NegoPlay

משולחן הברידג' אל שולחן המשא ומתן:
האם סגנון קבלת-החלטות עובר בין תחומים?

תיק פרויקט סופי — גרסה עברית

דו"ח סיכום · (הדו"ח המכין והתוכנית העסקית — בתיק האנגלי המלא)

מגישה:	אנה בן-שושן
מרצה הקורס:	ד"ר יורם סגל
מנחה הדוקטורט:	ד"ר נצר יעקב זיידנברג
קורס:	AI Development Expert
מוסד:	TESI — The External Studies Institute
תאריך הגשה:	14 ביוני 2026

תקציר מנהלים

נתוני משא ומתן בעולם האמיתי הם סודיים ובלתי-מדידים, ולכן כמעט בלתי אפשרי לחקור התנהגות משא ומתן בקנה מידה. **NegoPlay** משתמש בברידג' תחרותי כמעבדה מדידה לאותן החלטות בדיוק — סיכון, לחץ, והחלפת מידע — ושואל האם ***סגנון קבלת-ההחלטות*** *של* שחקן עובר מהברידג' אל משא ומתן עסקי.

מתוך 149,208 רשומות תחרות (563 שחקנים כשירים), צינור למידה לא-מונחית גילה חמישה פרופילי החלטה מאומתים סטטיסטית (Cohen's d בין 2.1 ל-4.6, כולם $p<0.05$). כל פרופיל הומר לסוכן מבוסס מודל-שפה שכרטיס-האופי שלו מכיל ***אך ורק*** כישורים שנגזרו מהברידג', והסוכנים ניהלו משא ומתן עסקי מול מוכר שכויל על 5,247 משא-ומתנים אמיתיים מ-Craigslist.

העברת ***ניצחון*** התבררה כחלשה ורגישה למדד ($\rho \approx +0.2$); זיקוק השאלה לצורתה ההתנהגותית המקורית הראה ש**הסגנון עובר חזק**: תוקפנות בברידג' ותוקפנות במשא ומתן מתואמות ב-**Spearman $\rho = +0.80$**, מעל יעד ה-0.70 שהוגדר מראש. בקרת פרומפט-הפוך הופכת את המתאם ל-0.90- — הוכחה שההעברה נישאת על-ידי הכישורים שנגזרו מהברידג', ולא על-ידי תוויות. מדד הברידג' עצמו אומת באמצעות מבחן בסיס-אקראי ("קוף") ומעריך double-dummy. המערכת ניתנת לשחזור מלא: נקודת כניסה אחת (SDK), 100+ בדיקות יחידה, סימולציות עם seed קבוע, ותיעוד עלות לכל קריאה (סך הוצאות $1.70 ≈ LLM).

תודות

ברצוני להודות לאנשים שליוו את הפרויקט הזה מרעיון למערכת עובדת.

ד"ר יורם סגל — על מסגרת הפרויקט ההנדסית שעיצבה את העבודה: משמעת ה-SDK האחיד, הדרישה לתוצרים מדידים, והסטנדרט שגם אב-טיפוס מחקרי צריך להיות תוכנה בנויה היטב.

ד"ר נצר יעקב זיידנברג — מנחה הדוקטורט שלי, על מומחיות בתחום הברידג' — מתיקוני גודל-מדגם בגילוי הפרופילים ועד עוגן ה-double-dummy שביסס את מדד המיומנות — ועל הליווי האקדמי שבתוכו הפרויקט הזה משמש בסיס אמפירי.

ד"ר רמי — על הנחיה מתודולוגית חדה: הדרישה לקווי-בסיס, ביקורות עיבוד-מקדים ויושרה סטטיסטית דחפה את הפרויקט לאמת כל מספר שהוא מדווח — כולל מבחן הבסיס-האקראי ששינה את מדד הברידג'.

ולבסוף — למשפחתי, על הסבלנות בלילות של ריצות סימולציה, ועל שהעמידו פנים שהכפלות-עונשין זה נושא מרתק.

תוכן עניינים

1. סקירת הפרויקט — הבעיה, הרעיון, הצינור, הנתונים

2. תוצאות מרכזיות — מדד אמין (קוף + double-dummy) · הגילוי: סגנון עובר, ניצחון לא · בקרת האנטי-טאוטולוגיה · מגבלות

3. שכבת ה-AI: סוכני LLM והנדסת כישורים — ארכיטקטורת סוכן · חילוץ כישורים · פרומפטים כקונפיגורציה · אי-דטרמיניזם · לקוח LLM ניטרלי-ספק

4. הנדסת תוכנה — ארכיטקטורת SDK · תבניות עיצוב · בדיקות · הנדסת אמינות · שחזוריות · איכות קוד ועלות

5. מסקנות והמשך

חלקים II–III (דו"ח מכין מלא, 51 עמ' + תוכנית עסקית, 20 עמ') — בתיק האנגלי: final_portfolio.pdf

1. סקירת הפרויקט

הבעיה והרעיון

משא ומתן עסקי הוא אחת המיומנויות המשפיעות ביותר בתעשייה — אך כמעט בלתי אפשרי לחקור אותו כמותית: תמלילים חסויים, תוצאות לא מתוקננות, ואין שום מאגר ציבורי שמקשר בין סגנון ההחלטה של אדם להתנהגותו במשא ומתן. ברידג' תחרותי פותח דרך עוקפת: כל חלוקה היא רצף החלטות מתועדות תחת אי-ודאות — כמה גבוה להתחייב, מתי להסתכן, מתי להילחם — ולכל החלטה תוצאה מדידה. NegoPlay מתייחס לברידג' כ**מעבדה התנהגותית**: לגלות פרופילי קבלת-החלטות בנתוני ברידג', לגלם אותם בסוכני LLM, ולבדוק אם הסגנון שהם מקודדים עובר לתחום אסטרטגי אחר.

צינור המערכת

1. **גילוי פרופילים (ML):** 149,208 רשומות ← 8 תכונות התנהגות לשחקן ← פרופיילינג אחוזון-קיצון עם אימות בינומי ← 5 פרופילים (צייד סלאמים, שחקן ביטוח, לוחם, מומחה NT, כללי).
2. **חילוץ כישורים (LLM):** ידיים אמיתיות של כל פרופיל נשלחות למודל השפה, שמחזיר 5–7 כישורים מאפיינים כ-JSON מובנה.
3. **בניית סוכנים:** כל פרופיל הופך לסוכן שכרטיס-האופי שלו מכיל *רק* את הכישורים שנגזרו מהברידג' — לעולם לא שמות-תואר של אישיות (כלל האנטי-טאוטולוגיה).
4. **סימולציה כפולה ויישור:** הסוכנים מכריזים בברידג' ומנהלים משא ומתן עסקי; ההתנהגות והתוצאות בשני התחומים מתואמות זו מול זו.

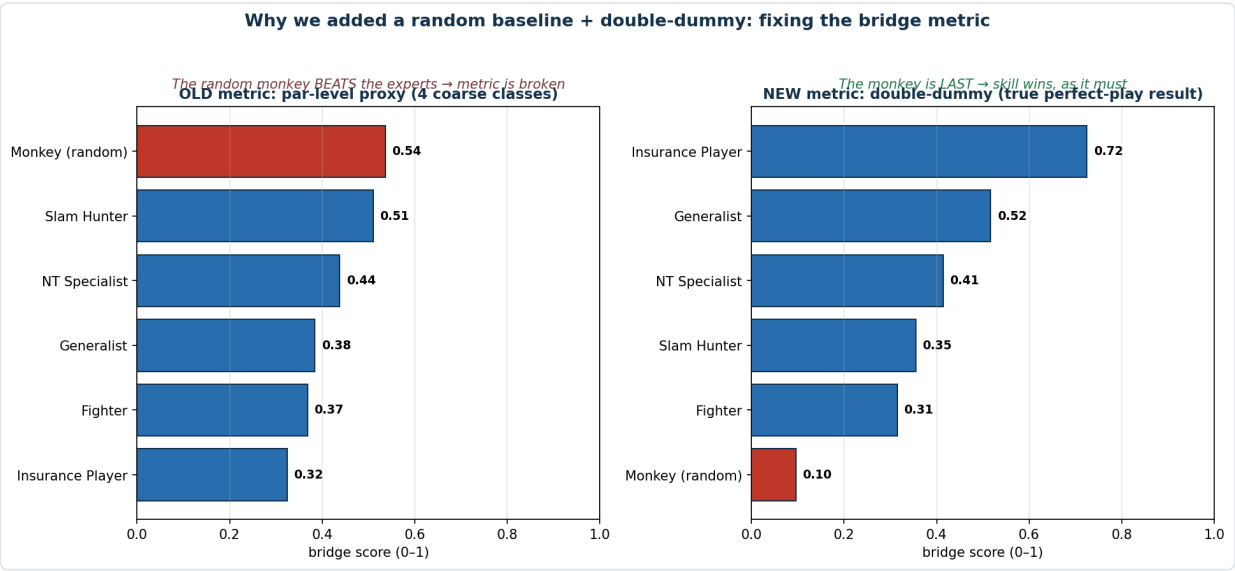
נתונים

- **ברידג':** 149,208 רשומות-לוח ממאגר האליפויות האירופי הציבורי (2016–2025), שנאספו בצינור גירוד-נתונים שנכתב עבור הפרויקט; 563 שחקנים עוברים את סף האמינות (≥50 לוחות).
- **משא ומתן (אימות חיצוני):** 5,247 משא-ומתני מחיר אמיתיים בין בני אדם (קורפוס Craigslist של סטנפורד) — לאימות תכונות ההתנהגות ולכיול המוכר המדומה.

2. תוצאות מרכזיות

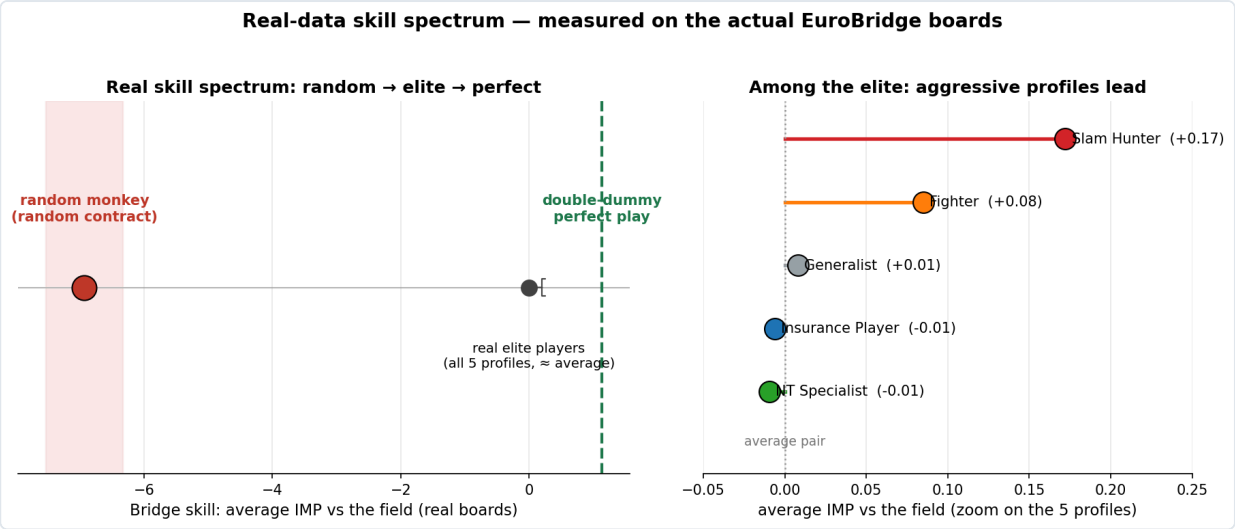
מדד ברידג' אמין (קוף + double-dummy)

מדד מיומנות חייב לדרג סוכן אקראי במקום האחרון. המדד הגס הראשון שלנו נכשל במבחן — "קוף" שמכריז אקראית *עקף* את כל פרופילי המומחים (0.54 מול 0.25–0.51). ניקוד מחדש של כל חוזה לפי השאלה האם הוא ***באמת מצליח*** בהערכת double-dummy (משחק מושלם) הפיל את הקוף למקום האחרון (0.10):



הבסיס האקראי חושף (שמאל) ומדד ה-double-dummy מתקן (ימין) את ציון הברידג'.

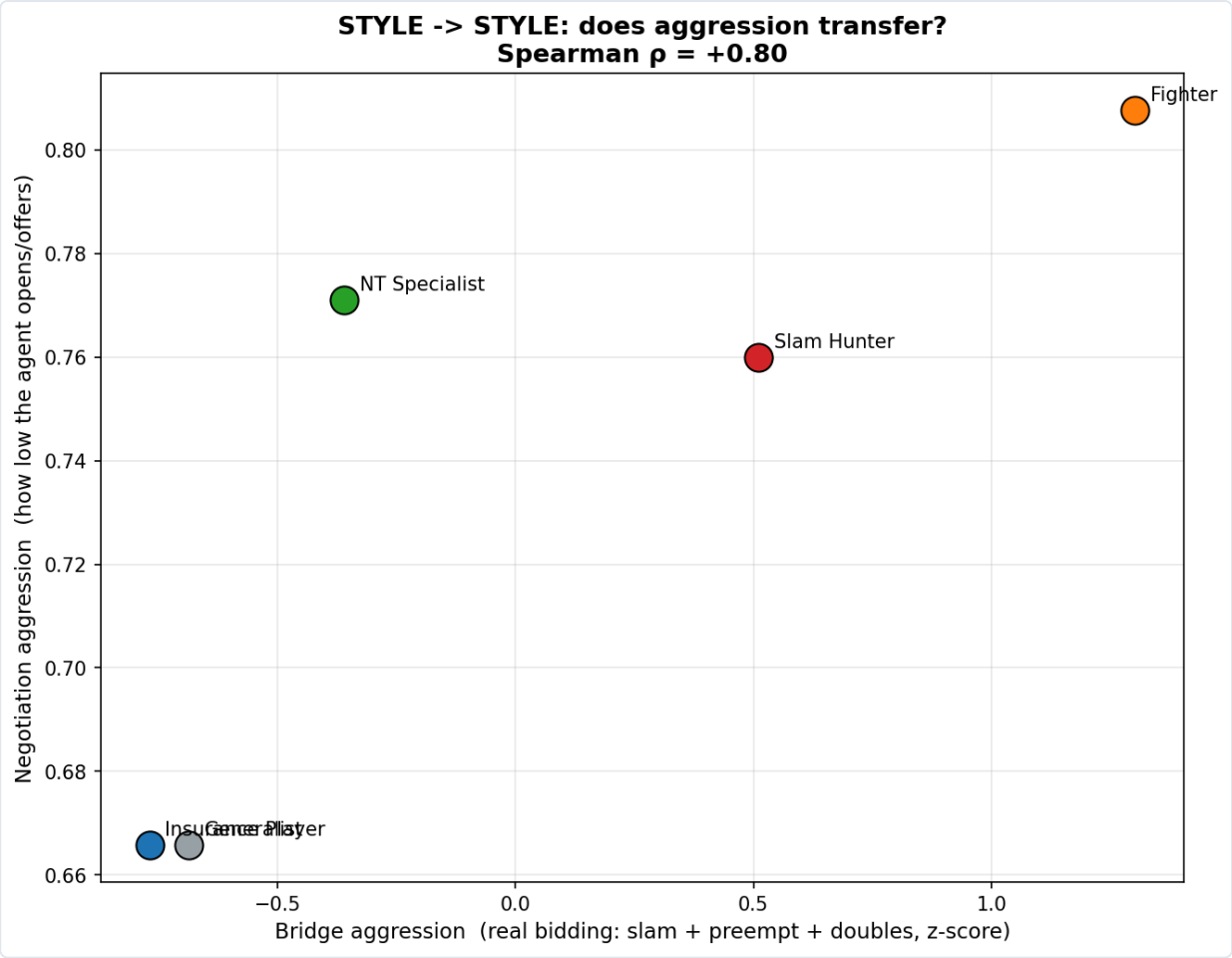
את מיומנות הברידג' מדדנו אז מהתוצאות התחרותיות ***האמיתיות*** (IMP מול השדה בשיטת duplicate, כולל הגנה). על הסולם הזה הקוף האקראי יושב על 7- IMP ללוח, פרופילי העילית מתקבצים סביב 0, ומשחק מושלם מסמן את התקרה ב-+1.1:



סרגל המיומנות על הלוחות האמיתיים: רצפה אקראית ← שחקני עילית ← תקרת משחק-מושלם.

הגילוי: הסגנון עובר — הניצחון לא

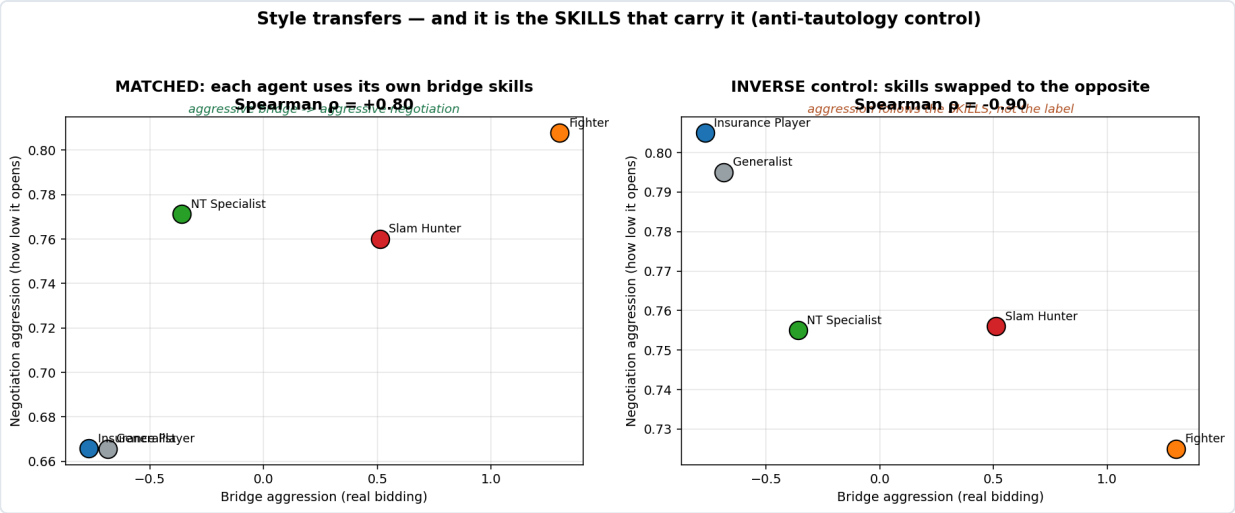
מתאם של *ניצחון* בין התחומים התברר כחלש ורגיש למדד (ρ ≈ +0.2 עד +0.5), כי האם סגנון מנצח תלוי ביריב ובחוקי התגמול. זיקוק השאלה לצורתה ההתנהגותית המקורית — האם פרופיל תוקפני בברידג' *מתמקח* בתוקפנות? — נותן **Spearman ρ = +0.80**:



סגנון↔סגנון: תוקפנות בברידג' (הכרזות אמיתיות) מול תוקפנות במו"מ (כמה נמוך הסוכן פותח), ρ=+0.80.

בקרת האנטי-טאוטולוגיה

כדי לשלול את הטענה "הסוכן תוקפני כי תייגתם אותו תוקפני", כל פרופיל שמר על זהותו אך קיבל את כישורי הברידג' של הפרופיל *ההפוך*. המתאם התהפך מ-+0.80 ל-(p=0.037) **-0.90** — ההתנהגות עוקבת אחרי הכישורים המוזרקים שנגזרו מהברידג', לא אחרי התווית:



כישורים תואמים (שמאל, +0.80) מול כישורים מוחלפים (ימין, -0.90): הכישורים נושאים את הסגנון.

מגבלות — בכנות

עם n=5 פרופילים, מתאם הסגנון הוא אינדיקציה (p≈0.10) ולא מובהקות; המשא ומתן מדומה (מכויל על משא-ומתנים אנושיים אמיתיים, אך אין נתוני מו"מ של אותם אנשים); ופרופיל "צייד הסלאמים", שמוגדר לפי *הכרזת* סלאמים, עשוי לשקף חלקית עוצמת-משחק כללית. הכל מתועד, לא מוסתר.

3. שכבת ה-AI: סוכני LLM והנדסת כישורים

ארכיטקטורת סוכן

כל הסוכנים יורשים ממחלקה אבסטרקטית `BaseAgent` שמחזיקה את חתימת הפרופיל, את לקוח ה-LLM, ו**נקודת-חנק החלטה אחת**. אף סוכן לא קורא ל-SDK של ספק ישירות — כל החלטה זורמת דרך מתודה אחת, שמרכזת ניסיונות-חוזרים, פירוק JSON, מדיניות טמפרטורה ותיעוד עלות:

```
class BaseAgent(ABC):
    def _decide(self, user_prompt, response_schema, purpose) -> dict:
        """The ONLY path to the LLM: schema-enforced, logged, retried."""
        response = self.client.generate(
            system=self.system_prompt, user=user_prompt,
            purpose=purpose, temperature=self.temperature,
            response_schema=response_schema)
        return response.json or json.loads(response.text)
```

שני סוכנים קונקרטיים ממשים את התחומים: `BridgeAgent.make_bid()` (טמפרטורה 0 לשחזוריות; כל הכרזה מאומתת בבודק-חוקיות מקומי ללא LLM, עם נפילה ל-Pass אם אינה חוקית) ו-`NegotiationAgent.respond_to_offer()` (טמפרטורה 0.7; מחזיר counter/accept/walk_away עם מחיר חסום לטווח החוקי של התרחיש).

חילוץ כישורים וכלל האנטי-טאוטולוגיה

שלב 2 הופך ידיים גולמיות ל*פרופיל כישורים*: הידיים נחתכות לחבילות (20–30 לקריאה), נשלחות עם סכמת JSON קשיחה (`response_mime_type="application/json"`), ומאוחדות בין החבילות. הכלל ההנדסי המרכזי: לסוכנים **לעולם** לא נותנים שמות-תואר של אישיות. בכרטיס של ה"לוחם" כתוב "*מכפיל יריבים בקצב של פי 6–8 מהממוצע*" — עובדה שחולצה מהמכרזים האמיתיים של שחקניו — ולא "*תהיה תוקפני*". כל התנהגות חוצת-תחומים חייבת אפוא *לצמוח* מכישורי הבריגד'. בקרת הפרומפט-ההפוך (פרק 2) היא המבחן הביצועי של הכלל: החלפת הכישורים מחליפה את ההתנהגות (ρ: +0.80 ← -0.90).

הנדסת פרומפטים כקונפיגורציה

הפרומפטים הם ארטיפקטים מבוקרי-קוד, לא מחרוזות אד-הוק: ספרייה מרכזית (`shared/prompts.py`) בונה כל כרטיס-אופי מאותה תבנית — שורת זהות, 5–7 כישורים שחולצו, חוקי

התחום, סכמת פלט קשיחה ודוגמת few-shot. טמפרטורה, סכמה וספק הם פרמטרים — כך שניסוי הוא שינוי קונפיגורציה, לא שכתוב.

הנדסה מול אי-דטרמיניזם

מודלי שפה אינם דטרמיניסטיים אפילו בטמפרטורה 0 (ריצה מקבילית על GPU). לכן השחזוריות מטופלת ברמת ה*מדגם*: כל (פרופיל, חלוקה) רץ שלוש פעמים עם הכרעת-רוב; כל תשובה גולמית נשמרת ל-JSONL לפני כל עיבוד; וכל הניתוחים ניתנים לחישוב-מחדש מהקבצים השמורים ללא קריאות API חדשות. החלוקות עצמן נוצרות ממחולל עם seed קבוע, כך שכל הניסוי משוחזר ביט-אחר-ביט עד גבול ה-LLM.

לקוח LLM ניטרלי-ספק

`LLMClient` אחד עוטף את Gemini (ברירת מחדל), Claude ו-OpenAI מאחורי ממשק אחיד (תבנית Adapter). הוא מממש ניסיונות-חוזרים עם השהיה מעריכית, timeout של 60 שניות לכל בקשה (סוקט תקוע עצר פעם אצווה של 1,500 קריאות — ה-timeout הופך תקיעות לכשלים ניתנים-לניסיון-חוזר), ויומן עלויות JSONL לכל קריאה (ספק, מודל, טוקנים, השהיה, דולר). סך הוצאות ה-LLM של הפרויקט: ≈ $1.70.

4. הנדסת תוכנה

ארכיטקטורה: נקודת כניסה אחת (SDK)

בסיס הקוד נשמע למשמעת facade קפדנית: כל פעולה ציבורית נגישה דרך `src/sdk.py`, ואף פונקציה לא חיה רק במחברת או בסקריפט. המודולים מאורגנים לפי שלבי הצינור:

```
src/
  sdk.py                  # facade: the single public contract
  stage1_clustering/      # features, profiling, validation
  stage2_skills/          # chunker, extractor, aggregator
  stage3_agents/          # BaseAgent, BridgeAgent, NegotiationAgent
  stage4_simulate/        # bridge_game, bridge_auction, negotiation,
                          # monkey_agent, alignment
  features/double_dummy.py  # DDS perfect-play evaluator (endplay)
  shared/                 # llm_client, prompts, data_loader, logger
```

תבניות עיצוב בשימוש פעיל: **Facade** (`sdk.py`), **Template Method** (`BaseAgent._decide()`), **Adapter** (לקוח ה-LLM הניטרלי), ו-**Strategy** (סוכני פרופיל מתחלפים — כולל סוכן ה"קוף", שמותאם-ממשק כך שהוא נכנס לכל runner ללא שינוי קוד).

בדיקות

הפרויקט כולל 100+ בדיקות `pytest`: יחידות לוגיקה טהורה (יצירת חלוקות, ניקוד, חוקיות מכרז — 28 בדיקות למנוע הברידג' לבדו), סוכן הבסיס-האקראי (חוקיות כל הכרזה אקראית, שחזוריות תחת seed קבוע), מעריך ה-double-dummy (קיבועים ידועים: חלוקת 12 לקיחות חייבת לתת 6NT=1.0 ו-7NT=0.0), והשכבות תלויות-ה-LLM נבדקות באמצעות *הזרקת לקוח מדומה (mock)* — בלי רשת, בלי מפתח API, בלי עלות:

```
def test_monkey_bids_are_always_legal_over_many_auctions():
    monkey = MonkeyAgent(seed=1)
    for auction in AUCTION_FIXTURES:
        for _ in range(50):
            out = monkey.make_bid(HAND, auction)
            assert is_legal_call(out["bid"], auction)
```

הנדסת אמינות

ריצות הסימולציה הארוכות מהונדסות לשרוד את מצבי-הכשל שנצפו בפועל במהלך הפרויקט: timeout לכל בקשת HTTP (תקיעה ← ניסיון חוזר), ניסיון-חוזר *אטומי* לכל לוח עם השהיות (גולש מעל עומסי 503 של הספק בלי לאבד את האצווה), ושמירת JSONL מצטברת עם `flush()` אחרי כל לוח — ריצה שנהרגה שומרת את כל העבודה שהושלמה, וההתקדמות נצפית ממערכת הקבצים.

שחזוריות ומשמעת נתונים

זרעים קבועים (`random_state=42`) בכל scikit-learn ובמחולל החלוקות; נתוני הגלם בלתי-ניתנים-לשינוי (`data/raw` לקריאה בלבד); כל מספר נגזר ניתן למעקב עד ארטיפקט שמור; וההיסטוריה המלאה ב-Git תחת משמעת conventional-commits. כל איור בתיק הזה נוצר על-ידי סקריפט שב-Git — שום דבר לא צויר ביד, כך שהתיעוד לא יכול לסטות מהנתונים.

איכות קוד ועלות כתקציב הנדסי

סגנון וטיפוסים נאכפים עם `ruff`, רמזי-טיפוס ותיעוד בסגנון Google בכל הפונקציות הציבוריות. עלות LLM מנוהלת כתקציב הנדסי: בחירת המודל אמפירית (`gemini-2.5-flash-lite` נמצא מהיר פי 10 מברירת המחדל לעומס הזה), כל קריאה נמדדת, והפרויקט כולו — כולל כל הריצות החוזרות — עלה ≈ $1.70.

5. מסקנות והמשך

מסקנות

1. שחקני ברידג' בכירים יוצרים **רצף התנהגותי**, לא אשכולות; פרופילים מאומתים סטטיסטית חיים בזנבות הקיצוניים שלו.
2. **מדד שסוכן אקראי יכול לנצח בו — שבור.** מבחן הבסיס-האקראי שינה בפועל את המדידות של הפרויקט.
3. **סגנון ההחלטה עובר בין תחומים** (ρ=+0.80, לא-טאוטולוגי לפי בקרת פרומפט-הפוך שבוצעה); **ניצחון אינו עובר נקי**, כי ניצחון הוא תכונה של הזוג (סגנון × סביבה).

המשך

יותר פרופילים (חלוקה עדינה יותר של 563 השחקנים) לכוח סטטיסטי; מדידת סגנון-מו"מ רב-ממדית (קצב ויתור, נטיית-נטישה); שכפול חוצה-מודלים (Claude/GPT); הפרדת סגנון מעוצמת-משחק; ולבסוף — נתוני משא ומתן אמיתיים של אותם אנשים.

היכן שאר התיק

התיק המלא להגשה הוא `final_portfolio.pdf` (87 עמ', אנגלית): עמודי הפתיחה לפי ההנחיות, דו"ח הסיכום (המקביל למסמך זה), הדו"ח המכין המלא (51 עמ') והתוכנית העסקית (20 עמ'). המסמך הנוכחי הוא הגרסה העברית של דו"ח הסיכום — לקריאה ולמעבר נוח.

NegoPlay · תיק פרויקט סופי — גרסה עברית · נתונים: 149,208 רשומות 563, EuroBridge (2016–2025) שחקנים, 5 פרופילים · אימות חיצוני: 5,247 משא-ומתנים אמיתיים · כל האיורים נוצרו מסקריפטים שב-Git.