EXACT-VERIFIER INTERVENTIONS IDENTIFY FUTURE-FAILURE RISK FOR LANGUAGE-MODEL CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Exact verifiers create a distinctive control problem for language-model systems: a controller must commit, branch, abstain, or escalate before a symbolic grader, test suite, proof kernel, or state checker reveals the terminal outcome. We formalize *control-object identification*, a matched intervention that fixes the decoder policy, verifier, branch pool, feature interface, calibration split, optimization budget, and decision interface while varying the supervision target and decision functional. The resulting estimand is deployment-policy future-failure risk, the conditional probability that a branch-local action will eventually be rejected under the continuation policy. We show that this scalar yields the Bayes rule for local commit-versus-escalate decisions with a fixed fallback contract, while branch-set utility requires explicitly conditional joint failure risks. The paper supplies a read-only frozen-feature controller interface and a complete experimental record specification for testing competing targets without confounding them with data, architecture, calibration, or cost changes. This formulation turns late exact-verifier failures into a precise target-identification problem and establishes the controls required for credible comparisons of language-model control objectives.

1 INTRODUCTION

Language-model systems increasingly act in settings with exact terminal checks: a symbolic answer is graded, a program is run against tests, a proof is type checked, or a tool trace is compared with an intended state. These checks establish an unambiguous acceptance event, but often only after a continuation has completed. A controller operating before that event must decide whether to accept a current action, request alternatives, abstain, or allocate a more costly fallback. The most consequential cases are locally fluent trajectories whose failure becomes visible only at terminal verification.

This setting supports a focused causal question. Which scalar should drive these pre-verification decisions? Terminal-outcome regression, preference ranking, process-value surrogates, and selective-decision objectives all produce scores from a verifier-labeled trace. Their outputs are nevertheless attached to different units, use different comparison laws, and serve different decision functionals. A comparison that changes these factors together cannot attribute a difference to the target being learned.

We introduce *control-object identification*: a matched exact-verifier intervention that fixes the deployment path and varies the statistical target and decision functional. The central object is policy-conditioned future-failure risk. It carries the semantics of the deployed continuation policy to the current branch-local action, makes the local escalation rule explicit, and distinguishes marginal branch risk from the conditional risks needed for a multi-branch decision. This distinction is important in low-entropy late-failure regimes, where a controller must identify risks not expressed by local token uncertainty alone.

The contribution is a formal and experimental framework for evaluating controller targets:

1. We define future-failure risk for an exact verifier and show its decision-theoretic role in local commit, abstention, and escalation.
2. We give a matched target-isolation protocol that separates the estimand from feature availability, decoder policy, branch construction, optimization, calibration, architecture, and resource topology.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

3. We specify the evidence record required to evaluate target-only comparisons, uncertainty, calibration, stratified risk coverage, delayed verification, and cost on a common denominator.

2 Control Objects and Related Work

Selective prediction learns a selection function at a coverage target (Geifman & El-Yaniv, 2017; 2019). Preference and listwise objectives learn an order over observed candidates (Cao et al., 2007; Xie et al., 2023), while process-verification work estimates progress, step value, or rollout success (Lightman et al., 2024; Setlur et al., 2025). When policy and verifier are fixed, a success Q-value and future-failure risk are strictly decreasing transforms of one another; after orienting scores toward failure, they encode the same population ranking. Control-object identification addresses the finite-sample and operational question left open by this equivalence: which labels, comparison law, and decision mapping recover a score that serves the deployed action.

Adaptive inference, repeated sampling, routing, and cascades change samples, search, model allocation, or replica topology (Feng et al., 2025; Brown et al., 2024; Snell et al., 2025; Gupta et al., 2024; Ong et al., 2025). Hidden-state probes and generative verifiers modify the information used to predict or establish correctness (Zhang et al., 2025a;b). Our framework is complementary: it fixes a declared deployment path, then asks what target a controller on that path should estimate. This makes target comparisons meaningful even when a broader system later changes architecture or compute allocation.

3 Problem Formulation

Let f be a language model, π a fixed continuation policy, and $V : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ an exact terminal verifier. At a branch point with frozen model state $\mathbf{h}_t$ and candidate action a_t, define

$$Q_V^\pi(\mathbf{h}_t, a_t) = \Pr_{\text{continuation}\sim\pi}\left[V(\text{complete}(t, a_t)) = 1 \mid \mathbf{h}_t, a_t\right], \qquad r_V^\pi(\mathbf{h}_t, a_t) = 1 - Q_V^\pi(\mathbf{h}_t, a_t). \tag{1}$$

The score r_V^π is indexed by the verifier and continuation policy; a change in either index changes the target. A strictly increasing energy transform, such as $-\log Q_V^\pi$ with its extended value at $Q_V^\pi = 0$, induces the same ordering. Calibration may map such a score to a probability scale, but a strictly increasing calibration map preserves every threshold-free ranking statistic computed on a fixed set of trajectories.

The target is local to the branch. It does not assume that sibling branches fail independently, and it does not identify a joint success probability for a set of branches. Let F_S be the event that every branch in a selected set S is rejected and let

$$R(S) = \Pr(F_S \mid \mathbf{h}_t), \qquad r(a \mid F_S) = \Pr(V(\text{complete}(t, a)) = 0 \mid F_S, \mathbf{h}_t)\,.$$

The distinction between $r_V^\pi(\mathbf{h}_t, a)$ and $r(a \mid F_S)$ is the boundary between a local controller score and a branch-set utility calculation.

Proposition 1 (Decision alignment under escalation and branching)**.** *Fix a branch-local action with risk r_i. Committing has utility u_{succ} after acceptance and u_{fail} after rejection. A fallback for item i has success probability q_i, cost c_i, and net utility*

$$e_i = q_i u_{\text{succ}} + (1 - q_i)u_{\text{fail}} - \lambda c_i.$$

If $u_{\text{succ}} > e_i > u_{\text{fail}}$, a Bayes-optimal local rule, with ties broken in favor of commitment, is

$$\text{commit } a_i \quad \Longleftrightarrow \quad r_i \le \tau_i^\star = \frac{u_{\text{succ}} - e_i}{u_{\text{succ}} - u_{\text{fail}}}.$$

For a branch set S with $R(S) > 0$, the marginal utility of adding a is

$$(u_{\text{succ}} - u_{\text{fail}})R(S)\,[1 - r(a \mid F_S)] - \lambda c(a).$$

Proof. The expected utility of committing is

$$(1 - r_i)u_{\text{succ}} + r_i u_{\text{fail}}.$$

Comparing it with e_i and rearranging gives the stated threshold. For the branch set, the chain rule gives

$$R(S \cup \{a\}) = R(S)r(a \mid F_S).$$

The change in the success term is therefore

$$(u_{\text{succ}} - u_{\text{fail}}) \left(R(S) - R(S \cup \{a\})\right),$$

and subtracting $\lambda c(a)$ yields the result. The conditional risk is indispensable: replacing it with marginal branch risk requires an additional dependence assumption that the proposition does not make. □

With equal fallback costs and a shared fallback contract, local escalation is ordered by r_i. Branch selection remains a joint-risk problem. This separation makes it possible to use a branch-local score for a stated local policy without overstating it as a globally optimal top-k rule.

4 Matched Target-Isolation Protocol

Control-object identification compares estimators under a pre-declared common interface. Each target-only controller receives the same frozen backbone features, decoder policy, verifier-labeled branch pool, candidate shortlist, train/calibration/test partition, optimizer budget, and decision rule. The intervention changes only the target construction and the loss used to recover it. Architecture-relaxed models, repeated-sample methods, and multi-replica systems are useful references, but they are not target-only controls because they change the information or compute path.

Controller class	Supervision unit	Statistical object	Deployment functional
Terminal regression	completed branch	verifier acceptance	commit or reject
Preference/listwise ranking	sibling branches	observed relative outcome	candidate order
Process surrogate	prefix or step	progress or proxy value	search or reward
Selective decision	request or shortlist	coverage-conditioned loss	abstain or accept
Future-failure controller	branch-local action	$r_V^{\pi}(\mathbf{h}_t, a_t)$	local commit or escalate

Target crosswalk for an exact-verifier matched intervention. The table defines comparison classes; it reports no empirical outcome.

A read-only controller can consume frozen state-derived features and produce a scalar energy $E_\theta(\mathbf{h}_t, a_t)$. A held-out increasing calibration map may place that energy on a risk scale for a thresholded decision. The implementation report must state any projection, trainable module, feature construction, and calibration fit; these are part of the model definition, not incidental details. If an adapter changes weights or trajectories, that intervention must be named separately from calibration. In particular, a threshold change can alter actions and utility at a chosen operating point, but cannot alter a full-coverage AURC on an unchanged ordered trajectory set.

5 Evidence Record for Exact-Verifier Evaluation

An empirical target comparison is credible only when its evidence record identifies the controlled objects. For every corpus, the record should give the release or version, immutable instance identifiers, inclusion rule, verifier and execution environment, and a disjoint train/calibration/test map. For every learned controller, it should give the source checkpoint, frozen and trainable modules, optimizer and schedule, branch-construction policy, seed pairing, and the precise decision interface. These fields ensure that a difference can be attributed to the intended intervention rather than to changed data, model capacity, or deployment policy.

Metrics require equally explicit units. A report of AUROC or AURC identifies the ordered prediction set, label definition, aggregation rule, and whether the metric is threshold-free. A report of a thresholded action identifies the threshold-selection split, the selected coverage or utility target,

and the emitted versus executed actions. Uncertainty requires unrounded per-unit outcomes, the resampling unit, matched pairing, the interval construction, and the inferential estimand. Rounded table cells are not a substitute for these observations.

Stratified analyses need a declared population before they can be interpreted. A risk-coverage curve must state whether successful trajectories are retained in each stratum, its denominator, the coverage direction, and the integral convention. Under the usual loss-as-failure convention, a stratum containing only failures has unit risk at every nonzero coverage and cannot have an AURC below one. Delayed-feedback studies additionally require a common event log for proposals, executed verifier calls, stopping rules, resolution denominator, runtime denominator, and serial or parallel topology. This common record makes quality–cost comparisons auditable rather than decorative.

6 Discussion

Exact verifiers provide more than a final score: they make a deployment-policy control target definable at the action that precedes eventual acceptance or rejection. The matched-intervention view focuses attention on the variables that must remain fixed when asking whether one target is better aligned with a controller's decision. It also makes calibration and adaptation claims precise. Calibration changes the score scale used by a local threshold; a changed representation, head, or trajectory is a different intervention and must be evaluated as such.

The framework applies across symbolic, executable, kernel, and state-based verifiers because its required objects are a fixed continuation policy, a branch-local action interface, and a terminal acceptance event. Its practical consequence is a disciplined comparison: distinguish the local risk needed for a commit or escalation rule from the conditional joint risks required for multi-branch utility, and preserve the data and cost record needed to test either claim.

7 Conclusion

Control-object identification casts exact-verifier language-model control as a target-selection problem with explicit deployment semantics. Policy-conditioned future-failure risk supplies the Bayes rule for local commit and escalation decisions, while the joint-risk analysis identifies the additional information required for branching. By fixing the deployment path and documenting the evidence record, the protocol enables technically meaningful comparisons among terminal, preference, process, selective, and future-failure objectives. This provides a rigorous foundation for building controllers that act before exact verification resolves.

AI Use Statement

Generative AI was used only for manuscript writing, editing, and presentation. The authors reviewed the edits and remain responsible for the final content.

References

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. *arXiv preprint arXiv:2407.21787*, 2024. doi: 10.48550/arXiv.2407.21787. URL https://arxiv.org/abs/2407.21787.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 129–136, 2007. doi: 10.1145/1273496.1273513. URL https://doi.org/10.1145/1273496.1273513.

Shengyu Feng, Xiang Kong, Shuang Ma, Aonan Zhang, Dong Yin, Chong Wang, Ruoming Pang, and Yiming Yang. Step-by-Step Reasoning for Math Problems via Twisted Sequential Monte Carlo. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2410.01920. URL https://arxiv.org/abs/2410.01920.

Yonatan Geifman and Ran El-Yaniv. Selective Classification for Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2151–2159. PMLR, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language Model Cascades: Token-Level Uncertainty And Beyond. In *International Conference on Learning Representations*, pp. 4147–4180, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/11f5520daf9132775e8604e89f53925a-Abstract-Conference.html.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs from Preference Data. In *International Conference on Learning Representations*, pp. 34433–34448, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding Progress: Scaling Automated Process Verifiers for LLM Reasoning. In *International Conference on Learning Representations*, pp. 60808–60838, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/98711dea460bdefe0e651ca23ec98ba2-Abstract-Conference.html.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *International Conference on Learning Representations*, pp. 10131–10165, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. Self-Evaluation Guided Beam Search for Reasoning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 41618–41650, 2023. doi: 10.52202/075280-1802. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81fde95c4dc79188a69ce5b24d63010b-Abstract-Conference.html.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning Models Know When They're Right: Probing Hidden States for Self-Verification, 2025a. URL https://arxiv.org/abs/2504.05419.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Seyed Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative Verifiers: Reward Modeling as Next-Token Prediction. In *International Conference on Learning Representations*, pp. 12476–12505, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/214308a2d5e3f83ef9ad2739e1cbc46d-Abstract-Conference.html.