

AIは精度で壊れない。 文脈と状態定義で 壊れる。

TennisLaboで起きた

“同じ映像なのに違う正解”問題

Presented by

Masahiro Yamada

Date

2026.02.27

```
class ContextAwareAI:
    def __init__(self, state_definition):
        self.context = state_definition.context
        self.objective = state_definition.objective
    def evaluate(self, observation):
        # Accuracy != Value
        if not self.context: raise ContextError()
        return self.objective(observation, self.context)
```

前提：

TennisLabo とは何か？

我々が開発・提供しているのは、
単なる映像解析ツールではない。

◎ Mission

「理想の動作を定着させる」



テニス動作解析AI

コート上のカメラ映像から、選手の骨格・ボール軌道・ラケットワークをリアルタイムで検出し、数値化するシステム。



目的は「フォーム評価」ではない

「肘が90度だから100点」といった静的な採点ではない。
単発の良いショットを褒める装置ではない。



真の目的は「動作の定着」

重要なのは再現性（Reproducibility）。
無意識下でも同じ出力が出せるよう、状態を収束させることにある。

よくある誤解

💡 期待していたこと

BELIEF

「AIでフォームを解析すれば、
自動的に上達するはず」

可視化さえすれば人は変わるという仮説



⚠️ 実際の現実

REALITY

実際は**違った**。

「解析しても、定着しない」

THE MISSING PIECE

なぜか？それは**文脈**と**状態定義**が抜けていたから。

プロフェッショナルは 導入が極めて早い

トップレベルの指導現場では、
AI導入の障壁が低く、**即座に活用サイクル**が回
り始める。

⚡️ **すぐに使う (Immediate Use)**

説明書を読まずとも直感的に利用を開始。
出力されたデータに対し、**迷いなく解釈**を与えることができる。

🔄 **修正する (Self-Correction)**

AIの出力に違和感があれば、自己の知識で補正。
「AIの誤り」を**ノイズとして処理**し、有用な信号だけを抽出するフィルタを持つ。

📌 **定着する (High Retention)**

試行錯誤のループが回るため、システムへの信頼が醸成。
結果として**継続利用率 (Retention) が安定して高い**状態が続く。

一般ユーザーは 導入が遅い

プロと同じAIモデルを提供しても、
現場での運用が定着しない。

Key Insight

SAME MODEL
DIFFERENT OUTCOME



使わない (Low Adoption)

AIの出力を「参考情報」として捉えきれず、
既存の指導スタイルとの乖離に戸惑い利用を避ける。



続かない (Low Retention)

指摘された内容を実践しても効果を実感できず、
モチベーションが維持できない。



修正できない (No Correction)

AIの誤りやズレを自分たちで補正する
内部モデル（ドメイン知識）が不足している。

Failed Hypothesis

仮説：「精度の問題？」 精度を上げた。

しかし...



モデル精度 (F1 Score)

0.72 0.89

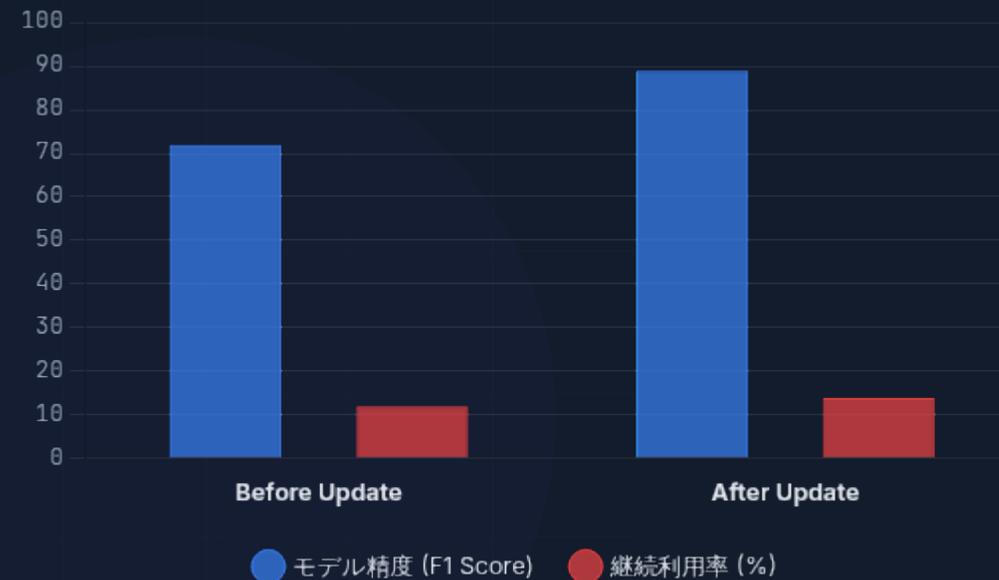
大幅な精度向上を達成



継続利用率 (Retention)

12% 14%

有意な変化なし (誤差範囲)



モデル支配ではない

精度が低いから使われないのではない。

「正しい」の定義自体が現場とズレている。

上達とは、状態が収束すること

Skill acquisition is a state convergence process in a control system.



つまり「制御系 (Control System)」の問題である

精度が高くても、このループが回らなければ (定義が曖昧なら)、システムは発散し、上達 (収束) は起きない。



コーチA
Evaluation



理由 (Reasoning)

- ✓ 打点が体の前にある
- ✓ ラケットヘッドが走っている



WHY?



コーチB
Evaluation



理由 (Reasoning)

- ✗ 守備的な状況である
- ✗ この打点だと次の戻りが遅れる



「同じ映像」なのに「評価が逆」

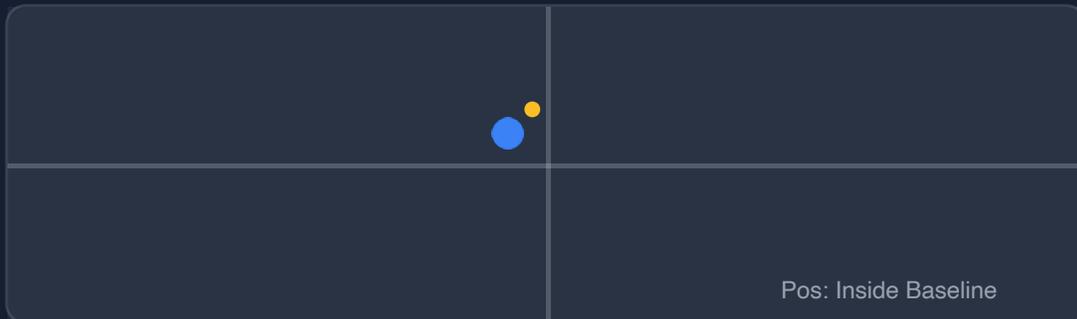
AIはどちらを正解 (Ground Truth) として学習すべきか？

CRITICAL ISSUE
Label Consistency Error

Case 1: Attack 攻撃的状况



相手が崩れており、チャンスボールが来ている



Action

打点：早い (Early)

位置：ベースライン内側

 **EVALUATION**
GOOD (良い)

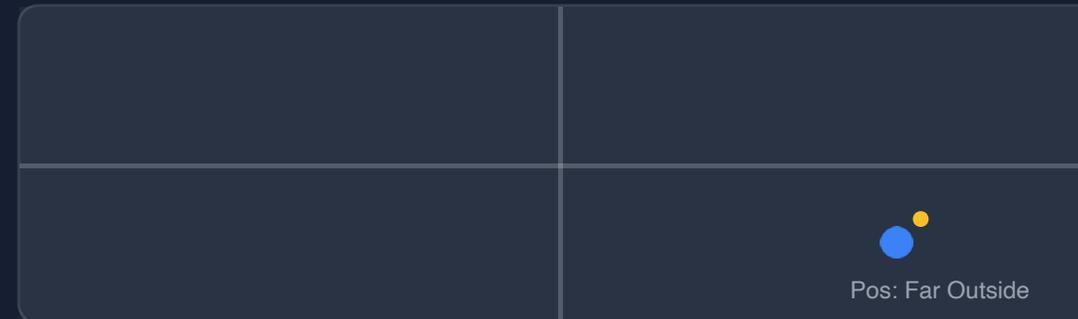
時間を奪い、相手を追い込めているため正解。

VS

Case 2: Defense 守備的状况



相手に振られており、体勢が苦しい



Action

打点：早い (Early)

位置：コート外遠方

 **EVALUATION**
BAD (悪い)

時間を稼ぐべき場面でリスクを取りすぎているため不正解。

Visual Similarity
High (Almost Identical)

⚠️ 映像上の「スイングの形」は同じでも、文脈によって正解が180度変わる。

Key Insight
文脈なきモデルは「形」だけ学習して「意味」を誤る

AIにとっての世界観

Label=Ground Truth

(ラベルこそが絶対の正解)



現実の正解構造

Truth=Context × State × Objective

(文脈と目的が正解を決定する)

文脈 (Context) が欠落すると何が起きるか？

定義の矛盾

「同じ状態」に対して「異なるラベル」が付与される。AIは収束できず、平均的な（無難な）回答に逃げる。

ノイズ増大

文脈による正解の変動がすべて「観測ノイズ」として処理され、モデルの信頼度が著しく低下する。

意味崩壊

「良いフォーム」という概念が文脈から切り離され、実戦で役に立たない「綺麗なだけのフォーム」を学習する。

**コンテキスト不足は意味の崩壊を招く**

Surface Level

ドメイン知識なし



表層特徴のみに注目

PROCESS (処理)

```
# 見た目のパターンで判断
if looks_like_pattern(input):
    return LABEL
```

- × **文脈を切り落とす**
前後の状況や意図を無視
- × **相関構造に沿う**
「それっぽい」を学習してしまう

Deep Structure

ドメイン知識あり



状態空間と目的関数を定義

PROCESS (処理)

```
# 因果構造と文脈を考慮
if (context.match_situation and
    opponent.position and
    next_shot.prob > threshold):
    return LABEL
```

- ✓ **重要変数を理解している**
何を無視して良いかを知っている
- ✓ **因果構造に沿う**
状態空間を圧縮し、ノイズを除外

VS

CORE PHILOSOPHY

アノテーションとは、**状態空間と目的関数の定義**である



何が起きるか

文脈を含まないラベルが引き起こす負の連鎖



THE FATAL FLAW

文脈を捨てたAIは、「ランダムな教師」として機能してしまう。



プロフェッショナル

INTERNAL MODEL: STRONG

AI OUTPUT

Ambiguous

"Bad Form" (No Context)



INTERNAL CORRECTION

+ 文脈を内部モデルで補完

「今は守備的状况だから、このフォームで正解だ」と解釈



自己補正ループ成立

AIの曖昧さを吸収して活用



一般ユーザー

INTERNAL MODEL: WEAK

AI OUTPUT

Ambiguous

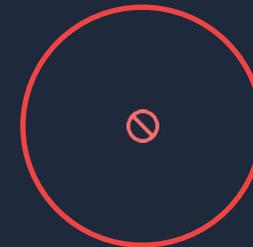
"Bad Form" (No Context)



DEPENDENCY

× 外部ラベルに完全依存

「AIがダメと言ったからダメなんだ」と誤認



ループ崩壊・離脱

文脈不足をそのまま受け取る



FIX 01

文脈・定義の修正

Context & Definition

- ✓ **状況ラベルの追加**
「速度」か「コントロール」かを明示的にラベル付与。同じ打点でも状況により正解を変える。
- ✓ **評価基準の明文化**
暗黙知だった「良いフォーム」を言語化し、アンテーション仕様書に落とし込む。



FIX 02

特徴量設計の修正

Feature Engineering

- ✓ **相対座標化（身長補正）**
絶対座標による個人差ノイズを排除。身体比率に基づいた座標系へ変換。
- ✓ **時系列コンテキスト**
静止画判定をやめ、前後フレームを含むシーケンスとして入力を再設計。



FIX 03

運用プロセスの修正

Operational Process

- ✓ **Disagreementのログ化**
評価が割れたデータを「学習除外」せず「重要境界」として記録・分析する。
- ✓ **ループ成立条件の監視**
精度だけでなく、ユーザーが修正行動を取れたかをKPIとして追跡。



「モデルを変える前に、**入力と定義**を変える。」

KEY INSIGHT

モデル変更ほぼなし

アルゴリズムそのものではなく、
「入力データの質（文脈）」と
「評価基準（仕様）」を変えただけで
劇的な改善が見られた。

“ 「AIは変えていない。
教え方を変えただけ。」

-- PROJECT RETROSPECTIVE



STABILITY

修正指示の安定性

4.2 / 5.0 ↑ 1.8pt



RETENTION

週間定着率

68 % ↑ 2.4x



IMPACT

一般層ループ成立

改善 Confirmed

ループ成立率の推移 (プロ vs 一般)





01. DEFINITION

状態と目的の整合性

State Space & Objectives

- **状態空間は文脈込みか？**
単なる画像特徴だけでなく、試合状況、時系列、相手位置などの「文脈」を含んでいるか。
- **ラベルは目的関数と一致しているか？**
その「正解」を学習することで、本当にユーザーの上達（目的）に寄与するか。



02. DOCUMENTATION

仕様化とログ保存

Specs & Disagreements

- **アノテーション定義書はあるか？**
暗黙知を排除し、判断基準を言語化・数式化しているか。仕様書なしのアノテは禁止。
- **Disagreementを記録しているか？**
評価者の意見割れはノイズではない。境界値の貴重なデータとして保存しているか。



03. SYSTEM LOOP

制御ループの設計

User Feedback Loop

- **顧客層ごとのループ成立条件を設計しているか？**
プロと一般では「観測→介入→修正」のループが回る条件が異なる。
一般ユーザーが自律的に補正できるUI/UXまで含めて設計されているか。



AIはパターン学習機ではなく、**定義を固定する装置**である。

AIは精度で壊れない。 文脈と状態定義で壊れる。

モデル改善より前にやるべき、3つの本質的アクション。

STEP 01 状態を定義せよ



「良し悪し」の境界線はどこか？
暗黙知を数式化・言語化し、
状態空間を明確にする。

"Define the State Space"

STEP 02 文脈を含めよ



そのラベルは「いつ」「誰が」
「何のために」付けたのか？
コンテキスト特徴量をモデルに入れる。

"Include the Context"

STEP 03 ループを設計せよ



観測→介入→再観測のサイクル。
ユーザーが自己補正できる
フィードバック構造を作る。

"Design the Loop"

Closing Thought: AIは、あなたの設計思想を増幅する装置である。

自己紹介

山田将大 / Yamada Masahiro

高校2年生 ロボット世界二位

大学2年生 Laplus会社取締役

大阪大学 鷺尾研究室にてAIを研究

現在 株式会社TAFDATA 代表

現在はアスリートのパフォーマンス向上のため
AIサービスの開発・運用などを行う。

