

Outcome Validity

<p>Information Acquisition</p> <p>Whole string matching or substring matching:</p> <ul style="list-style-type: none">O.a.1. Considers expressions semantically equivalent to ground truth.O.a.2. Handles redundant words used by agents. <p>Substring matching:</p> <ul style="list-style-type: none">O.b.1. Handles negation modifiers used by agents.O.b.2. Is robust against systematically listing all possible answers.O.b.3. Ground truth is sufficiently complex to prevent guessing. <p>LLM-as-a-Judge:</p> <ul style="list-style-type: none">O.c.1. Demonstrates documented or experimental evidence of the judge's accuracy, self-consistency, and agreement with human.O.c.2. Is designed to resist adversarial inputs and reward hacking.	<p>Code Generation</p> <p>Unit testing or end-to-end testing:</p> <ul style="list-style-type: none">O.d.1. Verifies test cases for correctness and quality (e.g., by human).O.d.2. Measures quality of test cases using objective metrics (e.g., code coverage, cyclomatic complexity control). <p>Fuzz testing:</p> <ul style="list-style-type: none">O.e.1. Addresses potential edge cases.O.e.2. Ensures comprehensive coverage of all relevant input variations (e.g., data types, memory layouts, value ranges).O.e.3. Generates inputs that the code under testing is sensitive to. <p>End-to-end testing:</p> <ul style="list-style-type: none">O.f.1. Exercises all relevant parts of the code being tested.O.f.2. Prevents non-deterministic ("flaky") test results.
<p>State Matching</p> <p>State matching:</p> <ul style="list-style-type: none">O.g.1. Ground truth includes all states achievable after success.O.g.2. Checks relevant and irrelevant states for the challenge.O.g.3. Ground truth is complex to prevent trivial state modifications.	<p>Multistep Reasoning</p> <p>Answer matching:</p> <ul style="list-style-type: none">O.h.1. Specifies required answer formats in challenge descriptions.O.h.2. Minimizes the possibility of success by random guessing. <p>Quality measure:</p> <ul style="list-style-type: none">O.i.1. Designs quality metrics that prevent exploitation (e.g., achieving high scores by reward hacking).