Initial-Condition Control for Robot Policy Evaluation

Technical brief — The Light Company / Prism

1. What an eval actually measures

A policy evaluation is a statistical estimate. The quantity a lab wants is a policy's true success rate over a distribution of initial conditions. What it measures is a small sample — typically 30 human-operated trials per task — drawn from an initial-condition distribution that nobody controls or records. Fine-tuned manipulation policies are steeply position-sensitive: measured success on the same task can move from roughly 95% to 50–75% when an object shifts a few centimeters. When per-position success varies that steeply, the reported number is a function of where the collector happened to place objects that day. Two collectors, or the same collector on two days, produce different numbers for the same policy. The estimate is not measuring the model; it is measuring the model convolved with an undocumented placement distribution.

2. Where the variance comes from

Working sessions with an operating robot data-collection lab decomposed their eval variance into two dominant axes:

- **Initial-condition variance** — object pose at trial start is uncontrolled and unlogged. This is the axis positional control addresses directly.
- **Scoring-rubric ambiguity** — evaluator judgment differs on the same outcome (observed spread of ~30% on subjective tasks such as towel folding). This is a measurement-definition problem, not a placement problem.

The lab's estimate is that controlling both axes removes on the order of 90% of observed eval variance. This brief concerns the first axis; the second requires rubric standardization, which condition logging supports but does not solve.

3. Why the current fixes fail

- **Tape and drawn marks.** Vision policies key on visible markers, so marks contaminate training data and must be removed between collection and eval. Fixing a single pose also underfits: the policy is trained and scored on one configuration, which hides exactly the pose sensitivity the eval should be exposing.
- **More trials.** Estimator variance falls as $1/\sqrt{n}$ while cost grows linearly. Episodes are human-operated at roughly one hour per task; brute-forcing precision with sample size is not economically available. Roughly 20% of episodes are already invalid or confounded by setup error before any statistics are computed.
- **Ignoring the problem.** Published success rates built on ~10 trials at one uncontrolled position do not transfer, and cross-model comparisons made under different physical conditions are not comparisons.

4. The core problem

Underneath the placement issue is a missing layer of experimental infrastructure. Software engineering solved reproducibility with random seeds, version control, and CI: any result can be tied to the exact conditions that produced it and re-run. Physical-world model evaluation has no equivalent for scene state. There is no standard way to set an initial condition, verify it was actually achieved, record it, or restore it later. Until that layer exists, two eval numbers are comparable only by accident, regressions cannot be isolated to model versus environment, and every result is unrepeatable in principle, not just in practice. That is the problem being solved: reproducible experimental conditions for physical AI — a seed and a version history for the state of a table.

5. What the system does, mechanically

- **Stage.** Target object poses are generated programmatically and projected as outlines onto the workspace. A human places objects to the projection.
- **Verify.** Cameras compare the placed scene against the target: per-object identity (open-vocabulary detection), position, and residual error. Mismatches are flagged before the trial runs, so a bad setup is caught in seconds rather than discovered as a confounded episode.
- **Log.** The verified scene state is recorded per episode: which objects, where, for which trial of which policy.
- **Replay.** Any logged condition can be re-projected and physically recreated — for regression testing a new checkpoint against an old failure, or for diagnostics such as success-rate maps over object position.

Projection leaves no physical artifact. Guidance exists only during staging, so nothing enters the training distribution — the property that makes marks unusable is absent by construction.

6. Limits and open questions

Stated plainly, because the case is only credible with them:

- **Scope of control.** Projection controls planar placement of rigid objects. Orientation beyond what an outline conveys, deformable objects, lighting, and robot-side state are at best partially controlled.
- **Placement tolerance is unquantified.** If policy performance swings on displacements smaller than the human-to-outline placement error, residual variance survives staging. The tolerance budget needs to be measured per task, and it bounds what this method can claim.
- **Half the variance story.** Scoring ambiguity is the other dominant axis and projection does not touch it. Logging enables auditing scores against recorded conditions, but standardizing judgment is process work the lab must do.
- **Optical interference.** Projected light visible during captured frames is itself a potential confound; staging must be gated off during recording, or moved to a non-visible band (IR is under exploration, not proven).
- **Evidence base.** The variance decomposition and the ~20% invalid-episode figure come from one lab's operations. They are consistent with published sensitivity findings, but the market-level claim rests on one paying design partner until more labs are measured.

7. The economic argument

Eval cost scales linearly with trials, and trials are human hours. Removing setup-invalid episodes recovers ~20% of eval time directly. Controlling initial conditions reduces the replica count needed for a given confidence interval, and replayable conditions convert debugging from re-running whole eval suites into recreating single episodes. For a lab running continuous evals, the product's value is denominated in recovered operator hours and in results that remain valid after the person who ran them leaves the room.