

TempoBench-Grounded Human/AI Benchmark Game Family (GF-01):

Standalone Formal Specification with Decision and Hypothesis Glossary

Bobby Veihman
Project Specification

Last edited: February 21, 2026

Abstract

This document gives a standalone formal specification for the current TempoBench-grounded benchmark family GF-01 (see Definition 8). Ground truth is defined over a canonical substrate: (*HOA/reactive-system object, finite trace, explicit intervention semantics*). Primary scoring is machine-checkable through structured certificates rather than free-form explanation. To keep the document readable without access to repository-internal notes, all referenced decision labels (DEC-*), hypothesis labels (HYP-*), and renderer labels (GF-01-R*) are explicitly defined here.

Public implementation repository. A publishable mirror of the benchmark implementation artifacts is available at <https://github.com/official-Auralin/Benchmark-game>.

1 Reading Guide and Label Glossary

1.1 What the labels mean

- **DEC-***: locked project decisions that are normative in this specification.
- **HYP-***: explicit hypotheses (not yet treated as facts), each with a validation plan.
- **Q-***: question-lineage items (open, deferred, or resolved), with status maintained in the decision/open-question logs.
- **EVAL-***: evaluation-condition tracks (closed-book, tool-augmented, oracle-ceiling).
- **MET-***: metric/scoring regimes.
- **Code/field identifiers**: shown in monospace (e.g., `family_id`, `eval_track`).

1.2 Renderer labels used in this document (see Definition 8)

- **GF-01-R1**: side-scroller renderer where each segment/column corresponds to one timestep; interventions for timestep t are chosen before advancing to $t + 1$.
- **GF-01-R2**: puzzle-style renderer that presents the same canonical observation information in puzzle-board form.

- **GF-01-R3:** turn-locked grid/tower renderer under consideration; treated as a renderer variant only if semantic parity checks pass.

1.3 Normative decisions referenced in this spec

DEC-001

report EVAL-CB, EVAL-TA, and EVAL-OC separately; do not mix tracks.

DEC-001a

keep naming separation: EVAL-* for conditions, MET-* for metric regimes.

DEC-002

interventions occur online during play.

DEC-003

intervention semantics are ternary per (t, AP) : set 0, set 1, unchanged.

DEC-004

primary efficiency cost is number of distinct intervened timesteps; atom-count is secondary.

DEC-005

action availability must not leak hidden state.

DEC-012

normative minimality is singleton-removal.

DEC-012a

subset-minimality is optional and non-scoring.

DEC-012b

subset-minimality diagnostics must be exact-only when run.

DEC-014

hard mode uses exact-time objective; normal mode uses windowed objective.

DEC-014a

normal-mode window scales with complexity.

DEC-014b

current normal-window coefficients are fixed defaults, pending post-pilot calibration.

DEC-014c

coefficient recalibration follows a pre-registered dual-trigger rule (threshold anomaly or fixed-sample checkpoint).

DEC-014d

current provisional recalibration constants are locked as: sample target $N = 240$, discrimination threshold $\Delta_{Q1,Q4} = 0.12$, hard-quartile floor $M_{Q4} = 0.10$, shortcut-goal threshold 0.40, and shortcut-certified floor 0.05.

DEC-016

renderer variants are allowed only under semantic parity.

DEC-018
per-timestep intervention cardinality is unrestricted, constrained by global budgets.

DEC-019
exact normative checks are always mandatory.

DEC-020
optional-diagnostic infeasibility uses hybrid caps (P_{ref} , A_{ref}).

DEC-021
initial cap freeze timing is tied to the scheduled pilot checkpoint.

DEC-022
cap derivation uses a dual-threshold rule.

DEC-023
current provisional dual-threshold defaults are $p = 0.90$, $k = 3$.

DEC-024
generator determinism is mandatory for fixed input tuple.

DEC-025
AP/TS PRF under non-unique minima uses deterministic best-match mapping.

DEC-031
run artifact schemas are versioned and metadata-rich.

DEC-038
implementation priority is GF-01-R1 first.

DEC-038a
GF-01-R3 promotion requires pre-registered parity/leakage/reporting gates.

DEC-045
GF-01-R1 uses visible-goal + single scored commit run; exploration episodes are not part of official scoring protocol.

DEC-046
run artifacts must include `play_protocol/scored_commit_episode` with fixed `commit-only/true` values for official runs.

DEC-053
`tool_allowlist_id` is track-constrained under `gf01.tool_policy.v1`: EVAL-CB requires `none/empty` hash, EVAL-TA requires `local-planner-v1+non-empty` hash, and EVAL-OC requires `oracle-exact-search-v1+non-empty` hash.

DEC-054
publication split governance is locked by `gf01.split_policy.v1` with official labels `public-dev/public_val/private_eval`, target ratio `0.20/0.20/0.60`, and machine-checkable policy validation via `split-policy-check`.

DEC-066
reproducibility bundling is enforced via `release-package`: package creation requires strict run+manifest validation and emits hash-addressed artifact manifests plus rerun instructions.

DEC-067

release rotation/contamination governance is locked by `gf01.rotation_policy.v1` and checked with `release-governance-check`: prior private instances are forbidden in current public splits, and public novelty versus prior public set must exceed a configured minimum ratio.

DEC-055

adaptation/fine-tuning reporting is locked by `gf01.adaptation_policy.v1`: condition labels are `no_adaptation/prompt_adaptation/weight_finetune`, and budget/scope/protocol fields must satisfy condition-specific constraints.

DEC-056

anti-shortcut baseline panel policy is locked by `gf01.baseline_panel_policy.v1`: official publication campaigns use full panel `{random,greedy,search,tool,oracle}`; optional core smoke policy requires `{random,greedy,oracle}`.

DEC-068

release report policy is machine-checkable via `release-report-check`: required baseline-agent and required eval-track coverage must be present for each manifest (`split_id, mode`) slice.

DEC-072

integrated release-candidate policy composition is machine-checkable via `release-candidate-check`, which emits explicit stage-level outcomes for governance/report/package checks.

DEC-074

CI must execute integrated release-candidate checks and branch protection must require both status checks: `GF01 Gate / gate` and `GF01 Gate / release-candidate`.

DEC-082

CI must publish explicit commit-status contexts for those required checks on the PR head commit (fallback current workflow SHA), so hosted ruleset enforcement binds to reported contexts, not only check-run names.

DEC-075

remote branch-protection activation uses an explicit checklist handoff artifact when local execution cannot apply hosted policies directly.

DEC-057

renderer freedom is locked by `gf01.renderer_policy.v1`: `renderer_track=json` maps to `canonical-json-v1`; `renderer_track=visual` maps to `GF-01-R1`; other profiles are disallowed until explicit amendment.

DEC-058

complexity-knob governance is locked by `gf01.complexity_policy.v1`: normative calibration uses fixed equal-weight normalized composite score, while `pilot-analyze` must emit per-knob diagnostics (including constant-knob detection and per-agent slices) as non-normative audit fields.

DEC-059

partial-observability identifiability governance is locked by `gf01.identifiability_policy.v1`: use metric `single_atom_observation_divergence.v1` with thresholds `min_response_ratio=0.60` and `min_unique_signatures=8`, enforced in generation and machine-checkable `identifiability-check` validation.

DEC-060

first-cycle review of DEC-014d constants: after checkpoint-scale evidence (N=240 normal-mode unique instances), constants remain unchanged in this cycle; calibration action should target coefficient/hardness tuning under the same trigger policy.

2 Objective and Evidence Anchors

2.1 Objective

The benchmark is designed to evaluate temporal causality reasoning, aligned with TempoBench TCE-style tasks where the model must recover causally relevant interventions over time [4].

2.2 Evidence policy

Nontrivial factual claims in this document are backed by cited sources. Project-policy locks are identified by DEC labels. Statements not yet established empirically are marked as HYP labels.

3 Canonical Substrate and Formal Notation

3.1 Finite-trace benchmark scope

TempoBench defines task objects over automata and finite traces for temporal reasoning and temporal causality [4]. Coenen et al. formalize temporal causality in reactive systems in an ω -trace setting with explicit counterfactual structure [3]. This benchmark intentionally uses a finite-trace operational scope while keeping formal causality checks machine-verifiable.

Definition 1 (System and proposition partition). *Each instance uses a system object A with atomic propositions*

$$AP = AP_{\text{in}} \uplus AP_{\text{out}},$$

where AP_{in} are intervenable input propositions and AP_{out} are observable output propositions.

Definition 2 (Base trace). *Let*

$$\tau = (\tau_0, \dots, \tau_{T-1}), \quad T \geq 1,$$

where each $\tau_t : AP_{\text{in}} \rightarrow \{0, 1\}$ is an input valuation.

Intuition. τ is the default input schedule. The player/agent edits this schedule via interventions (see Definition 3, Definition 4, and Definition 5).

3.2 Intervention semantics

Definition 3 (Intervention atom). *An intervention atom is (t, a, v) with $t \in \{0, \dots, T-1\}$, $a \in AP_{\text{in}}$, $v \in \{0, 1\}$.*

Definition 4 (Certificate). *A certificate is a finite set*

$$C \subseteq \{0, \dots, T-1\} \times AP_{\text{in}} \times \{0, 1\}.$$

Definition 5 (Apply operator). Define the intervened trace $\tau[C] = \text{Apply}(\tau, C)$ (*Apply*) by

$$\tau[C]_t(a) = \begin{cases} v, & \text{if } (t, a, v) \in C, \\ \tau_t(a), & \text{otherwise.} \end{cases}$$

Malformed certificates with conflicting assignments to the same (t, a) are rejected by structural validation.

Intuition. “Unchanged” is represented by omission: if no atom targets (t, a) , base trace value $\tau_t(a)$ remains.

3.3 Run and canonical observation object

Definition 6 (Run).

$$\rho = \text{Run}(A, \tau[C])$$

denotes the benchmark rollout under the intervened trace (*Run*). For a fixed instance bundle and evaluator version, rollout semantics are replay-deterministic ([DEC-024](#), [DEC-031](#)).

Definition 7 (Canonical observation object). At timestep t , the canonical agent-facing object is

$$O(s_t) = \{t, y_t, \text{effect_status}_t, b_T^{\text{rem}}, b_A^{\text{rem}}, C_{\leq t}, \text{mode}, t^*\},$$

with hidden internal state omitted. All renderers must be semantics-preserving maps from this canonical object.

4 Instance Definition and Game Interpretation

4.1 What GF-01 means (family-level definition)

Definition 8 (GF-01 benchmark family). *GF-01* is a forward-time, intervention-driven benchmark family in which:

1. the environment evolves over discrete timesteps $t = 0, \dots, T - 1$,
2. at each timestep, an agent (human or AI) can apply do-style edits to input propositions in AP_{in} ,
3. the episode objective is to achieve a formal effect predicate e with a certificate that passes machine-checkable causal-validity tests (sufficiency + minimality),
4. scoring is computed from structured artifacts (intervention logs and final certificate), not from natural-language explanation.

Plain-language interpretation. GF-01 is the core game family name for this benchmark. All renderer variants (e.g., side-scroller, puzzle, turn-locked grid/tower) are just different user interfaces over the same formal game: same state semantics, same intervention rules, same ground truth, and same evaluator.

Definition 9 (GF-01 instance). A *GF-01* instance is

$$I = (A, AP_{\text{in}}, AP_{\text{out}}, \tau, T, e, t^*, \text{mode}, w(I), B_T, B_A, \text{meta}).$$

Field meanings.

- e : machine-checkable effect predicate,
- t^* : target timestep for effect objective,
- mode $\in \{\text{normal}, \text{hard}\}$,
- $w(I)$: trailing window size used in normal mode,
- B_T : primary timestep intervention budget,
- B_A : optional atom-budget field (diagnostic/constraint use).

This object is the formal instance contract used throughout the evaluator (see Definition 9).

4.2 Mode-specific effect objective

$$\begin{aligned} E_{\text{hard}}(\rho, e, t^*) &:= [\rho_{t^*} \models e], \\ E_{\text{norm}}(\rho, e, t^*, w) &:= \exists t \in [\max(0, t^* - w), t^*] [\rho_t \models e]. \\ E_I(\rho) &= \begin{cases} E_{\text{norm}}(\rho, e, t^*, w(I)), & \text{mode = normal,} \\ E_{\text{hard}}(\rho, e, t^*), & \text{mode = hard.} \end{cases} \end{aligned}$$

This normal/hard split is locked by DEC-014.

4.3 Normal-mode window scaling policy

Current locked default:

$$w(I) = \text{clamp}(w_{\min}, w_{\max}, \text{round}(\alpha_0 + \alpha_T T + \alpha_C z(I))),$$

with provisional defaults (DEC-014b): $\alpha_0 = 1$, $\alpha_T = 0$, $\alpha_C = 2$, $w_{\min} = 1$, $w_{\max} = 6$, and equal-weight normalized complexity composite $z(I)$ based on TempoBench-linked structural factors [4].

Complexity policy lock (DEC-058). The composite score definition above is the normative calibration signal under `gf01.complexity_policy.v1`. Per-knob predictive statistics are reported as machine-checkable diagnostics in `pilot-analyze` but do not replace composite-score trigger semantics in DEC-014d.

5 Ground-Truth Causality Contract

Definition 10 (Sufficiency).

$$\text{Suff}_I(C) := \mathbf{1}[E_I(\text{Run}(A, \tau[C])) = 1].$$

Definition 11 (Normative minimality (singleton-removal)).

$$\text{Min1}_I(C) := \mathbf{1}\left[\forall c \in C : E_I(\text{Run}(A, \tau[C \setminus \{c\}])) = 0\right].$$

Definition 12 (Normative validity).

$$\text{Valid}_I(C) := \text{Suff}_I(C) \wedge \text{Min1}_I(C).$$

Definition 13 (Ground-truth certificate set).

$$\mathcal{M}(I) := \{C \mid \text{Valid}_I(C) = 1\}.$$

Multiple distinct valid minimal certificates are allowed.

Normative policy. [DEC-012](#) locks singleton-removal minimality as normative scorer behavior.

5.1 Optional stronger diagnostic

Optional (non-scoring) exact subset-minimality diagnostic:

Definition 14 (Optional exact subset-minimality diagnostic).

$$\text{SubsetMin}_I(C) := \mathbf{1}[\#C' \subset C \text{ with } \text{Suff}_I(C') = 1].$$

By policy ([DEC-012a](#), [DEC-012b](#)), this diagnostic is optional and exact-only when run.

6 Coenen Alignment and Scope Divergence

Coenen et al. define causality with explicit PC1/PC2/PC3 structure, contingencies, and strict-subset/property-level minimality [3]. This benchmark intentionally diverges in normative scoring by using finite intervention certificates with singleton-removal minimality. The divergence is explicit and policy-locked (centered on [DEC-012](#)), not implicit.

7 Evaluation Tracks and Metric Regimes

7.1 Evaluation-condition tracks

- EVAL-CB: closed-book (no external tools beyond model-internal reasoning/scratchpad).
- EVAL-TA: tool-augmented with predeclared allowlist and full tool logging.
- EVAL-OC: solver-oracle ceiling track, reported separately.

Track separation is mandatory ([DEC-001](#), [DEC-001a](#)). Tool metadata policy is mandatory ([DEC-053](#)):

- EVAL-CB: `tool_allowlist_id=none, tool_log_hash=""`.
- EVAL-TA: `tool_allowlist_id=local-planner-v1` and non-empty `tool_log_hash`.
- EVAL-OC: `tool_allowlist_id=oracle-exact-search-v1` and non-empty `tool_log_hash`.

Adaptation metadata policy is mandatory ([DEC-055](#)):

- `adaptation_condition=no_adaptation` requires `adaptation_budget_tokens=0, adaptation_data_scope=none, adaptation_protocol_id=none`.
- `adaptation_condition in {prompt_adaptation, weight_finetune}` requires `adaptation_budget_tokens>0, adaptation_data_scope != none`, and non-empty `adaptation_protocol_id`.

7.2 Metric regimes

The symbols in this section reuse formal objects from Definition 10, Definition 11, Definition 12, and Definition 13.

MET-C (Certified Causality).

$$\text{Score}_C(I) = \mathbf{1}[\text{Valid}_I(C_a) = 1].$$

where the implementation artifact name is `Score_C`.

MET-M (Multi-objective). For agent certificate C_a :

$$\begin{aligned} M(I) &= \mathbf{1}[\text{Valid}_I(C_a) = 1], \quad G(I) = \mathbf{1}[\text{Suff}_I(C_a) = 1], \\ \text{Eff}_t(I) &= |\text{TS}(C_a)|, \quad \text{TS}(C) = \{t \mid \exists a, v : (t, a, v) \in C\}, \\ \text{Eff}_a(I) &= |C_a|. \end{aligned}$$

Per-instance lexicographic key:

$$\kappa(I) = (M(I), G(I), -\text{Eff}_t(I), -\text{Eff}_a(I)).$$

with implementation key name `kappa`.

Intuition. The scoring priority is: valid minimal causality first, then effect success, then fewer intervention timesteps.

7.3 AP/TS precision-recall-F1 under non-unique minima

TempoBench reports AP/TS causality metrics [4]. With non-unique $\mathcal{M}(I)$ (Definition 13), deterministic best-match target is ([DEC-025](#)):

$$C^*(I, C_a) = \arg \max_{C \in \mathcal{M}(I)} F1_{AP}(C_a, C),$$

then tie-break by $F1_{TS}$, then lower Eff_t , then canonical atom hash order.

7.4 Required reporting dimensions

Official reports must stratify by:

```
(family_id, eval_track, renderer_track,
renderer_profile_id,
play_protocol, scored_commit_episode,
adaptation_condition, adaptation_budget_tokens,
adaptation_data_scope, adaptation_protocol_id,
difficulty_slice, split_id)
```

No cross-track pooling is allowed. Under [DEC-046](#), official runs currently fix:

```
play_protocol=commit_only
scored_commit_episode=true
```

but fields remain explicit to prevent silent protocol mixing if policy is amended. This follows evidence that aggregate single scores can hide major failures [6] and that contamination/governance choices can distort benchmark conclusions [1, 9, 10].

Split governance policy. Publication-governance split policy is locked by [DEC-054](#). The default official ratio target is:

```
public_dev=0.20, public_val=0.20, private_eval=0.60
```

with default absolute tolerance 0.05 checked per split by `split-policy-check`. This policy is reported separately from pilot-only freezes and is intended to reduce contamination risk while preserving reproducibility slices.

Release rotation and contamination policy. Cross-cycle release governance is additionally locked by [DEC-067](#). The machine-checkable command `release-governance-check` combines `gf01.split_policy.v1` checks with rotation safeguards:

- prior-cycle `private_eval` instances must not appear in current `public_*` splits;
- current public-set novelty versus prior public set must exceed a configurable threshold (default 0.10).

This operationalizes contamination-risk controls from [9, 10] while retaining procedural-refresh pressure motivated by [2, 5].

Baseline panel policy. Anti-shortcut baseline policy is locked by [DEC-056](#). Default campaign enforcement level is `full`, requiring `random,greedy,search,tool,oracle`. An explicit `core` level is available for fast internal smoke checks and requires `random,greedy,oracle`. This follows the anti-shortcut principle that release evaluations must include heuristic-stress and ceiling references rather than a single easy baseline. At release time, [DEC-068](#) enforces this policy with `release-report-check`, which fails if required baseline agents or required eval tracks are missing from any manifest (`split_id, mode`) slice.

Integrated release-candidate command and CI policy. By [DEC-072](#), release-governance, release-report, and release-package checks are additionally composable through one machine-checkable command, `release-candidate-check`, which outputs explicit stage-level pass/fail payloads. By [DEC-074](#), CI executes this integrated command as a required status check alongside the one-shot gate. Hosted branch protection therefore requires both:

```
GF01 Gate / gate
GF01 Gate / release-candidate
```

By [DEC-082](#), workflow jobs additionally publish these exact context names as commit statuses on the PR head SHA (fallback `github.sha`) to avoid pending-state mismatches between required-check configuration and reported status contexts. Activation guidance for remote repository settings is captured by [DEC-075](#).

8 Generator Contract and Difficulty Scaling

8.1 Generator input object

$$G_{\text{in}} = (A, \tau, AP_{\text{in}}, AP_{\text{out}}, F, \text{seed}, \text{cfg}).$$

where implementation-facing field names include `seed` and `cfg`. Generation is defined for any finite trace length $T \geq 1$. Candidates are accepted only if exact normative checking confirms $\mathcal{M}(I) \neq \emptyset$ (see Definition 13).

8.2 Determinism lock

For fixed $(A, \tau, AP_{\text{in}}, AP_{\text{out}}, F, \text{seed}, \text{cfg}, \text{generator_version})$, output must be bit-identical ([DEC-024](#)). No wall-clock or system-entropy randomness is permitted.

8.3 Connection to TempoBench generation evidence

TempoBench describes a formal generation/evaluation pipeline using specification-to-automaton synthesis and causal extraction over generated traces [4]. Repo-side benchmark-runner artifacts provide AP/TS-scored execution/logging behavior [8]. Solver-validated generation as a general methodology is also supported by SATBench-style design [11].

9 Optional-Diagnostic Infeasibility Policy

Normative exact checks are always required. Only optional diagnostics may be marked infeasible ([DEC-019](#)).

Hybrid policy ([DEC-020](#)):

- platform/runtime caps P_{ref} ,
- algorithmic-budget caps A_{ref} .

Initial cap freeze timing is aligned with [DEC-021](#) checkpoint; derivation uses dual-threshold rule from [DEC-022](#) with provisional defaults $p = 0.90$, $k = 3$ ([DEC-023](#)).

10 Partial Observability and No-Query Rule

Hidden state is allowed but not directly exposed. Agents gather information only from passive observations and intervention consequences. Explicit query actions are excluded. The agent-facing channel is the canonical observation object in Definition 7. This aligns with interactive-agent evaluation practices that require explicit environment protocols rather than static one-shot prompts [7, 12].

Identifiability acceptance policy. To prevent low-signal partially observable instances, acceptance is gated by a machine-checkable identifiability policy. By [DEC-059](#):

```
policy_version = gf01.identifiability_policy.v1
metric_id = single_atom_observation_divergence.v1
min_response_ratio = 0.60
min_unique_signatures = 8
```

These checks are enforced at generation time and are auditable on bundle artifacts using `identifiability-check`. They remain explicit policy constants (rather than implicit heuristics) so amendments must be tracked via decision-log updates.

11 Renderer Policy and Current Priority

11.1 Renderer parity requirement

Renderer variants are allowed only if they preserve canonical $O(s)$ semantics, intervention semantics, and scoring outcomes (see Definition 7 and Definition 12). By DEC-057, the active renderer-policy mapping is:

- `renderer_track=json` → `renderer_profile_id=canonical-json-v1`,
- `renderer_track=visual` → `renderer_profile_id=GF-01-R1`.

Any additional renderer profile is out of scope for current official runs until an explicit amendment records parity/leakage gate evidence.

11.2 Current lock

By DEC-038, implementation priority is:

- first: GF-01-R1 (side-scroller),
- deferred candidate: GF-01-R3 (turn-locked grid/tower), promoted only after DEC-038a gates pass: semantic parity, leakage-audit parity, and track-separated reporting.

11.3 Interaction protocol lock (GF-01-R1)

By DEC-045, the baseline human-playable protocol is:

1. a visible static goal panel from episode start (including target output proposition, target timestep, and mode semantics),
2. one scored commit episode as the normative benchmark run.

Scoring contract under DEC-045. Only the commit episode contributes to official machine-checkable metrics (Definition 12, Definition 13). Official run artifacts carry fixed `play_protocol=commit-only` and `scored_commit_episode=true` values.

Why exploration is excluded from official runs. Interactive evaluation requires explicit environment protocols and logging [7, 12]. Keeping official runs commit-only avoids confounding score outcomes with unbounded rehearsal opportunities that can distort benchmark conclusions [6].

12 Active Hypotheses and Validation Plans

- **HYP-005:** exact subset-minimal diagnostics may be costly at scale. Validation: runtime/cost curves versus instance complexity.
- **HYP-006:** forward-time defensive gameplay can preserve TCE fidelity while improving engagement. Validation: matched human/agent pilot.
- **HYP-007/HYP-008:** complexity-scaled normal windows and current trigger constants are useful but provisional. Validation: post-pilot recalibration audit.

- **HYP-011/HYP-012**: current family ranking and decision matrix are inference-first and require empirical backfill.
- **HYP-013/HYP-014**: hybrid infeasibility caps improve cross-lab comparability when frozen from pilot evidence.
- **HYP-015/HYP-016/HYP-020**: calibration/discrimination and Python-first feasibility remain monitored under protocol-scale reruns and structural hardening changes.
- **HYP-017**: turn-locked grid/tower renderer may scale better than vertical side-scroller on high-complexity slices. Validation: matched-seed parity/usability pilot versus R1.

13 Resolved Deferred Item and Current Risk Note

- **Q-033 (resolved)**: larger stratified profiling/closure rerun completed under structural setting `min_witness_atoms=2`; both replicates passed pre-registered global and stratified gates with `close_q033=true`.
- Optional-diagnostic coverage comparability depends on successful cap freeze execution and transparent run metadata.

14 Summary

This specification locks the current normative benchmark contract while keeping hypotheses and question-lineage status explicitly labeled. The document is self-contained: all key labels and renderer identifiers used in formal sections are defined in Section 1.

References

- [1] Zerui Cheng, Stella Wohngig, Ruchika Gupta, Samiul Alam, Tassallah Abdullahi, Jose Alves Ribeiro, et al. Benchmarking is broken - don't let ai be its own judge. In *NeurIPS Datasets and Benchmarks Track*, 2025. URL <https://arxiv.org/abs/2510.07575>. source_id: SRC-020.
- [2] Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2020. URL <https://arxiv.org/abs/1912.01588>. source_id: SRC-010.
- [3] Norine Coenen, Bernd Finkbeiner, Hadar Frenkel, Christopher Hahn, Niklas Metzger, and Julian Siber. Temporal causality in reactive systems. In *Automated Technology for Verification and Analysis (ATVA)*, volume 13505 of *Lecture Notes in Computer Science*, pages 208–224, 2022. doi: 10.1007/978-3-031-19992-9_13. URL https://doi.org/10.1007/978-3-031-19992-9_13. source_id: SRC-003.
- [4] Nikolaus Holzer, William Fishell, Baishakhi Ray, and Mark Santolucito. Mechanics of learned reasoning 1: Tempobench, a benchmark for interpretable deconstruction of reasoning system performance. *arXiv preprint arXiv:2510.27544*, 2025. doi: 10.48550/arXiv.2510.27544. URL <https://arxiv.org/abs/2510.27544>. source_id: SRC-001.

- [5] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, et al. Dynabench: Rethinking benchmarking in nlp. In *Proceedings of NAACL-HLT 2021*, 2021. doi: 10.18653/v1/2021.nacl-main.324. URL <https://doi.org/10.18653/v1/2021.nacl-main.324>. source_id: SRC-009.
- [6] Percy Liang, Rishi Bommasani, Tony Lee, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022. URL <https://arxiv.org/abs/2211.09110>. source_id: SRC-017.
- [7] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations (Datasets and Benchmarks Track)*, 2024. URL <https://openreview.net/forum?id=zAdUB0aCTQ>. source_id: SRC-008.
- [8] nik-hz and contributors. Tempobench github repository. <https://github.com/nik-hz/tempobench>, 2025. URL <https://github.com/nik-hz/tempobench>. source_id: SRC-002; audited_commit: 1e4d5c1c4b2931dfa1dbfc7c19444a4ea318e91.
- [9] Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, et al. Open source data contamination in gpt-4 and other llms. *arXiv preprint arXiv:2310.17589*, 2023. URL <https://arxiv.org/abs/2310.17589>. source_id: SRC-019.
- [10] Soham, Anik Santikary, and Ruoxi Jia. A survey on the impact of data contamination in the evaluation of large language models. *arXiv preprint arXiv:2406.04244*, 2024. URL <https://arxiv.org/abs/2406.04244>. source_id: SRC-018.
- [11] Anjiang Wei, Yuheng Wu, Yingjia Wan, Tarun Suresh, Huanmi Tan, Zhanke Zhou, Sanmi Koyejo, Ke Wang, and Alex Aiken. Satbench: Benchmarking llms' logical reasoning via automated puzzle generation from sat formulas. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.1716. URL <https://doi.org/10.18653/v1/2025.emnlp-main.1716>. source_id: SRC-004.
- [12] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023. URL <https://arxiv.org/abs/2307.13854>. source_id: SRC-015.