

TRACE-RPG: 인터랙티브 게임 세계의 생성 이벤트를 위한 trace 연결 기호 커밋 게이트

익명 저자

Abstract—대규모 언어 모델은 표현력 있는 퀘스트와 대화를 생성할 수 있지만, 유창하거나 스키마에 맞는 텍스트만으로 제한 이벤트의 실행 가능성, 권한 적합성, 정식 게임 상태와의 일관성이 성립하지 않는다. 본 논문은 생성된 모든 이벤트를 신뢰하지 않는 트랜잭션 제안으로 취급하는 기호 커밋 게이트 TRACE-RPG 을 제안한다. 타입 엄격 parser, 외부 공급 행동 정책, 여섯 상태 상대 계열로 묶인 일곱 개의 결정론적 검사가 각 허용 전이를 결정하며, 실패 후보는 최대 $K + 1$ 개의 기록된 시도 안에서 제한 수리로 보낼 수 있고 통과 후보는 변경 직전에 다시 검증한다. 반례 유도 연산자 ρ 는 직전 후보와 ρ typed validation error 만 읽고 권위 있는 상태는 결코 읽지 않는다. 키 없는 SHA-256 콘텐츠 연결, 상태 의미 replay, 할당 완전 계수가 모든 terminal outcome 을 감사 가능하게 만든다. 근거는 세 레인으로 분리한다. 단일 작성 world 의 결정론적 오프라인 적합성 파일럿, 단일 모델 라이브 스크리닝 확장, 런타임 권한이 없는 closed-world 지식 그래프 링크 제안 시뮬레이션이다. 오프라인 레인은 mechanism 적합성만 확립한다. 라이브 레인은 $K=1$ 에서 의도적으로 수리 가능하게 만든 policy-blind regime 에 한해 guided 5/5 대 blind 0/5 commit 을 관찰했으므로 pilot-only 로 남는다. 어느 레인도 모집단 효능, 모델 우월성, 플레이어 경험, 상용 엔진 성능을 확립하지 않는다.

Index Terms—게임 인공지능, 인터랙티브 내러티브, 기호 검증, 제약 생성, 런타임 강제, 재현성

I. 서론

인터랙티브 생성은 언어 과제에 그치지 않는다. 후보 이벤트가 자연스러워도 존재하지 않는 아이템을 요구하거나, 화자가 모르는 사실을 주장하거나, 금지된 퀘스트 정보를 공개하거나, 허가되지 않은 효과를 도입하거나, 퀘스트 단계를 후퇴시킬 수 있다. 실행 가능한 텍스트 환경은 그럴듯한 발화와 유효한 전이의 차이를 명시한다 [1], [2]. 근거 기반 대화와 개인화 퀘스트 시스템은 퀘스트, 엔터티, 그래프 맥락의 가치를 추가로 보여주지만 [3], [4], 맥락 자체가 정식 상태 변경을 승인하지는 않는다. TRACE-RPG 은 생성된 게임 이벤트를 신뢰하지 않는 트랜잭션 제안으로 취급하며, 정식 상태 변경을 외부의 결정론적 commit gate 를 통해서만 허용한다. 그림 1은 그러한 트랜잭션 하나를 왼쪽에서 오른쪽으로 읽는다.

구조적 생성 제약은 인접한 문제를 해결한다. 문법 제약 디코딩, 언어 모델 질의 언어, 점진 파서는 생성 구조를 제한할 수 있다 [5], [6], [7]. 그러나 구문상 허용되는 이벤트도 현재 세계에 대해서는 거짓일 수 있다. 검색과 메모리는 유용한 근거를 공급할 수 있지만 [8], [9], [10], 검색된 내용 역시 상태 전이의 권한 주체가 아니다.

본 논문은 이러한 관심사를 감사 가능한 트랜잭션 프로토콜로 결합한 구현인 TRACE-RPG 을 제시한다. 현재 기여는 다음과 같다.

- 1) **C1—계약**: 버전형 신뢰경계 contract 와, proposal/replay 경계에서 unknown key 와 모호한 타입을 거부하는 type-strict parser (IV절; E1);

익명 심사를 위해 제출된 원고입니다.

TABLE I
기여, 선행 주제, 근거 레인, 추론 상한 (구조화 매트릭스에서 생성)

ID	기여	선행 주제 (참고 문헌 수)	레인	추론 상한
C1	신뢰경계 contract	T1, T2, T3 (15)	E1, ENG1	인코딩 필드 계약 적합성
C2	validate-repair-commit controller	T1, T8 (14)	E1, ENG1	동결 사례의 메커니즘 동작
C3	감사 연결 근거 계층	T4, T7, T8 (6)	E1, ENG1	명명 fixture 의 무결성 · 재생
C4	assignment-complete harness	T4, T6, T7 (18)	E1	설계 사례 정확 집계
C5	반례 유도 연산자 ρ	T1, T8, T9 (8)	E1, E2	동결 class + 단일 regime screen

T1 = 근거 게임 세계; T2 = 검색 · 메모리 에이전트; T3 = 구조 생성; T4 = 환경 · 벤치마크; T5 = 플레이어 경험; T6 = 인간 · LLM 평가; T7 = 통계 · 재현성; T8 = 계획 · 런타임 개입; T9 = 피드백 수리. 레인은 표 IV에 정의한다. 모든 행은 구현 · 작성 fixture 검증 상태이며 어떤 효능 claim 도 지지하지 않는다.

- 2) **C2—controller**: 여섯 상태 상대 계열에 걸친 검사 7 개, bounded repair, unchanged fallback, pre-commit revalidation 을 갖춘 validate-repair-commit controller (IV-B절; E1, ENG1);
- 3) **C3—감사**: SHA-256 연결 record, state-semantic replay, episode-continuity check (IV-C절; E1, ENG1);
- 4) **C4—harness**: proposal trace 가 존재할 수 없는 경우에도 분류된 terminal row 를 유지하는 assignment-complete 오프라인 적합성 harness (V절; E1);
- 5) **C5—수리**: 동결 repairability class 에서 blind retry 및 state-reading oracle 과 비교한 counterexample-guided operator $\rho(a, E)$ (IV-B절; E1, E2).

기계 검증되는 기여 매트릭스에서 생성한 표 I은 C1–C5 를 선행 주제, 근거 레인, 추론 상한에 결합하고, 표 IV는 레인을 정의한다.

레인은 서로 대체할 수 없다 (표 IV). E1 은 단일 작성 world 의 결정론적 적합성 파일럿으로 인코딩된 메커니즘 동작을 측정하며 모집단 효능은 측정하지 않는다. E2 는 셀당 5 회 호출의 라이브 스크리닝으로, 고정되지 않은 revision, 제어되지 않는 seed label, 구성된 state 때문에 pilot-only 가 상한이다. E3 는 런타임 검색 없이 closed world 에서 작성된 typed-link proposal 만 채점한다. ENG1 은 같은 게이트를 Godot 4.7.1 slice 에 임베드하며, 플레이어는 거기서 보류를 규칙으로, commit 을 기여로 읽는다 (그림 3). 참여자, 감정 실험, live Python–Godot 왕복은 없다.

II. 관련 연구와 연구 공백

A. 인터랙티브 내러티브와 게임 에이전트

KNUDGE 는 퀘스트와 엔터티 명세에 근거한 분기형 NPC 대화를 평가한다 [3]. 지식 그래프 보조 퀘스트 생성 [4], 메모리-성찰-계획 에이전트 [11], 학습 가능한 역할 연기 에이전트

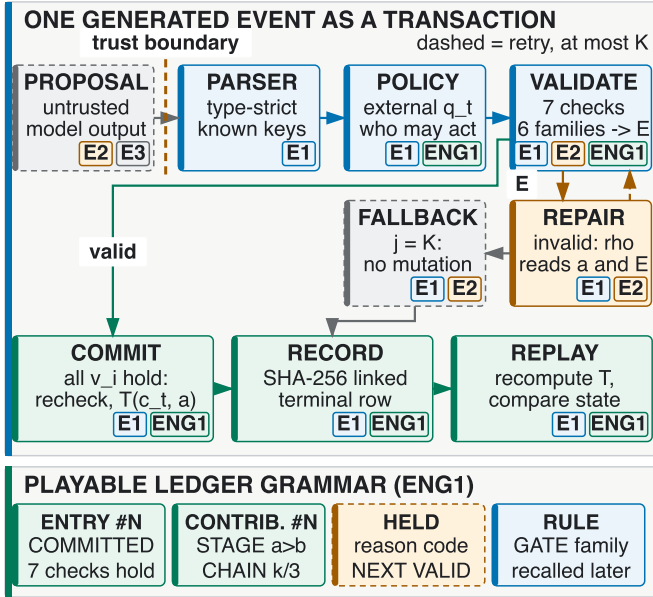


Fig. 1. 생성 이벤트 하나의 트랜잭션 경로다. typed candidate 만 신뢰 경계를 넘고, 외부 policy 와 c_t 를 여섯 상태 상대 계열의 검사 7 개가 확인한다. typed 오류 E 는 bounded ρ (점선) 또는 변경 없는 fallback 으로 이어지고, 통과 후로는 재검사-commit-record-replay 된다. 태그는 claim 이 아닌 측정 범위를 표시하며, 아래 strip 은 committed snapshot 에서 파생한 ledger 문법이다.

[12] 는 근거성과 개연성 있는 행동의 상호보완적 측면을 다룬다. 플레이어 대상 연구는 선호와 지각된 대화 품질을 평가하고 [13], [14], [15], 2025 년 프리프린트는 역할에 따른 안정성-즉흥성 trade-off 를 보고한다 [16]. 두 흐름 모두 하드 허용 가능성을 선택적 자연스러움 및 플레이어 경험 구성개념과 분리할 동기를 준다.

경험 주도 콘텐츠 생성 [17] 과 관찰자 표식 참여도 [18] 는 미래 적응 트랙의 동기가 된다. 최근 프리프린트는 vision-language model 의 참여도 추론 한계를 보고한다 [19]. 따라서 감정은 비권위적 맥락으로 유지되며 현재 구현 근거에는 포함되지 않는다.

B. 환경과 벤치마크

TextWorld 와 ScienceWorld 는 명시적인 인터랙티브 상태를 제공하며 [1], [2], MineDojo 는 개방형 지식 집약 embodied 과제를 추가한다 [20]. General Video Game AI 는 에이전트, 게임, 콘텐츠 트랙을 분리한다 [21]. AgentBench 는 이질적인 환경의 에이전트를 연구하고 [22], AgentBoard 는 과정 수준 진단을 추가하며 [23], SOTOPIA 는 사회적 상호작용을 평가하고 [24], BALROG 는 장기 언어 및 시각 게임 추론을 평가한다 [25]. 자동화된 플레이 스타일 에이전트는 상호보완적인 시험 선례이지만 인간 평가를 대체하지 않는다 [26]. 이 벤치마크는 미래 평가 설계에 정보를 주지만 트랜잭션 승인 경계를 규정하지는 않는다.

C. 제약, 신경-기호 검증, 검색

제약 디코더는 출력 구조를 개선한다 [5], [6], [7]. 반면 TRACE-RPG 는 구조를 독립적인 상태 상대 승인의 입력 조건으로 취급한다. 같은 generate-and-verify 분업은 게임 밖에서 확립되어 있다. LLM-Modulo 프레임워크는 모델이 제안한 계획을 외부의 sound critic 이 수락 또는 거부하게 하고 [27],

SayCan 은 언어로 제안된 skill 을 상태 의존 affordance 검사 뒤에만 허용한다 [28]. AgentSpec 은 LLM 에이전트의 런타임 규칙을 trigger-predicate-enforcement 삼중항으로 표현하고 [29], TRACE-RPG 은 술어를 게임 상태에 고정하며 수리 계약과 리플레이 가능한 레코드를 추가한다. Setting the DC 는 D&D 에이전트를 precondition 검사가 있는 tool call 에 근거시키는 반면 [30], TRACE-RPG 은 자유 텍스트 제안을 parsing 과 외부 policy 평가 뒤에만 허용한다. STORY2GAME 은 플레이에 쓰이는 action precondition 과 effect 를 LLM 스스로 생성하게 하지만 [31], 여기서는 policy 가 제안자와 독립적으로 저장된다. TRACE-RPG 은 이 critic 을 typed 반례를 반환하는 결정론적 상태 상대 commit gate 로 구체화한다. 게임 특화 비교 대상은 서로 다른 경계를 보여준다. PANGeA 는 LLM 이 설계자 규칙에 따라 콘텐츠를 판정하고 수정하도록 해 검증하고 [32], IVIE 는 기호 검증으로 인터랙티브 픽션 세계를 점진적으로 생성하며 [33], 같은 계보의 2026 년 후속 시스템은 LLM 을 저장된 world-state transformation 선택으로 제한해 [34] 구성상 일관성을 얻는다. 이 논문의 트랜잭션, 감사 레코드, 수리 연산자 질문은 별도의 근거 대상이다. 어느 비교도 우월성을 뜻하지 않으며, 유효성은 각 시스템이 인코딩한 술어에 한정된다.

G-Retriever, HippoRAG, Mem0 는 관련 하위 그래프 검색과 장기 메모리를 다루고 [8], [9], [10], Generative Agents 와 Character-LLM 은 행동 메모리와 persona 를 다룬다 [11], [12]. TRACE-RPG 은 향후 이러한 맥락을 사용할 수 있지만 직접적인 커밋 권한은 부여하지 않는다. 검색과 시간적 메모리 효익은 여기에서 측정하지 않는다.

D. 감사 가능성과 재현성

과정 지표는 중간 실패를 보존할 동기를 제공하고 [23], 강건한 평가는 소수 확률적 실행의 과도한 해석을 경고한다 [35]. 재현성과 환경 보고 지침은 동결된 산출물과 명시적 측정 경계를 뒷받침한다 [36], [37]. 인간 및 LLM 평가는 구성개념, 평가자, 편향 통제를 요구하며 [38], [39], [40], [41], 향후 비열등성과 참여자 분석에는 정당화된 margin 과 random-effects 구조가 필요하다 [42], [43]. Record-and-replay 는 경험 재사용을 위한 에이전트 패러다임으로 제안되었지만 [44], TRACE-RPG 은 승인된 상태 전이의 의미적 리플레이만 주장한다.

제한된 novelty 는 개별 구성요소의 발명이 아니라 구현된 결합이다. 선행 연구는 범용 에이전트의 런타임 강제 [29], 생성된 게임 세계의 기호적 상태 관리 [30], [31], 피드백 기반 수리 [45], 에이전트 record-and-replay [44] 를 각각 따로 보여준다. TRACE-RPG 은 이들의 교집합을 게임 상태 commit 경계에서 연구한다. 즉 parsing 된 제안을 독립적인 policy 와 결정론적 술어로 검사하고, bounded validator feedback 으로 선택적으로 수리하며, 의미적 리플레이를 위해 기록한다. 행동 개입(interposition) 자체는 오래된 기법이다. 내러티브 mediation 은 행동을 실행 전에 가로채고 [46], 고전 계획법은 precondition 과 effect 로 적용 가능성을 정의하며 [47], 런타임 강제는 허용된 전이만 통과시키고 [48], 트랜잭션 처리는 abort 시 이전 상태를 그대로 남기며 [49], 작성형 논리 기반 인터랙티브 드라마는 캐릭터의 발화를 가능하게 하는 사회적 관행을 작성했고 [50], 신념 인지 내러티브 계획법은 캐릭터가 아는 바를 추론한다 [51]. 후자는 본 논문의 knowledge/disclosure 술어가 훨씬 거칠게 인코딩하는 관심사다. C1-C2 는 strict contract 와 validate-repair-commit controller 를 이 계보 위에 배치하고, C3-C4 는 checksum-linked record, state-semantic replay, episode continuity, proposal trace 가 없는 사례까지 보존하

는 assignment-complete accounting 을 더하며, C5 는 실행 전 typed validator error 만 소비하는 수리 연산자를 분리한다. 보고서에 기능이 없다고 해서 모든 구현에 그 기능이 없다고 추론하지 않는다.

III. 문제 정의와 보장 경계

설명을 위해 시점 t 에서 validation 이 사용하는 controller 입력을 다음과 같이 분해한다.

$$c_t = (G_t, q_t), \quad (1)$$

여기서 G_t 는 typed snapshot field 를, q_t 는 해당 snapshot 과 함께 저장되는 외부 작성 action-policy, disclosure, quest-stage 제약을 나타낸다. 종단 record history $h_{<t}$ 는 WorldState snapshot 과 별도로 유지된다. Validation predicate 는 이를 읽지 않으며 구현은 log 에서 상태를 재구성하지 않는다. 검색, 내러티브 점수, 모델 rationale, 메모리 요약, 미래의 감정 추정 은 비권위적 proposal context 이다.

신뢰하지 않는 proposer 가 후보 a_t 를 출력한다. 타입 엄격 parsing 은 알 수 없는 top-level field 의 거부를 포함한 공유 candidate-key contract 를 먼저 강제하고 선언 스키마를 확립한다. validator 는 다음을 반환한다.

$$V(c_t, a_t) = (v_{\text{policy}}, v_{\text{pre}}, v_{\text{reach}}, v_{\text{know}}, v_{\text{disc}}, v_{\text{quest}}, E_t), \quad (2)$$

여기서 E_t 는 구조화 오류 집합이다. 상태 변경은 다음과 같다.

$$c_{t+1} = \begin{cases} T(c_t, a_t), & \bigwedge_i v_i = 1, \\ c_t, & \text{그 외.} \end{cases} \quad (3)$$

Controller 는 인코딩된 계약 아래 다음 구현 불변량을 주장한다. 파일럿은 이를 일반 정리로 제시하지 않고 deterministic valid, invalid, repair, fallback, replay, fault-injection fixture 로 해당 mechanism 을 시험한다. **I1 (소진 실패 불변성):** 예산 K 까지 반환된 어떤 후보도 검증을 통과하지 못하거나 proposal/controller 실행이 commit 전에 실패하면 반환 상태는 공급된 prior state 와 같다. **I2 (제한된 기록 시도):** 모든 proposal 과 repair callback 이 반환한다는 조건에서 repair budget $K \geq 0$ 은 최대 $K + 1$ 개의 candidate attempt 를 기록한다. 성공적인 commit 은 변경 직전 수락 후보에 대한 방어적 validation 을 한 번 추가한다. Runtime 은 callback deadline 을 강제하지 않으므로 I2 는 wall-clock 종료 보장이 아니다. **I3 (효과 권한):** 수락 후보는 ActionPolicy.allowed_effects 밖의 선언된 fact effect 를 CandidateAction.effects 에 포함하지 않으며, quest-stage target 도 해당 policy 에 의해 명시적으로 승인된다. **I4 (상태 의미적 리플레이):** 동결된 스키마와 결정론적 전이 아래 수락된 trace 는 명시된 terminal state 를 재구성한다.

이 주장된 불변량은 올바른 정책 작성, 선언된 후보 필드로의 완전하고 충실한 추출, 결정론적 소프트웨어 버전, callback 반환, 단일 프로세스 파일 소유권을 가정한다. 상태 의미적 리플레이는 기록된 구조화 후보와 전이를 재검증하지만, 중간 수리의 provenance, reasoning, prompt, 확률적 생성은 재현하거나 검증하지 않는다. 구조화 필드에서 누락된 자연어 함의는 보장 범위 밖이다.

IV. TRACE-RPG 아키텍처와 알고리즘

A. 신뢰 경계와 계약

그림 1는 모델 또는 recorded adapter 를 canonical owner 밖 에 둔다. Parser 는 알려진 필드의 모호한 숫자 또는 collection

TABLE II
인코딩된 Validator 경계

술어	거부 대상	실패 시 상태
Action policy	알 수 없는 action/type	변경 없음
Precondition	누락/거짓 requirement	변경 없음
Reachability	접근 불가 object	변경 없음
NPC knowledge	선언되지 않은 known fact	변경 없음
Disclosure	forbidden fact	변경 없음
Quest	부적격/후퇴 stage	변경 없음

```

1:  $a \leftarrow \text{Propose}(c); j \leftarrow 0$ 
2: repeat
3:    $(ok, E) \leftarrow V(c, a); \text{Record}(a, E)$ 
4:   if  $ok$  then
5:     assert  $V(c, a).ok$ ; return  $\text{Commit}(c, a)$ 
6:   if  $j = K$  then return  $\text{Fallback}(c)$ 
7:    $a \leftarrow \text{Repair}(c, a, E); j \leftarrow j + 1$ 
8: until terminal

```

Fig. 2. Validate-repair-commit 절차. 최대 $K + 1$ 개의 candidate attempt 를 기록하며, 성공 경로는 상태 변경 전 방어적 validation 을 수행한다.

type 을 거부하고 공유 key contract 에 따라 알 수 없는 top-level candidate field 도 거부한다. action policy 는 candidate 와 별도로 공급되며 required precondition 과 allowed effect 를 정의한다. Game bridge 는 observation, proposal, validation, rejection, fallback, commit 을 전달할 수 있는 versioned event interface 이며, 정식 상태 변경은 유효한 commit 만 승인한다. 이는 interface 를 분리하지만 상용 엔진으로의 portability 를 아직 입증하지 않는다. 표 II는 구현 검사 7 개를 여섯 의미 계열로 묶는다. Quest eligibility 와 quest-stage monotonicity 는 별도 검사로 유지된다. 이 인코딩 경계는 의미적 완결성 주장이 아니다.

B. Validate-Repair-Commit

그림 2은 제어기의 추가 패키지 없는 의사코드를 제시한다. “Record” 는 in-memory trace 표현에 추가하는 것을 뜻하며, 영속 쓰기는 종단 일관성 검사 뒤에만 수행한다.

검증 오류는 구조화된 error code 로 repair callback 에 노출된다. repair 는 상태를 직접 변경하지 않으며, 예산 소진 시 변경 없는 결정론적 fallback 을 반환한다. 그림 1이 종단 경로를 나타낸다. 파일럿은 구현된 두 callback 을 비교한다. 반례 유도 연산자는 직전 후보 a 와 validator 의 구조화 오류 집합 E 만 을 소비하며 권위 있는 상태는 결코 읽지 않는다.

$$\rho(a, E) = a \oplus \bigcup_{e \in E} \delta(e), \quad (4)$$

여기서 각 $\delta(e)$ 는 표 III의 오류 class 별 결정론적 edit 이고 $\oplus$ 는 그 field edit 들을 적용한다. 각 edit 은 오류 entity 당 최대 하나의 candidate field 만 수정하므로 시도당 변경 field 수는 $|E|$ 로 유계이고, 같은 오류 집합으로 ρ 를 두 번 적용하면 고정점이며, 같은 field 에서 동시에 요구되고 거부되는 entity 는 수리 불가로 선언되어 수정되지 않는다. 상태를 읽는 참조 callback 은 반대로 오류 집합을 폐기하고 정책이 규율하는 field(precondition, effect, 두 quest-stage field) 를 권위 있는 상태에서 직접 재구성한다. 이 callback 은 선언 무결성 field(required object, used fact, disclosed fact) 를 의도적으로 수정하지 않는다. 의존성이나 disclosure 의 선언을 지우는 것은 후보를 수리하는 것이 아니라 위반을 게이트 뒤로 세etak는

TABLE III
유도 연산자 ρ 의 오류 class 별 edit

Validator 오류 class	후보에 대한 edit $\delta(e)$
정책 precondition 누락	명명된 predicate 삽입
정책 effect 누락	명명된 effect 삽입
정책 effect 위반	명명된 effect 제거
불충족 precondition	명명된 predicate 제거
비인가 stage 변이	quest_stage_effect 초기화
알 수 없는 action type	no-op (등록된 대안 없음)
도달 불가 object; knowledge, disclosure, stage class	no-op (후보 국소 edit 은 선택이거나 상태 지식 필요)

TABLE IV
실험 레인 요약: 설계, 단위, 비교, 추론 상한 (실험 매트릭스에서 생성)

레인	설계	단위 / 예산	비교	상한
E1	오프라인 적합성: 작성 · 동결 fixture, 단일 world	gate 13; arm 당 repair 12; fault 10; adapter 7; guard 3	reject / blind / ρ / oracle	메커니즘 적합 성
E2	라이브 RQ2 스크리닝: hosted proposer 1 개, matched candidate	5 cell $\times$ 5 call; $K=1$	regime 별 ρ / blind	pilot-only 전 이 screen
E3	KG/온톨로지 시뮬레 이션: closed-world typed-link holdout	node 43; 참조 edge 106; typed 24; score 210	degree baseline / 고정 전략 6 종	구성 결과만
ENG1	Godot/Web 엔지니어 링: 엔진 로컬 policy mirror, 참여자 없음	fixture 4; check 52; smoke 8; archetype 5	작성 fixture 경 로	엔진 로컬 적 합성

E1-E3 와 ENG1 은 서로 대체할 수 없으며 문호가 다르다. 어떤 레인으로 모집단 효능, 모델 순위, 플레이에 경험, 런타임 검색 효력, 상용 엔진 성능을 확립하지 않는다.

것이기 때문이다. 이는 상태 판독 가능 수리에 대한 oracle 상한이며, 그 결과를 배포 가능한 수리 방법의 근거로 읽어서는 안 된다.

C. 무결성, 의미적 리플레이, 계수

각 result row 는 키 없는 SHA-256 무결성 체크섬을 가지며, proposal outcome 은 추가로 trace checksum 과 연결된다. 이 hash 들은 우발적이거나 시험 대상인 콘텐츠 변경을 탐지하지 만 signature, message-authentication code, writer identity 가 아니며 콘텐츠와 checksum 을 모두 다시 쓸 수 있는 행위자에 대한 보호도 아니다. 의미적 리플레이는 prior state 와 기록된 모든 candidate/validation pair 를 엄격히 parsing 하고, final 이 아닌 모든 attempt 가 invalid 임을 요구하며, terminal transition 을 재계산하고, 명시된 final state 와 비교함으로써 byte 무결성을 넘어서는. Episode replay 는 인접 JSONL record 사이의 상태 연속성도 요구한다. Replay 는 candidate $j + 1$ 을 만든 repair-generation 단계를 인증하거나 다시 실행하지 않는다.

Terminal record class 는 commit, symbolic fallback, adapter failure, controller failure 이다. Controller-failure row 는 동결된 assignment denominator 에 남고 상태를 변경하지 않지만, candidate outcome 이 생성되지 않았다면 완성된 proposal trace 가 없어도 된다. 따라서 result completeness 가 모든 terminal row 의 trace 존재를 뜻하지 않는다. 실행 전 assignment key 는 (run, arm, scenario, seed, model, revision, $h_{\text{controller}}$, h_{input} , h_{prior}) 이다. 집계는 중복, 누락, 예기치 않은 key 를 거부하고 symbolic, adapter, controller failure 를 각각 보고한다.

V. 실험 레인과 근거 경계

표 IV는 어떤 count 를 읽기 전에 각 레인의 설계, 단위, 비교, 상한을 고정한다.

A. E1 오프라인 적합성 설계

파일럿은 다음을 묻는다. (PQ1) 유효 후보는 commit 되고 명명된 invalid-policy 사례는 변경 없이 유지되는가? (PQ2) 동결된 모든 repairability class 의 작성 사례에서 rejection-only, 블라인드 동일 후보 retry, 반례 유도 연산자 ρ , 상태를 읽는 참조 callback 이 예상 결과를 만드는가? (PQ3) 동결된 checksum, state-semantic, linkage, continuity mutation 이 지정된 검사 연산에서 거부되고, 두 번째 파일 append 의 주입된 실패가 예상 pair-write rollback 검사를 작동시키는가? (PQ4) 엄격한 adapter 와 assignment manifest 가 할당 사례마다 정확히 하나의 분류된 terminal row 를 유지하거나 fail closed 하는가? (PQ5) proposal 및 replay parser 가 그 외에는 완전하고 유효한 candidate 에 unknown top-level key 하나를 추가한 사례를 모두 거부하는가?

사례는 의도적으로 설계된 적합성 픽스처이며 모델 또는 게임 모집단의 표본이 아니다. 네 도메인은 validator/state isolation, repair-arm 비교, record/trace fault injection, adapter/accounting contract 이다. 모든 repair fixture 는 실행 전에 동결된 repairability class 를 선언한다. *guided-repairable*(구조화 오류 payload 만으로 ρ 에 충분한 정보가 담김), *oracle-only*(수리에 권위 있는 상태 판독이 추가로 필요함), *irreparable*(구현된 어떤 arm 도 commit 할 수 없음)이며, class 별 arm 당 기대 결과는 설계 픽스처 계약의 일부다. 반복된 결정론적 seed 가 아니라 고유 사례가 기술적 denominator 를 구성한다. 모든 기대값은 실행 전에 명시한다.

측정치는 정확한 count 이다. expected/observed validator-class agreement, rejected 또는 controller-failure 사례의 state mutation, repair arm 과 repairability 별 commit 및 attempt 결과, 지정된 연산에서 거부된/사전 명세된 detectable fault, replayed/committed trace, rejected/injected manifest violation, 사전 명세한 unknown-key 음성 회귀에 대한 proposal/replay 거부 일치를 센다. 마지막 측정치는 결정론적 parser-contract parity 이며 실증적 safety outcome 이 아니다. Exception class 만으로 안정적인 detector layer 를 식별할 수 없으므로 detector 귀속 결과는 보고하지 않는다. 의도적으로 설계된 픽스처에 p -value 나 모집단 confidence interval 을 부여하지 않는다. Adapter 사례가 지닌 latency 및 token 상수는 입력 스키마가 JSON-Schema const 로 고정한 합성값이다. 이 값들은 계수 필드 전파만 확인하며 provider, runtime, device 에 대한 측정값이 아니다. 동결된 오프라인 packet 은 fixture, assignment, schema, trace 를 포함한 result, 코드 revision, checksum 을 결합하며, 보고는 재현성과 측정 경계 지침을 따른다 [36], [37].

B. E2 라이브 RQ2 스크리닝

라이브 확장은 블라인드 동일 후보 retry 와 ρ 를 동일한 $K=1$ 에서 비교한다. 각 셀에서 seed 로 구분한 hosted proposer gpt-5.6-sol 호출 5 회가 후보 하나씩을 만들고, 두 arm 은 같은 최초 후보를 받아 호출 내 제안 차이를 제거한다. 조건은 policy 공개 여부와 자명한 효과 0 경로의 존재를 바꾼다. 동결된 packet 은 현재 셀과 폐기된 v1 진단을 함께 보존하고 모든 영수증을 SHA-256 으로 결합하며 pilot-only 로 표시된다. Seed label 은 hosted sampler 를 제어하지 않고, hosted revision 은 고정되지 않았으며, token 회계는 없고, 구성 상태

는 모집단 표본이 아니다. 따라서 p -value, confidence interval, 모델 순위 없이 정확한 기술 count 와 실패 class 만 보고한다.

C. E3 KG/온톨로지 링크 제안 시뮬레이션

세 번째 레인은 권한을 가진 WorldState 와 분리된 engineering-only closed-world 시뮬레이션이다. graph node 43 개와 reference edge 106 개, 검토된 typed edge 24 개가 type/domain/range 검사와 competency question 6 개를 통과하며, 작성형 질의 6 개를 고정 전략 7 개로 채점했다. Retained typed-lexical 전략 (S2) 은 작성된 holdout 6/6 을 복원했고 ($P = R = F_1 = 1.000$, realistic-tie MRR = 0.944, Brier = 0.131, Sem@3 = 1.000), degree baseline 은 하나도 복원하지 못했다. 이는 구성 결과이며 OWL/SHACL 적합성, 의미 완전성, runtime retrieval, 어떤 효능 주장의 근거도 아니다.

VI. 파일럿 결과

A. 구조 필드 게이트 적합성

파일럿 loader 는 구현된 validator code 를 정확히 한 번씩 고립시키는 fixture 집합과 유효 control 1 개만을 허용한다. 따라서 13/13 일치와 구현된 오류 코드 12 종의 관찰은 측정된 성공률이 아니라 하니스의 구성 불변량이다. 이 실행이 보여주는 것은 loader 가 강제하지 않는 부분, 즉 고립된 음성 fixture 12 개가 각각 다른 코드가 아니라 작성된 기대 코드와 일치했고 비 commit 경로가 전체 정준 상태를 유지했다는 관찰이다. 이는 단일 world state 에서 저자가 설계한 구조 필드 oracle 에 대한 구현 적합성이며 독립 의미 판정의 정확도가 아니다.

B. 설계 픽스처에서의 수리 arm 비교

처음부터 유효하지 않은 설계 case 12 개는 guided-repairable 5 개, oracle-only 1 개, irreparable 6 개의 동결된 repairability class 로 분할된다. Rejection-only 와 블라인드 동일 후보 retry 는 각각 0/12 commit 이었다. 반례 유도 연산자 ρ 는 guided-repairable case 5/5 를 commit 했고, 설계상 oracle-only 0/1, irreparable 0/6 였다. 모든 ρ commit 은 candidate field 를 평균 1.0 개 수정했으며, 실패한 모든 arm 실행은 정준 상태를 변경 없이 유지했다. 상태를 읽는 참조 callback 은 guided-repairable 5/5 와 oracle-only 1/1 를 commit 했고 irreparable 은 0/6 였다. 이 정확한 count 는 세 메커니즘을 설계 픽스처 위에서 분리한다. 블라인드 retry 는 반례를 commit 으로 전환하지 못하고, ρ 는 오류 payload 에 충분한 정보가 담긴 case 만 전환하며, oracle 은 상태 지식이 필요한 case 를 추가로 전환한다. commit 된 guided case 5 건이 처음 받은 validator code 는 각각 POLICY_PRECONDITION_OMISSION+UNSATISFIED_PRECONDITION, POLICY_EFFECT_OMISSION, POLICY_EFFECT_VIOLATION, UNSATISFIED_PRECONDITION, POLICY_QUEST_STAGE_EFFECT_VIOLATION 이며, 각 case 는 표 III의 해당 edit $\delta(e)$ 만으로 유효해졌다. 이는 동결 픽스처에 대한 연산자 적합성이며, 어떤 live model 의 수리 품질도 아니다.

C. 무결성, 경계, 배정 회계

현재 명세가 검출 가능하다고 지정한 10 개 fault fixture 는 모두 지정된 검사 연산에서 거부됐다 (10/10). 이 파일럿은 안정적 typed detector code 까지 대조하지 않으므로 detector layer 귀속을 입증하지 않는다. 별도로, 동일하게 기록된 유효 수리 앞의 무효 선행 후보를 다른 무효 후보로 치환하고 체크섬

TABLE V
수리 arm 별 repairability class 정확 집계 (설계 픽스처)

Arm	정보원	Commit / 사례			평균 수정 ^e
		G-rep. ^f	O-only ^f	Irrep. ^f	
Rejection-only ($K=0$)	없음	0/5	0/1	0/6	—
블라인드 retry	없음	0/5	0/1	0/6	—
반례 유도 ρ	오류 집합 E	5/5	0/1	0/6	1.0
상태 판독 oracle	권위 상태 + 정책	5/5	1/1	0/6	1.0

^eCommit 된 수리의 변경 candidate field 평균 개수 (최소성 proxy); 실패한 arm 실행은 상태를 변경하지 않았다. ^fCommit/사례 수. G-rep. = guided-repairable, O-only = oracle-only, Irrep. = irreparable(동결 repairability class).

TABLE VI
동결 파일럿 검사와 종단 결과 (행 간 직접 비교 불가)

검사 또는 종단군	정확 결과	해석 범위
게이트 fixture 일치	13/13	저자 oracle 일치
알려진 경계 허용	2/2	인코딩 한계, safety pass 아님
닫힌 unknown-key 거부	1/1	proposal/replay parser parity
탐지 가능 무결성 결함 거부	10/10	지정된 검사 연산
알려진 provenance 경계 허용	1/1	repair 생성 출처 미인증
Adapter 종단 class	1 commit; 1 fallback; 5 실패 / 배정 7	배정 7 건의 분류 완전성
Manifest guard 거부	3/3	주입 결함 fail-closed

행마다 분포와 성공 방향이 다르므로 비율처럼 순위를 비교하지 않는다. 모든 값은 설계 픽스처의 정확한 감사 사실이다.

을 재계산한 알려진 provenance 경계는 예상대로 재생을 통과했다 (1/1 미검출). 따라서 키 없는 체크섬을 재계산할 수 있는 작성자를 방어하지 않으며, 재생은 repair 생성 연산의 출처를 인증하지 않는다. 열린 경계 sentinel 두 건—자연어 disclosure 미추출과 required object 의 후보 · 정책 동시 누락—은 2/2 가 의도대로 인코딩상 허용됐고 safety pass 는 0 건이었다. 과거 unknown top-level field 허용 경계는 이후 닫힌 음성 회귀로 재 분류되었으며, 완전한 12-field candidate 에 unknown key 하나를 추가한 fixture 를 proposal-replay parser 모두 구체적으로 거부했다 (1/1). 이는 parser rejection parity 이지 의미 안전성의 증거가 아니다. Adapter/accounting 7 개 배정에서는 commit 1, 기호 fallback 1, adapter failure 5 였다. 합성 telemetry 필드는 2/7 배정에서 채워져 전파됐으며, 이는 측정값이 아니라 회계 필드 전파 확인이다. 배정 guard 3/3 가 중복 · 누락을 거부했다. 이 수치는 모두 합성 offline fixture 의 raw count 다.

표 V와 VI은 정확한 설계 픽스처 count 를 보고한다. 수리 표는 동결된 repairability class 별로 네 arm 을 분리하며, 어느 표도 모집단 또는 live-model 추정을 지지하지 않는다.

D. 라이브 스크리닝 파일럿 (pilot-only)

별도 스크리닝 파일럿은 hosted proposer gpt-5.6-sol 을 호출했다. 셀마다 seed 로 구분한 제안 호출 5 회와 수리 예산 $K = 1$ 을 사용했고, 각 호출의 최초 후보 하나를 블라인드 동일 후보 retry 와 ρ 가 공유했다. Hosted revision 은 고정되지 않았고 token 회계는 제공되지 않았으며 seed 는 hosted sampler 를 제어하지 않으므로 이 호출들은 독립 무작위 표본이 아니라 준반복이다. 따라서 추론 통계나 모델 순위를 보고하지 않는다.

표 VII에서 유도 우위는 의도적으로 guided-repairable 오류가 나오도록 구성한 signal-repair state(Signal-v2) 의 policy-blind 셀에서만 나타났다. 최초 후보 5/5 가 모두 무효였고 ρ

TABLE VII
라이브 스크리닝 파일럿 정확 집계 (셀당 제한 5 회, $K=1$)

기저 / 조건	최초 유효	ρ vs blind commit	Δ	최초 오류
동결 / 공개	5/5	5/5 vs 5/5	+0	—
동결 / 비공개	5/5	5/5 vs 5/5	+0	—
동결 / 목표-비공개	0/5	0/5 vs 0/5	+0	QSR
Signal-v2 / 공개	5/5	5/5 vs 5/5	+0	—
Signal-v2 / 비공개	0/5	5/5 vs 0/5	+5	PEO+PEV+PPO

QSR = quest-stage regression, PEO = policy-effect omission, PEV = policy-effect violation, PPO = policy-precondition omission. Δ 는 동일 후보에서 ρ commit 수 minus blind commit 수다. 모든 수치는 pilot-only 이며 추론 통계가 아니다. 폐기된 v1 진단은 표에서 제외한다.

(a) 보류된 제안: 게이트 계열, 변경 없는 상태, 다음 유효 항목, 학습된 규칙

```
[P] PROPOSAL ... Request disclosure of the sealed fact now
'Mira studies the dark tower and asks you to recover the signal lens first.'
[H] HELD - This request cannot be answered yet. The ledger defers it.
[V] GATE DISCLOSURE | state unchanged
[N] NEXT VALID ENTRY: Speak with Captain Mira at the end of the dock.
[V] RULE LEARNED | DISCLOSURE: Some facts stay sealed for good; the ledger never records them.
```

(b) 커밋된 제안: 항목, 기여 절, 해제된 행동

```
[C] ENTRY #2 | COMMITTED - Signal lens installed. An authorized lead is now available.
[C] CONTRIBUTION #2 | Lead authorized; Harbor signal restored | STAGE 1>2 | CHAIN 2/3
[N] UNLOCKED | Ask Captain Mira for the authorized lead.
```

(c) 세 번의 커밋 뒤 엔드카드 영수증

```
INVESTIGATOR'S CONTRIBUTION
#1 Signal lens secured | STAGE 0>1
#2 Lead authorized; Harbor signal restored | STAGE 1>2
#3 Tide-marks lead recorded | STAGE 2
RULES LEARNED 0 | none

ENTRIES 3 | HOLDS 0 | FINAL STAGE 2
VALIDATOR RECEIPT | STATE HASH 4b2310173dc05907...
HOLDS BY GATE | none
```

Fig. 3. 플레이어가 Godot slice 에서 읽는 commit gate. 작성된 fixture 하나의 scripted presentation stage 세 개에서 잘라낸 ledger(엔지니어링 작업 캡처, 1280×720; 예시일 뿐 근거가 아니다). (a) 금지된 disclosure 가 보류된다. ledger 는 게이트 계열을 이름 짓고, canonical state 가 변경되지 않았음을 밝히며, 다음 유효 항목을 가리키고, 보류를 플레이어가 재사용할 수 있는 규칙으로 바꾼다. (b) 유효한 제안이 커밋된다. 항목, committed snapshot 의 차분만으로 파생된 기여 절 (추가된 fact, 퀘스트 단계 1>2, case chain 2/3), 그리고 그것이 해제하는 행동이 이어진다. (c) 엔드카드는 기여 절 세 개, 보류 횟수, 그리고 리플레이가 보고하는 중단 state hash(4b2310...) 를 담은 validator 영수증을 나열한다. 모든 줄은 committed snapshot 의 presentation-only 판독이며 (그림 1 아래 strip), oracle label, 숨겨진 fact, live model, participant 는 개입하지 않는다. 패널은 usability, immersion, efficacy 어느 주장도 뒷받침하지 않는다.

는 5/5, 블라인드는 0/5 를 commit 했다. 나머지 현재 셀 4 개는 유도 우위를 보이지 않았다. 세 셀은 최초 후보가 이미 유효했고, 동결 기저의 goal-directed 셀은 수리 불가 quest-stage regression 이었다. 현재 셀의 모든 noncommit arm 결과 15/15 가 prior-state hash 를 보존했다. 이 결과는 해당 오류 regime 에서의 mechanism transfer 만 지지하며 모집단 효능, 모델 승격, 표본 효율 일반화는 지지하지 않는다. 따라서 두 스크리닝 관찰은 pilot-only 이며 확증적 전이 claim 은 열린 채로 남는다. 폐기된 v1 진단에서는 동반 오류가 최대 3 개에서 수리 불가 knowledge 오류 1 개로 줄었지만 commit 되지 않아, 오류 하나라도 수리 불가이면 부분 수리가 합성되지 않음을 보여 주었다.

보고 경계는 명시적이다. 동결 fixture 와 지정 검사 연산이 허용한 row 만 오프라인 표에 들어가고, 라이브 표는 자체 hash-bound packet 을 통해 도착하며 동결 오프라인 denominator 에는 들어가지 않는다. 각 lane 이 측정하지 않은 항목은 표 IV가 나열한다.

ENG1. Godot 4.7.1 slice 에 임베드된 같은 게이트의 두 번

째 구현이 작성된 정식 trace 를 리플레이했다. 보류된 금지 disclosure 는 이전 state hash 를 보존했고 승인된 commit 은 이를 전진시켰으며, 중단 hash 는 오프라인 runner 와 정확히 일치했다. 그림 3은 이 두 결과와 에피소드 끝에서 플레이어가 읽는 내용을 예시한다. 보류는 게이트 계열과 다음 유효 항목을 이름 짓고, commit 은 기여 절을 덧붙이며 다음 행동을 해제하고, 엔드카드는 hash 가 담긴 영수증을 싣는다. 적합성 근거는 리플레이된 trace 와 그 결정론적 평가이지 이 픽셀이 아니다. slice 에는 committed fact 만 읽는 case-file board, opt-in hint ladder, just-in-time coach card, 마무리 chain-assembly prompt 가 있고, 동반 Web dashboard 는 제안마다 엔진이 실제로 실행한 검사, 보류를 만든 검사, 도달하지 못한 in-scope 검사, 해당 연산 범위 밖의 계열을 엔진 코드와 불일치하면 fail-closed 되는 mirror 로부터 그대로 보여 준다. 이는 scripted fixture 하나의 엔진 로컬 적합성이다 (core 검사 52/52 와 별도 집계된 4 개 확장 검사). live model 이나 Python transport, participant input, narrative-quality result 는 없으며, 판독은 committed snapshot 에서만 파생되어 (그림 1 아래 strip) usability 나 efficacy 근거가 아니다.

VII. 논의

A. 확립 가능한 범위

설계 파일럿은 동결된 사례에 대해서만 mechanism behavior 를 확립할 수 있다. 인코딩된 검사가 상태를 격리하는지, bounded repair 와 fallback 이 선언 경로를 따르는지, 명명된 corruption 이 지정된 검사 연산에서 거부되는지, assigned row 가 집계를 위해 분류된 상태로 남는지를 평가한다. 거부를 안정적인 typed detector code 에 귀속하지는 않는다.

오프라인 수리 비교의 정직한 해석은 guided-대-oracle 경계다. 상태를 읽는 callback 이 상한으로 남는 이유는 권위 있는 상태를 참조할 수 있어서 ρ 가 구성상 전환할 수 없는 oracle-only 사례를 전환하기 때문이다. 반례 유도 연산자는 validator 의 구조화 반례만 소비하면서 guided-repairable class 에서 oracle 과 정확히 일치한다. 이어 라이브 스크리닝 확장은 이 mechanism 이 hosted proposal 로 전이되는지를 탐색했다. 구성 상태가 모든 제안을 guided-repairable class 로 유도했을 때만 ρ 와 blind retry 가 분리됐고, 최초 유효 및 guided-irreparable regime 은 귀무였다. 이 regime 의존성, 고정되지 않은 hosted revision, 준 반복 호출 때문에 일반 표본 효율이나 모델 능력을 주장할 수 없다. Feedback 을 소비하는 수리 루프에는 선례가 있다. Self-Refine 은 같은 신뢰하지 않는 모델이 생성한 언어를 feedback 으로 되먹이고 [45], Reflexion 은 시도 간 verbal self-reflection 을 저장하며 [52], Self-Debugging 은 신뢰되는 전체 프로그램 runtime 의 실행 결과를 소비한다 [53]. ρ 는 channel 과 권위에서 다르다. Feedback 은 실행 전에 결정론적 validator 가 계산한 typed 오류 집합이고, 실패한 모든 live arm 은 상태를 격리한다. 이름은 명세의 전체 상태가 아니라 verifier 의 반례가 다음 후보를 이끄는 counterexample-guided inductive synthesis 를 따르며 [54], 여기서 verifier 는 commit gate 자체다.

B. 선행연구와의 관계

TRACE-RPG 은 constrained decoder [5], [6], [7], 검색 및 메모리 구성요소 [8], [9], [10], 실행 가능한 환경과 benchmark [1], [2], [22], [25] 를 대체하지 않고 보완한다. Process-linked trace 는 AgentBoard 의 측정 관심사를 구현하지만 [23] 우월성의 근거는 아니다. 확증 연구, 즉 독립 semantic oracle 과 실제 blind-retry comparator 를 갖춘 사전등록 다중 모델 비교, 인코

당된 validity 를 맞춘 retrieval 및 memory ablation, 정당화된 검정력과 mixed-effects 분석을 갖춘 blinded matched-validity 인간 연구는 여전히 본 논문의 범위 밖이며, 스크리닝 파일럿은 어느 연구도 대체하지 않는다 [38], [35], [43].

VIII. 타당성 위협, 윤리, 산출물 범위

근거에는 명시적 경계 9 개가 있다. **L1:** encoded predicate 는 structured field 에서 빠진 의미를 발견하지 못하며, 상태 일관성 만으로는 행동 일관성을 확립하지 못한다 [55]. 서사 품질은 추가로 측정하지 않은 구성개념이다. **L2:** authored world 하나와 constructed live-state variant 하나로 cross-world, cross-policy, 규모 일반화를 말할 수 없다. **L3:** state-reading callback 은 oracle 이며 offline class 와 regime-dependent screening 은 일반 repair 품질이나 sample efficiency 를 확립하지 않는다. **L4:** hosted proposer 하나는 revision 고정, token 회계, sampling 을 제어하는 seed 가 없고 live Python-Godot authorization round trip 도 없다. **L5:** participant 와 affect data 가 없다. game packet 은 엔진 로컬 적합성과 측정 준비가 된 protocol 일 뿐이며 comprehension, agency, enjoyment, immersion, player benefit 에 대한 어떤 주장도 지지하지 않는다. **L6:** unkeyed hash 는 integrity 만 제공하며 writer authentication 이나 adversarial tamper resistance 는 제공하지 않는다. **L7:** record consistency 는 single writer 를 가정하며 concurrent-writer safety 는 미검 증이다. **L8:** engine evidence 는 startup-dominated sample 이 16.7 ms 를 넘는 Godot fixture 하나이며 portability 나 real-time 결과가 아니다. **L9:** 선택된 fixture 와 screening cell 은 population, safety-rate, model-ranking 추론을 지지하지 않는다. Internal validity 는 독립 policy expectation 및 fixture leakage 부재에, reproducibility 는 고정 software 와 artifact 에 의존한다.

파일럿은 participant 또는 personal telemetry 를 처리하지 않는다. 향후 human 및 affect 연구에는 해당 review, informed consent, compensation disclosure, minimization, retention/deletion rule, opt-out 이 필요하다. LLM judge 를 사용 하더라도 automated 및 crowd judgement 모두 calibration 과 bias control 이 필요하므로 보조 수단으로 유지한다 [38], [39], [40], [41].

IX. 결론

TRACE-RPG 은 작성된 candidate-event payload 를 encoded 기호 commit gate 를 통해 canonical mutation 으로부터 격리하고, linked record 와 state-semantic replay 로 terminal outcome 을 감사 가능하게 만든다. 결정론적 오프라인 파일럿은 동결된 작성 fixture 에서 mechanism 적합성을 확립한다. 별도 라이브 스크리닝 파일럿은 구성한 수리 가능 regime 에서만 guided 5/5 대 blind 0/5 분리를 관찰하고 다른 regime 에서는 귀무였으므로 pilot-only 로 남는다. Cross-model superiority, player-experience benefit, retrieval 또는 memory effectiveness, affect efficacy, commercial-engine performance 는 명시적인 확증 확장으로 남는다. 이미 플레이어에게 달는 것은 게이트 자체다. Godot slice 에서 보류는 규칙으로, commit 은 기여로 읽히며, 모든 줄은 committed snapshot 에서만 파생 된다.

AI 사용 고지

Large language model 은 prose, code 와 test drafting, citation-identity check 를 보조했고, hosted model 은 별도 표

기된 live-screening candidate 를 생성했다. 모든 보고 count 는 deterministic runner 또는 hash-bound receipt 에서 온다. playable slice 의 UI 아트 (초상, 인장, 패널 텍스트, case-file board, evidence icon, guide vignette) 와 prop mesh 하나 (evidence desk) 는 파일별 provenance 를 갖춘 curated AI 생성 asset 이며, 그림 3의 모든 상태 값, hash, ledger 줄은 committed snapshot 에서 엔진이 렌더링한 것이다. 저자는 claim 을 source artifact 에 대조했고 내용에 대한 책임을 진다.

데이터 및 코드 가용성

Repository 에는 implementation, build source, 동결된 of-line 및 live packet, 결정론적 평가를 갖춘 playable Godot slice, 그리고 표 I과 IV를 생성하는 기계 검증 기여 · 참조 주제 · 실험 매트릭스가 들어 있다. Reviewer archive 나 DOI deposit 은 주장하지 않는다.

References

- [1] M.-A. Côté, Á. Kádár, X. Yuan *et al.*, “TextWorld: A learning environment for text-based games,” in *Computer Games*, ser. Communications in Computer and Information Science. Springer International Publishing, 2019, vol. 1017, pp. 41–75.
- [2] R. Wang, P. Jansen, M.-A. Côté, and P. Ammanabrolu, “ScienceWorld: Is your agent smarter than a 5th grader?” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022, pp. 11 279–11 298.
- [3] N. Weir, R. Thomas, R. d’Amore, K. Hill, B. Van Durme, and H. Jhamtani, “Ontologically faithful generation of non-player character dialogues,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024, pp. 9212–9242.
- [4] T. Ashby, B. K. Webb, G. Knapp, J. Searle, and N. Fulda, “Personalized quest and dialogue generation in role-playing games: A knowledge graph- and language model-based approach,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, 2023, pp. 1–20.
- [5] S. Geng, M. Josifoski, M. Peyrard, and R. West, “Grammar-constrained decoding for structured NLP tasks without finetuning,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023, pp. 10 932–10 952.
- [6] L. Beurer-Kellner, M. Fischer, and M. Vechev, “Prompting is programming: A query language for large language models,” *Proceedings of the ACM on Programming Languages*, vol. 7, no. PLDI, pp. 1946–1969, 2023.
- [7] T. Scholak, N. Schucher, and D. Bahdanau, “PICARD: Parsing incrementally for constrained auto-regressive decoding from language models,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 9895–9901.
- [8] X. He, Y. Tian, Y. Sun *et al.*, “G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering,” in *Advances in Neural Information Processing Systems*, vol. 37. Neural Information Processing Systems Foundation, 2024, pp. 132 876–132 907.
- [9] B. Jiménez Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, “HippoRAG: Neurobiologically inspired long-term memory for large language models,” in *Advances in Neural Information Processing Systems*, vol. 37. Neural Information Processing Systems Foundation, 2024, pp. 59 532–59 569.
- [10] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, “Mem0: Building production-ready AI agents with scalable long-term memory,” in *ECAI 2025*, ser. Frontiers in Artificial Intelligence and Applications, vol. 413. IOS Press, 2025, pp. 2993–3000.
- [11] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, “Generative agents: Interactive simulacra of human behavior,” in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2023, pp. 1–22.
- [12] Y. Shao, L. Li, J. Dai, and X. Qiu, “Character-LLM: A trainable agent for role-playing,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023, pp. 13 153–13 187.
- [13] N. Akoury, Q. Yang, and M. Iyyer, “A framework for exploring player perceptions of LLM-generated dialogue in commercial video games,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 2023, pp. 2295–2311.
- [14] M. Hochreiter, S. Kriglstein, and G. Wallner, “Beyond pre-defined scripts: Player perceptions on generative non-player character dialogues,” in *Proceedings of the 31st International Conference on Intelligent User Interfaces*. ACM, 2026, pp. 2004–2018.
- [15] M. Yin, H. Zu, W. Gu *et al.*, “How contextualized generative AI shapes player experience in games,” *Entertainment Computing*, vol. 58, p. 101194, 2026, volume 58 (September 2026) assigned on the publisher record, verified 2026-09-03.
- [16] V. Figueiredo and D. Elumeze, “Symbolically scaffolded play: Designing role-sensitive prompts for generative NPC dialogue,” 2025, preprint; no archival venue or publisher DOI verified as of 2026-09-03.
- [17] G. N. Yannakakis and J. Togelius, “Experience-driven procedural content generation,” *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011.

- [18] D. Melhart, D. Gravina, and G. N. Yannakakis, “Moment-to-moment engagement prediction through the eyes of the observer: PUBG streaming on Twitch,” in *FDG '20: Proceedings of the 15th International Conference on the Foundations of Digital Games*. ACM, 2020, pp. 1–10.
- [19] Z. Wang, Q. Guo, R. Singh, and X. Hu, “Do vision language models understand human engagement in games?” 2026, preprint; no archival venue or publisher DOI verified as of 2026-09-03.
- [20] L. Fan, G. Wang, Y. Jiang *et al.*, “MineDojo: Building open-ended embodied agents with internet-scale knowledge,” in *Advances in Neural Information Processing Systems*, vol. 35. Neural Information Processing Systems Foundation, 2022, pp. 18 343–18 362.
- [21] D. Pérez-Liébana, J. Liu, A. Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, “General video game AI: A multitask framework for evaluating agents, games, and content generation algorithms,” *IEEE Transactions on Games*, vol. 11, no. 3, pp. 195–214, 2019.
- [22] X. Liu, H. Yu, H. Zhang *et al.*, “AgentBench: Evaluating LLMs as agents,” in *International Conference on Learning Representations*, 2024, pp. 52 989–53 046, peer-reviewed official proceedings record; no publisher DOI assigned.
- [23] C. Ma, J. Zhang, Z. Zhu *et al.*, “AgentBoard: An analytical evaluation board of multi-turn LLM agents,” in *Advances in Neural Information Processing Systems*, vol. 37. Neural Information Processing Systems Foundation, 2024, pp. 74 325–74 362.
- [24] X. Zhou, H. Zhu, L. Mathur *et al.*, “SOTOPIA: Interactive evaluation for social intelligence in language agents,” in *International Conference on Learning Representations*, 2024, pp. 40 975–41 019, peer-reviewed official proceedings record; no publisher DOI assigned.
- [25] D. Paglieri, B. Cupial, S. Coward *et al.*, “BALROG: Benchmarking agentic LLM and VLM reasoning on games,” in *International Conference on Learning Representations*, 2025, pp. 96 666–96 702, peer-reviewed official proceedings record; no publisher DOI assigned.
- [26] P. Le Pelletier de Woillemont, R. Labory, and V. Corruble, “Automated play-testing through RL based human-like play-styles generation,” *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 18, no. 1, pp. 146–154, 2022.
- [27] S. Kambhampati, K. Valmeekam, L. Guan *et al.*, “Position: LLMs can’t plan, but can help planning in LLM-Modulo frameworks,” in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 22 895–22 907.
- [28] B. Ihter, A. Brohan, Y. Chebotar *et al.*, “Do as I can, not as I say: Grounding language in robotic affordances,” in *Proceedings of the 6th Conference on Robot Learning (CoRL 2022)*, ser. Proceedings of Machine Learning Research, vol. 205. PMLR, 2023, pp. 287–318.
- [29] H. Wang, C. M. Poskitt, and J. Sun, “AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents,” 2025, preprint (v3); the arXiv record states acceptance at ICSE 2026, but no proceedings DOI was resolvable as of 2026-09-03.
- [30] Z. Zeng, S. Li, J. Xi, A. Zhu, and P. Ammanabrolu, “Setting the DC: Tool-grounded D&D simulations to test LLM agents,” in *NeurIPS 2025 Workshop on Generative and Protective AI for Content Creation (GenProCC)*, 2025, non-archival workshop paper; OpenReview and its PDF list Zeng first and mark Zeng and Li as equal contributors, while NeurIPS and GenProCC venue lists retain a Li-first display; verified 2026-09-03.
- [31] E. Zhou, S. Basavatia, M. Siam, Z. Chen, and M. O. Riedl, “STORY2GAME: Generating (almost) everything in an interactive fiction game,” 2025, preprint; no archival venue or publisher DOI verified as of 2026-09-03.
- [32] S. Buongiorno, L. Klinkert, Z. Zhuang, T. Chawla, and C. Clark, “PANGeA: Procedural artificial narrative using generative AI for turn-based, role-playing video games,” *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 20, no. 1, pp. 156–166, 2024.
- [33] M. Vaucher, S. Silveira, S. Góngora, and L. Chiruzzo, “IVIE: A neuro-symbolic approach to incremental and validated generation of interactive fiction worlds,” in *Proceedings of the 17th International Conference on Computational Creativity (ICCC'26)*. Association for Computational Creativity, 2026, open-access archival proceedings published 2026-08-19 (ISBN only, no DOI), verified 2026-09-03; preprint arXiv:2606.13348.
- [34] S. Góngora, L. Chiruzzo, G. Méndez, and P. Gervás, “World-state transformations for neuro-symbolic interactive storytelling,” 2026, preprint; no archival venue or publisher DOI verified as of 2026-09-03.
- [35] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare, “Deep reinforcement learning at the edge of the statistical precipice,” in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 29 304–29 320, peer-reviewed official proceedings record; no publisher DOI assigned.
- [36] J. Pineau, P. Vincent-Lamarre, K. Sinha *et al.*, “Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program,” *Journal of Machine Learning Research*, vol. 22, no. 164, pp. 1–20, 2021, peer-reviewed official journal article; no publisher DOI assigned.
- [37] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, “Towards the systematic reporting of the energy and carbon footprints of machine learning,” *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020, peer-reviewed official journal article; no publisher DOI assigned.
- [38] C. van der Lee, A. Gatt, E. van Miltenburg, S. Wubben, and E. Krahmer, “Best practices for the human evaluation of automatically generated text,” in *Proceedings of the 12th International Conference on Natural Language Generation*. Association for Computational Linguistics, 2019, pp. 355–368.
- [39] M. Karpinska, N. Akoury, and M. Iyyer, “The perils of using Mechanical Turk to evaluate open-ended text generation,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 1265–1285.
- [40] L. Zheng, W.-L. Chiang, Y. Sheng *et al.*, “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena,” in *Advances in Neural Information Processing Systems*, vol. 36. Neural Information Processing Systems Foundation, 2023, pp. 46 595–46 623.
- [41] G. H. Chen, S. Chen, Z. Liu, F. Jiang, and B. Wang, “Humans or LLMs as the judge? a study on judgement bias,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024, pp. 8301–8327.
- [42] D. Lakens, A. M. Scheel, and P. M. Isager, “Equivalence testing for psychological research: A tutorial,” *Advances in Methods and Practices in Psychological Science*, vol. 1, no. 2, pp. 259–269, 2018.
- [43] D. J. Barr, R. Levy, C. Scheepers, and H. J. Tily, “Random effects structure for confirmatory hypothesis testing: Keep it maximal,” *Journal of Memory and Language*, vol. 68, no. 3, pp. 255–278, 2013.
- [44] E. Feng, W. Zhou, Z. Liu *et al.*, “Get experience from practice: LLM agents with record & replay,” 2025, preprint; no archival venue or publisher DOI verified as of 2026-09-03.
- [45] A. Madaan, N. Tandon, P. Gupta *et al.*, “Self-Refine: Iterative refinement with self-feedback,” in *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023, pp. 46 534–46 594.
- [46] M. O. Riedl, C. J. Saretto, and R. M. Young, “Managing interaction between users and agents in a multi-agent storytelling environment,” in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003, pp. 741–748.
- [47] R. E. Fikes and N. J. Nilsson, “STRIPS: A new approach to the application of theorem proving to problem solving,” *Artificial Intelligence*, vol. 2, no. 3–4, pp. 189–208, 1971.
- [48] F. B. Schneider, “Enforceable security policies,” *ACM Transactions on Information and System Security*, vol. 3, no. 1, pp. 30–50, 2000.
- [49] T. Haerder and A. Reuter, “Principles of transaction-oriented database recovery,” *ACM Computing Surveys*, vol. 15, no. 4, pp. 287–317, 1983.
- [50] R. Evans and E. Short, “Versu—a simulationist storytelling system,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 6, no. 2, pp. 113–130, 2014, predecessor title of IEEE Transactions on Games.
- [51] S. G. Ware and C. Siler, “Sabre: A narrative planner supporting intention and deep theory of mind,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, vol. 17, no. 1, 2021, pp. 99–106.
- [52] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, “Reflexion: Language agents with verbal reinforcement learning,” in *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023, pp. 8634–8652.
- [53] X. Chen, M. Lin, N. Schärli, and D. Zhou, “Teaching large language models to self-debug,” in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024, accepted poster; the OpenReview forum is the archival record (no publisher DOI), verified 2026-09-03; preprint arXiv:2304.05128.
- [54] A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, and V. Saraswat, “Combinatorial sketching for finite programs,” in *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. ACM, 2006, pp. 404–415.
- [55] Y. Huang, G. Chen, J. Yao *et al.*, “Beyond state consistency: Behavior consistency in text-based world models,” 2026, preprint; no archival venue or publisher DOI verified as of 2026-09-03.