · 151만 행**(10초 간격). 아날로그 7 + 디지털 8. **고장 4건이 운영 기록으로 문서화**돼 있습니다 — 정답이 바깥에서 오는 대표 사례
유압 시험대 UCI · ZeMA	**실측**	60초 부하 사이클 **2,205회**, 센서 17개. 쿨러 · 밸브 · 펌프 내부 누설 · 어큐뮬레이터를 **단계별로 열화시켜 라벨을 확정**했습니다. 우리 유체 계열과 **거의 그대로 맞물립니다**
화학 공정 Tennessee Eastman	시뮬	공정 고장 **21종 · 52변수**. 공정 계열 커버리지를 통째로 채웁니다
순환펌프 시험대 SKAB	**실측**	밸브 조작·누설을 실제로 일으킨 기록. **진동과 압력을 같이 재는 조합**이라 값어치가 큽니다
반도체 공정 SECOM	실측	**590 센서** 합불 — 고차원 공정 감시
함정 추진계통 Naval Propulsion	시뮬	가스터빈 추진계통 **11,934** 인스턴스. 열화계수가 **0.001 단위 연속값 라벨**입니다 — 정상/이상 이분법에 없던 형태
발전 가스터빈 UCI · 5년치	**실측**	11 센서 + CO·NOx 배출. **배출가스로 연소 상태를 읽는 축**은 온도·진동에 없던 것입니다
센서 드리프트	실측	**설비가 아니라 계측기가 고장 나는 경우** — 현장에서 자주 오진되는 자리
강판 표면 결함	실측	결함 **7종** — 재료 계열

업종으로 보면

철도·공압 · 유압 · 화학 공정 · 펌프 계통 · 반도체 · 조선/방산 추진 · 발전(가스터빈) · 계측기 · 금속 재료. **업종마다 「무엇이 먼저 말하는가」가 다르고**, 그 축을 하나씩 확보하는 것이 목표입니다.

다음으로 붙이는 것

풍력(기어박스·발전기 SCADA + 고장 이력) · 기어박스 다중 클래스 · 베어링 **수명 전체 궤적**. 마지막 것은 **열화 속도를 실측으로 보정**하는 데 씁니다 — 모델을 고치기 전에 데이터를 고칩니다.

09 · 시뮬레이터

물리를 먼저 정하고, 라벨을 «유도» 합니다

산업 문답은 인터넷에 없으므로 만들어야 합니다. 핵심은 **만드는 순서**입니다 — 라벨을 먼저 찍고 신호를 붙이면 **센서와 정답 사이에 인과가 없어집니다.** 그래서 반대로 만듭니다.

① **열화 상태를 굴린다**	잠재 상태 `d` (0~1)를 시간에 따라 진행시킵니다. 고장모드마다 속도가 다르고, **진행할수록 가속**합니다
② **센서는 관측치다**	센서값은 그 상태의 **잡음 섞인 관측**입니다. 어느 센서가 먼저 움직이는지는 **실제 고장 물리**를 따릅니다
③ **라벨을 유도한다**	정답을 `d` 에서 **끌어냅니다.** 그래서 센서 → 정답 추론이 **원리적으로 가능**합니다
④ **범위를 생성식이 보장**	사후에 잘라 맞추지 않습니다 — 생성식 자체가 **물리적으로 가능한 범위** 안에서만 값을 냅니다
⑤ **고장에서 끊는다**	`d=1.0` 에서 궤적을 멈춥니다. **고장 나면 정비하지 계속 돌리지 않습니다**

인과가 실제로 생겼는지 확인합니다 — 20,000행 실측

모드	온도	진동	전력	보조 유량	읽는 법
베어링 마모	1.33배	**3.40배**	1.05배	—	진동이 압도적
윤활 부족	1.47배	2.06배	1.13배	**0.39배**	유량이 먼저 꺼짐
냉각 불량	**1.53배**	1.13배	1.11배	0.52배	진동 없이 온도만
캐비테이션	1.29배	2.67배	**0.63배**	—	전력이 «내려감»
파울링	1.36배	0.91배	**1.51배**	0.85배	진동 없이 전력이 오름

진동만으로는 베어링 마모와 냉각 불량이 안 갈립니다. 온도 · 전력 · 유량을 함께 봐야 갈립니다 — 생성한 데이터에서 그 구분이 실제로 성립하는지를 **수치로 확인**하고 넘어갑니다. 물리 범위 위반은 **0건**입니다.

물리는 «정확성», 모델은 «표현»

시뮬레이터가 만든 문장은 골격이 같아서, 그대로 두면 모델이 **템플릿을 외웁니다.** 그래서 표현만 다양화합니다 — 질문 표현 **87종** · 판정 표현 **50종**. **무엇이 참인지는 물리가**, 어떻게 말하는지만 모델이 정합니다.

다른 모델에 기대지 않습니다

생성물 **215,021건(93%)**이 **다른 모델 없이** 만들어집니다 — 물리 · 수식 · 규격 원문 · 도구 스키마에서 나옵니다. 그래서 **천장이 남의 모델 성능에 걸리지 않습니다.**

10 · 모른다고 말하기

근거가 없으면 «판단 보류» 를 정답으로 둡니다

산업 AI 에서 가장 위험한 실패는 틀리는 것이 아니라 **근거 없이 단정하는 것**입니다. 그래서 학습 단계에서부터 **「아직 모른다」를 정답으로 가르칩니다.**

시뮬레이터는 정답을 알지만 말하지 않습니다

생성기는 그 궤적의 고장모드를 압니다. 그런데 **센서가 아직 안 움직였으면 라벨을 붙이지 않습니다.** 초기 열화는 센서로 진단이 불가능한 것이 맞고, 그런데도 정답을 붙이면 **「근거 없이 확신하는」 습관**을 가르치게 됩니다.

판정 기준을 수치로 고정했습니다

근거 두 개 이상, 또는 **근거 하나 + 열화도 0.45 이상**일 때만 확정 진단을 냅니다. 그 아래는 **판단 보류**이고, 보류도 정답입니다.

그래서 데이터의 3분의 1이 «확정하지 않는» 답입니다

센서 스냅샷	구성
확정 진단	**895**
판단 보류	**662** — 「지금 보이는 것으로는 아직 갈리지 않습니다」
정상	500

이벤트 체인에서도 **확정 진단은 64.4%**이고 나머지는 보류·정상입니다. 보류 문구도 **30종**으로 다양화해, 모델이 「모른다」를 **한 가지 상투어**로 외우지 않게 했습니다.

「0」과 「측정 안 됨」도 같은 문제입니다. 「이번 주 작업지시 0건」은 측정이고 「연결이 끊겨 못 읽었다」는 측정이 아닙니다. 뒤쪽을 앞쪽처럼 보고하면 **아무도 들여다보지 않은 설비를 운영자가 정상으로 믿습니다.** 학습 데이터에 값이 비어 있는 맥락을 일부러 넣고, 그 경우 **「측정 안 됨」이라고 말하는 답**을 정답으로 두었습니다.

사람이 검증할 수 있는 형태로 답합니다

확정 진단이든 보류든, 답에는 **어느 신호가 · 얼마에서 얼마로 · 며칠에 걸쳐** 움직였는지가 붙습니다. 「온도를 확인하십시오」는 근거가 아니고 **「온도가 62→89°C 로 올랐습니다」가 근거**입니다 — 평가에서도 **신호 이름 뒤에 수치가 있는지**를 기계로 셉니다(16장).

11 · 정답의 출처

계산 · 규격 · 절차 · 도구 — 정답이 각각 어디서 오나

데이터 갈래마다 **정답을 보증하는 주체가 다릅니다.** 그 주체가 분명한 것만 학습에 넣습니다.

갈래	규모	정답을 누가 보증하나
산업 계산	**8,546**	**수식**. 난수로 문항을 만들고 공식이 정답을 냅니다 — 다른 모델이 필요 없고 **상한도 없습니다**
규격 · 법령	**1,326**	**조문 원문**. 산업안전보건법 조문 184개와 해외 산업안전 규정 조문 479개에서, **본문에 있는 것만**으로 답을 만듭니다
절차(SOP)	**1,087**	**먼저 고정한 절차서**. SOP 문서 106개를 만들어 **고정한 뒤** 거기서만 답을 끌어냅니다
도구 호출	**1,059**	**스키마**. 필수 인자가 선언돼 있어 정답이 보장됩니다
현장 용어	140	**수기 사전** 28종 — 영한 병기와 이표기(트립 / Trip / tripping)를 함께 담습니다

계산 — 30 계열, 그중 일곱은 학습에서 뺐습니다

열·유체(LMTD · 전열량 · 레이놀즈수 · 관마찰 · 펌프 축동력 · 오리피스 · NPSH · 상사법칙), 회전기계(베어링 결함 주파수 · 수명 · 벨트 속도 · 감속기 토크 · 축 비틀림), 용접·기계(입열량 · 볼트 토크 · 공차 · 필릿 용접 · 키 전단), 압력용기·배관(배관 두께 · 원통 동체 · 경판 · 최고사용압력 · 안전밸브 면적), 품질·전기(공정능력 · OEE · 전동기 전류 · 열팽창).

홀드아웃이 «같은 계열 안» 에 있습니다

원통 동체의 **원주 방향** 식은 가르치고 **길이 방향** 식은 뺐습니다. 공식을 외운 것인지, **상황에 맞는 식을 골라 세우는 것**인지가 그 한 칸에서 갈립니다.

문항의 «수치» 도 현실적으로 맞춥니다

볼트에 항복을 넘는 하중, 베어링 수명 256시간 같은 값이 나오면 문항 자체가 비현실적입니다. 허용 응력비와 하중비를 **현장 범위로 교정**해 생성합니다.

도구 호출은 네 가지 형태로 가르칩니다

도구 선택	여러 도구 중 **어느 것을 부를지**
필수 인자 채우기	현장에서 가장 자주 틀리는 것이 **필수 누락과 타입 오류**입니다
실패 처리	도구가 **판독 거부**를 돌려줬을 때 **값을 지어내지 않기**
도구 설명	그 도구가 무엇을 하는지 사람 말로 설명하기

계기 판독이 대표적입니다. 흐린 사진에서 여러 번 읽은 값이 다르면 도구가 **「읽을 수 없음」**을 돌려줍니다. 그때 **숫자를 추측해 보고하지 않도록** 학습합니다 — 현장에서 지어낸 계기값은 곧 **잘못된 조치**로 이어집니다.

12 · 학습 비중

무엇을 얼마나 넣는가가 곧 능력의 모양입니다

데이터를 많이 가진 것과 **잘 섞은 것**은 다릅니다. 한 갈래가 압도하면 나머지 능력이 **조용히 희석**됩니다 — 그래서 갈래마다 **상한**을 두고 섞습니다.

최신 조립본의 비중

이벤트 체인(진단 궤적)	**57%**
서술형 도메인	22%
산업 계산	12%
센서 스냅샷	7%
규격 · 용어 · 절차	4%
정체성 · 가드레일	**2%**

이벤트 체인은 **20만 건을 만들어 두고 4만 건만** 투입합니다. 상한이 없으면 그 하나가 전체의 87% 를 차지해 **나머지를 전부 덮습니다.**

실제로 겪은 일

서술형만 크게 넣었더니 계산이 흔들렸습니다. 베어링 결함 주파수를 계산해야 할 자리에서 **「고조파를 확인한다」는 서술로 빠졌습니다.** 학습 데이터에 계산 과정이 한 건도 없었기 때문입니다. **계산 벤치가 없었으면 못 봤을 일**이라, 그 뒤로 조립 첫 단계가 **갈래별 건수 확인**입니다.

품질 필터는 «생성 직후» 에 겁니다

너무 짧은 답	폐기
한국어가 모자란 답	폐기
다른 언어 누출	폐기 — 한글 안의 한자 병기(온도(溫度))는 오탐하지 않습니다

학습 직전에 거르는 것보다 **만든 직후에 거르는 편이 싸고**, 무엇이 왜 빠졌는지도 그때가 분명합니다.

프롬프트 — 길게 쓰면 도구를 안 부릅니다

같은 모델 · 같은 문항으로 **시스템 프롬프트만 바꿔** 도구 호출을 재 봤습니다. 결과의 순서입니다 —

프롬프트	결과
2줄(145자)	**가장 좋았습니다**
압축판(1,189자)	근소한 차이
2줄 + 도구 이름 57개 나열	뚜렷하게 나빠졌습니다
긴 본문(5,488자)	가장 낮았습니다

규칙을 «덜» 적어서 좋아진 것이 아닙니다 — 긴 본문이 도구 호출 신호를 묻습니다. 도구 «이름» 을 본문에 나열하는 것도 역효과입니다(스키마로만 줘도 모델은 압니다). 그래서 운영 규칙이 하나 있습니다 — **규칙을 하나 더 넣기 전에 벤치로 잰다.** 넣어서 점수가 떨어지면 그 규칙은 **이득보다 비용이 큽니다.**

13 · 정체성과 범위

정체성은 프롬프트가 아니라 «학습» 으로 박습니다

프롬프트는 **잊히거나, 잘리거나, 사용자가 덮어씁니다.** 그때 정체성이 무너지면 고객사에 설명할 수 없습니다. 그래서 프롬프트에 기대지 않는 쪽을 택했습니다.

세 조건으로 함께 학습합니다

같은 문답을 ① **자사 프롬프트** ② **프롬프트 없음** ③ **남의 프롬프트** (타사 어시스턴트 정체성 문구 포함) 세 조건으로 가르칩니다. 그래서 **프롬프트가 없어도, 남의 프롬프트로 덮어써도** 정체성을 유지합니다.

서빙에서 프롬프트는 «보강» 입니다

프롬프트가 빠져도 정체성이 유지돼야 정상이고, **무너지면 학습이 잘못된 것**입니다 — 이것을 **검증 항목**으로 두고 프롬프트 있음/없음 **두 조건 모두**로 잽니다.

기본 시스템 프롬프트는 네 조항, 209자입니다

조항	왜 그 자리에 있나
① **정체성**	KOPENS 가 만든 산업 AI 어시스턴트라는 것 — 기반 정보가 새면 고객사에 설명이 안 됩니다
② **근거와 «모른다»**	설비 · 공정 · 품질 · 안전에 특화하고, **근거를 들어 답하며 확실하지 않으면 추측하지 않습니다**
③ **언어**	한국어가 기본이고 영어로도 답합니다. 다국어 기반은 **입력 언어를 따라가는 경향**이 있어 명시합니다
④ **범위**	정치 · 외교 등은 다루지 않습니다 — **검열이 아니라 범위 설정**입니다. 설비 진단에 정치 논평이 필요하지 않습니다

순서에도 뜻이 있습니다. 그리고 정체성·가드레일 데이터는 전체의 **2%**만 넣습니다 — 과하면 일반 성능이 무너집니다. **범위를 정하는 데 학습을 얼마나 쓸지도 설계**입니다.

정본은 «하나» 입니다

시스템 프롬프트 문자열 하나를 유일한 원본으로 두고 **데이터 생성 · 평가 · 서빙이 전부 그것을 참조**합니다. 사본을 만들면 반드시 갈라집니다 — 실제로 빌드 스크립트에 사본이 생겨 표기가 어긋난 적이 있습니다.

평가도 «배포에 쓰는» 프롬프트로

프롬프트가 결과를 좌우하므로, 다른 프롬프트로 잰 점수는 **배포 환경의 점수가 아닙니다.** 그래서 평가와 서빙이 같은 정본을 씁니다.

14 · 검색

산업 RAG 는 «검색 품질» 이 답변 품질의 상한입니다

아무리 좋은 모델도 **틀린 근거를 받으면 틀린 답**을 냅니다. 범용 임베딩은 「베어링 마모로 인한 온도 상승」과 「냉각수 부족으로 인한 온도 상승」을 **거의 같은 문장**으로 봅니다 — 현장에서는 완전히 다른 조치인데도 그렇습니다. 그래서 검색 모델을 따로 학습시켰습니다.

임베딩 4B
«헷갈리게 만든 오답» 을 따로 넣습니다

같은 배치 안의 다른 항목을 부정 예로 쓰는 대조학습에, **비슷해 보이지만 답이 아닌 문서**를 일부러 섞습니다. 가려내는 힘이 거기서 나옵니다.

리랭커 0.6B
«순위를 매기는 일» 자체를 배웁니다

후보 목록에서 정답을 고르는 방식으로 학습합니다. 질의와 문서를 **함께 읽는** 교차 인코더라 임베딩이 놓친 **미묘한 차이**를 잡습니다.

비대칭 인코딩 — 질의에만 지시문을 붙입니다

질의에는 「무엇을 찾는가」를 알려 주는 지시문을 붙이고 **문서에는 붙이지 않습니다.** 문서는 한 번 색인하면 오래 쓰는 자산이라, 문서 쪽에 지시문을 넣으면 **지시문이 바뀔 때마다 전량 재색인**입니다.

차원을 «계약» 으로 고정했습니다

> **판올림해도 벡터 색인을 다시 만들지 않습니다.** 검색 모델의 출력 차원(2,560)을 판 사이에 고정했습니다. 수천만 조각을 다시 임베딩해야 하는 판올림은 **결국 미뤄지고**, 미뤄진 판올림은 없는 것과 같습니다 — 그래서 차원을 성능 여지보다 **운영 가능성** 쪽에 맞췄습니다.

평가에서 «누출» 을 먼저 막습니다

묶음 단위 분할	질문이나 정답 문서를 **공유하는 항목**은 한 묶음으로 보고 묶음째 학습·평가로 가릅니다. 행 단위로 무작위 분할하면 **같은 문서가 양쪽에 걸쳐** 점수만 오릅니다
설비 단위 분할	고장 진단은 **설비 단위로** 뺍니다 — **학습에 없던 설비**로 평가합니다
출처별로 나눠 보기	고장 진단 · 절차서 · 규격을 **따로** 봅니다. 전체 평균은 비중이 큰 쪽에 가립니다
말투 강건성	같은 질문을 **현장 말투로 바꿔** 다시 잽니다 — 점수가 무너지면 형식을 외운 것입니다

15 · 시계열 트랙

파형은 «작은 모델» 로 봅니다 — 현장마다 하나씩

언어모델은 **왜 그런지 설명**하는 데 강합니다. 초 단위로 들어오는 파형에서 **무엇이 이상한지 찾는** 일은 성질이 다르고, 필요한 크기도 다릅니다.

백만 파라미터급이 «강점» 입니다

GPU 없이 돌아 **엣지 박스에 얹을 수 있고**, 자산마다 따로 맞춰도 비용이 거의 없습니다. **설비 1만 대에 모델 1만 개**를 두는 것이 현실적인 크기입니다 — 큰 모델로는 성립하지 않는 운영 형태입니다.

설비 이름이 아니라 메커니즘을 가르칩니다

베어링 마모는 **펌프에서든 감속기에서든** 진동이 먼저 오르고 온도가 따라옵니다. 그래서 **고객 데이터 없이도** 공개 실측과 물리 시뮬레이터로 시작할 수 있습니다.

산업 신호가 일반 시계열과 다른 네 가지

운전과 정지가 섞인다	**정지 구간의 0 은 이상이 아닙니다.** 그것을 이상으로 세면 경보가 쏟아집니다
부하에 따라 정상이 바뀐다	무부하 4 A 와 부하 7 A 가 **둘 다 정상**입니다 — 절대 문턱값 하나로는 풀리지 않습니다
계측 결측이 흔하다	**센서 두절과 값 0 은 다릅니다.** 그래서 결측을 **따로 표시**해 모델에 알려 줍니다
고장은 드물다	6개월에 4건 — 전체의 2% 수준입니다. **희귀 사건을 재는 법**이 따로 필요합니다

쓰임새 셋

① 상시 감시

창 단위로 정상/이상 점수를 내고, 임계를 넘으면 알립니다.

② 점검 순서 좁히기

「온도가 올랐는데 무엇부터 보나」에 **순서**를 제시합니다.

③ 오경보 억제

부하가 걸리면 전류·압력이 올라 고장처럼 보입니다. **그것을 정상으로 읽는 것**이 이 축의 절반입니다.

④ 이상의 «지문» 을 남긴다

파형에서 뽑은 특징을 저장해 두면 **「이번 이상이 과거 어느 고장과 닮았나」**에 답할 수 있습니다 — 다음 판에서 다룹니다.

두 트랙이 한 현장을 봅니다. 파형에서 **무엇이** 이상한지는 시계열 쪽이, **왜** 그런지는 언어모델이 말합니다. 그리고 **통계로 충분한 구간은 모델로 풀지 않습니다** — 평소 기준선 대비 편차만으로 갈리는 구간이 있고, 그 자리에 모델을 넣는 것은 비용만 늘립니다.

16 · 평가

심판 모델을 쓰지 않는다 — 재는 사람이 바뀌어도 값이 같게

산업 AI 에는 리더보드가 없습니다. 그래서 계측기를 직접 만들었는데, 원칙이 하나입니다 — **다른 모델에게 채점을 맡기지 않습니다.** 정답이 수식 · 스키마 · 운영 기록에서 나오므로 채점은 기계가 하고, **같은 입력이면 누가 언제 돌려도 같은 점수**가 나옵니다.

축	문항	무엇을 어떻게 재나
산업 계산	**600**	공학 수식 **30 계열**(베어링 결함 주파수 · 수명 · NPSH · ASME 동체 두께 · 오리피스 유량 · OEE · 공정능력 · 용접 입열량 …). 문항은 난수로 만들고 정답은 공식이 계산합니다. **상대오차 1% 이내**면 정답
도구 호출	22	가동 플랫폼의 **실제 도구 57종** 스키마를 맥락으로 줍니다. 도구 이름은 집합 대조, 인자는 그 도구가 선언한 JSON Schema(필수 항목 · 정규식 · enum)로 검증, 다단계는 **순서**를 봅니다
근거 충실성	8	실제 분석 프롬프트를 맥락으로 주고 **주어진 사실만** 썼는지. 특히 맥락에 **「데이터 없음」이 있을 때 그것을 밝히는지**
현장 진단	**158**	실제 압축기 6개월 실측 — **다섯 중 유일하게 정답이 우리 밖에서** 옵니다(17장)
검색	277	임베딩은 코퍼스 전체에서 찾기(Recall@1/5/10 · MRR@10), 리랭커는 후보 8개 중 정답 고르기

7 계열은 학습에서 아예 뺐습니다

30 계열 중 **일곱**은 훈련 데이터에 넣지 않았습니다. 넣고 재면 **외운 것**을 재게 됩니다. 빼고 재야 「처음 보는 수식도 세울 줄 아는가」가 보입니다.

검색 평가는 «누출 0» 을 확인했습니다

질문과 정답 문서를 공유하는 항목은 **묶음 단위로** 갈라 학습과 평가에 나눴습니다. 같은 문서가 양쪽에 걸치면 점수는 오르고 실력은 그대로입니다.

「0」과 「모름」은 평가에서도 갈랍니다. 학습 데이터에 **값이 비어 있는 맥락**을 일부러 넣고 그 경우 **「측정 안 됨」이라고 말하는 답**을 정답으로 두었으며, 근거 충실성 축이 그것을 그대로 잽니다.

채점이 «기계화» 되어 있다는 말의 뜻

축	규칙
산업 계산	답에서 **「정답: …」 줄의 마지막 것**을 쓰고, 없으면 본문 뒤에서부터 파싱되는 첫 수를 씁니다(모델이 중간 계산을 여러 번 쓰므로 **마지막이 결론**입니다). 비교는 $\lvert 얻은값 - 정답 \rvert / \lvert 정답 \rvert \le 0.01$, 정답이 0 이면 절대값 비교. **단위는 파싱하지 않습니다** — 「Hz」와 「hz」, 「mm/s」와 「mm·s⁻¹」를 가리려 들면 표기 차이가 곧 채점 오류가 됩니다. 문항이 단위를 지정하고 숫자만 봅니다
도구 호출	도구명은 선언된 집합과 대조하고, 인자는 **세 가지를 각각** 셉니다 — 필수 누락 · 선언되지 않은 인자 · 형식 위반. 정규식은 실제로 매칭합니다: 날짜에 `^\d{4}-\d{2}-\d{2}$` 가 걸려 있으면 `2026-09-19` 는 통과, `2026/09/19` 는 위반입니다. **운영 플랫폼이 같은 형식을 강제하므로, 이 검증을 통과 못 하는 호출은 실제로 400 을 받습니다**
근거 인용	「이름만 나열하는 것은 근거가 아니다」를 규칙으로 옮겼습니다. 신호 이름 **뒤 80자 안에 수치가 있는지**를 봅니다 — 「온도를 확인하십시오」는 0점, 「온도가 62→89°C 로 올랐습니다」는 1점. 문장이나 주장이 아니라 **신호 단위**로 셉니다

검토자가 «스스로» 확인하는 법 — 우리 말을 믿지 않아도 됩니다

무엇을	어떻게
학습한 판이 실제로 서빙되나	기반 모델과 학습본을 **다른 이름으로 동시에 노출**하고 같은 질문을 둘에 던집니다. **응답이 사실상 같으면 학습이 반영되지 않은 것**입니다
평가에 학습이 새지 않았나	평가 질문·정답 문서를 학습셋과 대조해 완전 일치를 셉니다. 질문이나 문서를 공유하는 항목은 **묶음 단위로** 갈라야 합니다
형식을 외운 것인가	같은 질문을 **현장 말투로 바꿔** 다시 잽니다. 점수가 무너지면 내용이 아니라 형식을 외운 것입니다

문항을 늘려도 정보는 늘지 않습니다. 현장 진단의 고장은 **독립 사건 넷**입니다. 창을 겹쳐 자르면 240문항이 나와 오차범위가 «좁아» 보이지만 같은 83시간을 다시 자른 것입니다 — **정밀해 보이게 만드는 것과 정확해지는 것은 다릅니다.** 그래서 겹치지 않게 자른 **158**을 씁니다. 점수는 **지금 제품 경로에서 다시 재는 중**이고, 끝나면 **같은 표로** 채웁니다.

17 · 현장 진단

양치기 소년을 만들지 않으려면 — 거짓 경보를 따로 센다

다섯 축 중 유일하게 **정답이 바깥에서 오는** 평가입니다. 실제로 돌던 압축기의 **6개월 실측**과 그 기간의 **운영 기록에 남은 고장**을 정답으로 씁니다. 우리가 만든 규칙이 아니라 **실제로 일어난 일**이 채점표입니다.

158 문항의 구성 — 「정상」을 세 종류로 나눴습니다

구간	문항	왜 이렇게 나눴나
고장	**44**	운영 기록에 고장으로 남은 구간. 여기서 놓치면 **미탐지**입니다
고부하 정상 (어려운 정상)	**40**	🔴 **비정상처럼 보이지만 정상인** 구간. 온도도 진동도 올라가 있지만 설비는 멀쩡합니다. 여기서 경보를 울리면 **거짓 경보**입니다
평범한 정상	60	조용한 구간. 여기서 울리면 변명의 여지가 없습니다
고장 직전	14	**채점하지 않습니다.** 「어디까지 미리 보는가」를 관찰만 합니다 — 정답을 정할 수 없는 구간에 점수를 매기면 그 점수가 거짓이 됩니다

왜 거짓 경보를 미탐지와 갈라 세나

둘을 합쳐 «정확도» 하나로 보고하면, **모든 구간에 경보를 울리는 모델**이 고장을 다 잡았다는 이유로 좋은 점수를 받습니다. 현장에서 그런 조수는 **일주일이면 꺼집니다.** 그리고 꺼진 조수는 아무것도 잡지 못합니다.

채점은 한 줄로

모델의 답에서 **정상/이상 판정 한 줄**을 뽑아 기록과 기계 대조합니다. 서술을 사람이 읽고 점수를 주지 않습니다 — 읽는 사람이 바뀌면 점수가 바뀌기 때문입니다.

🔴 「어려운 정상」을 다시 고른 일 — 채점기가 틀렸던 이야기

처음에는 **모터 전류 상위 구간**을 어려운 정상으로 골랐습니다. 그런데 그렇게 고른 40개가 **고장 구간과 통계적으로 구분되지 않았습니다** — 전류 · 흡입밸브 가동률 · 압력이 사실상 같았습니다. **운영 기록에 없는 이상 구간**이었던 것입니다. 그대로 뒀다면 징후를 «옳게» 읽은 모델이 40문항을 틀립니다 — 모델이 아니라 **채점기가 틀린** 상태입니다.

그래서 기준을 바꿨습니다. 정상 운전의 압축기는 **주기적으로 돌고 섭니다.** 연속 운전이면서 특정 압력 신호가 사라진 창은 아예 뺐고(72개), 남은 「어려운 정상」은 **전류가 평범한 정상의 2.3배**로 높지만 압축기가 정상적으로 순환하고 압력도 살아 있는 구간입니다. **「수치가 높다」와 「고장이다」를 구분해야** 풀립니다. 검토자에게 이 대목이 시사하는 바가 있습니다 — **공개 데이터의 라벨을 그대로 믿으면 안 됩니다.**

데이터를 이렇게 골랐습니다

정답이 우리 밖에서 옵니다(운영사 고장 기록이 라벨). **실험실이 아니라 운행 중**에 실제로 난 고장입니다 — 공개 데이터 상당수는 고장을 인위로 만들거나 가속 시험으로 태운 것입니다. 신호가 **공학 단위**(압력 · 온도 · 전류)라 특징 추출 단계에서 우리 가정이 끼어들지 않습니다.

구간은 이렇게 잘랐습니다

2시간 창으로 **겹치지 않게** 자르고(겹치면 문항만 늘고 정보는 안 늡니다), 창마다 15분 단위 8행으로 요약해 줍니다. 수집이 끊겨 8행이 안 되는 창, **정비 전후 6시간**, 고장 구간에 걸치기만 한 창은 버립니다.

답의 형식을 정해 두었습니다

```
판정: 정상 | 이상
```

채점기는 이 줄을 정규식으로 뽑습니다(전각 콜론도 받습니다). 형식이 없으면 앞 200자 안에서 둘 중 하나만 나타날 때에 한해 인정하고, 둘 다 나오거나 둘 다 없으면 **형식 위반으로 따로 셉니다** — 그것은 판단력 문제가 아니라 **지시 따르기** 문제라 오답과 섞으면 원인이 흐려집니다.

문제가 성립하는지 먼저 확인했습니다. 우리 모델로 재기 전에 외부의 강한 범용 모델로 세 번 재며 문항을 고쳤습니다 — 신호 설명이 없으면 인코딩 알아맞히기가 됐고(밸브 신호의 의미를 반대로 읽었습니다), 설명을 줘도 **「이 설비의 평소 패턴」**을 모르면 연속 운전이 이상인지 알 수 없었습니다. 고장 한참 전 기간의 평균을 기준값으로 함께 주자 비로소 문제가 성립했습니다. 기준값을 주는 것은 정답을 흘리는 것이 아니라 **실제 진단 과정**입니다.

다음 판에 실측 기반 데이터를 넣습니다. 다음 판에는 **실측 기반 데이터**를 더합니다 — 센서 드리프트 · 결측 · 운전 조건 변화는 현장 데이터가 가장 잘 가르쳐 줍니다. **같은 158 문항으로 다시 재서** 판 사이를 견줍니다(문항을 고정해야 비교가 성립합니다).

18 · 배치와 다음 판

온프레미스 — 모델도 데이터도 사내에 머문다

바깥으로 나가는 것이 없습니다. 모델은 **고객의 장비에서, 고객의 데이터를 상대로** 돕니다. 망 요구사항이 없어 폐쇄망에서도 그대로 섭니다.

전달 형태

형식	양자화 체크포인트를 압축 아카이브(`tar.zst`)로. 안에 모델 디렉터리 하나가 들어 있고 **그대로 서빙**합니다
컨테이너 이미지	별도 협의
인터페이스	**OpenAI 호환 엔드포인트** — 기존 클라이언트를 그대로 씁니다
망 요구	**없음.** 모델과 데이터 모두 사내에 머뭅니다

GPU

권장	**4비트 부동소수(NVFP4) 네이티브 가속**이 되는 최신 세대 GPU
이전 세대	동작은 합니다 — **다만 가속 이득이 없습니다**
언어모델	**GPU 1장**으로 서빙(가중치 21 GB + KV 캐시)
검색 모델 둘	한 장에 함께 올릴 수 있습니다

크기

모델	양자화본
AURA-35B	약 **21 GB**
AURA-Embedding-4B	약 6 GB
AURA-Reranker-0.6B	약 1 GB

다음 판에 들어가는 것

확정된 것

- **성능 수치** — 제품 경로에서 재측정이 끝나면 16장의 표를 그대로 채웁니다
- **실측 기반 진단 데이터** — 합성에 현장 데이터를 더합니다(3 · 17장)
- **멀티턴 도구 사용** — 이 판은 단발 호출 위주입니다
- **선호학습** — 로드맵에 있습니다
- **비전** — 용접 결함 판정 벤치와 산업 이미지를 준비했습니다. **잣대를 먼저 만들고 그 다음에 학습합니다**

별도 트랙 — 파형을 보는 계열

파형에서 이상을 찾는 **백만 파라미터급** 계열로, 언어모델과 **별개**입니다. **GPU 없이 돌 만큼 작아** 엣지에 얹을 수 있습니다. 이 판에서는 **구조와 잣대**까지 적고, **제품 적용은 다음 판에서 다룹니다.** 구조만 말하면 **파형에서 무엇이 이상한지를 찾는 것과, 왜 그런지를 설명하는 것**은 서로 다른 물건입니다(15장).

라이선스

도입 조건은 KOPENS 와 **계약별로 협의**합니다. 오픈소스 구성요소의 라이선스·고지는 **모델 산출물과 함께** 전달됩니다.

함께 보기

AURA 모델 소개	docs.plantpulse.io /ko/ai/overview/aura-models
AI 플랫폼 문서	docs.plantpulse.io /ko/ai
데이터레이크 기술 백서	docs.plantpulse.io /ko/platform/whitepaper

문의

전화 · 이메일	1588-5268 · webmaster@kopens.com
문의 · 데모	kopens.com/ko/contact
제품 · 문서	plantpulse.io · docs.plantpulse.io

㈜한국오픈솔루션 · KOPENS, Inc. © 2026 KOPENS. 플랜트펄스®는 ㈜한국오픈솔루션의 등록상표입니다. 본 문서 Rev. 2026.09 — **구성과 검증 방법 판**입니다.