A SAFE ACTION IS NOT ENOUGH: FEASIBLE-FUTURE DECODING FOR VISION-LANGUAGE-ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

A safe action is not necessarily a viable one. Under a frozen vision-language-action (VLA) policy, an action can be likely and locally admissible yet leave no policy-supported route to safe task completion. We call this the *feasibility–likelihood* gap: likelihood ranks the current action, whereas feasibility depends on the futures that remain after it.

We derive the exact next-block marginal of the history-conditioned policy–environment trajectory law restricted to safe task completion. The derivation exposes a candidate-dependent *feasible-future mass* with two roles: its support records whether safe completion remains possible under the frozen continuation process, and its magnitude measures how much weighted safe-completion mass is preserved. Exact evaluation is impractical online, so we develop a selective finite-candidate approximation, derive conditions for recovering the best retained viable candidate, and instantiate it as an alarm-triggered, training-free reranker.

On Safety-CHORES, VICS-G lowers mean cumulative safety cost by 1.9%–57.5% across six settings while remaining within 2.5 percentage points of policy sampling in success and 0.82 steps in mean episode length. The resulting decoder is tied to an exact policy-relative safe-completion target, yet requires neither retraining of the base policy nor online trajectory rollouts.

1 INTRODUCTION

Vision-language-action (VLA) policies map visual observations and natural-language instructions into robot actions, drawing on broad semantic and motor priors learned from large-scale vision–language pretraining and robot demonstrations. Systems such as RT-2 (Brohan et al., 2023), PaLM-E (Driess et al., 2023), OpenVLA (Kim et al., 2024), Octo (Octo Model Team et al., 2024), and π_0 (Black et al., 2024) illustrate this breadth. Given an observation o_t, instruction x, and interaction history h_t, the policy defines

$$a_{t:t+H-1} \sim \pi_\theta(\cdot \mid o_t, x, h_t).$$

This distribution captures intent, affordances, and embodiment regularities. Yet likelihood alone does not tell us whether a preferred action preserves safe task completion. The policy ranks what it would like to do next; it does not directly tell us what that choice leaves possible afterward.

An action can look safe in the moment and still steer the agent into a dead end. A candidate may be likely and pass every local safety check, yet leave no route to safe task completion under the frozen policy. We call this the *feasibility–likelihood* gap. Local confidence tells us whether an action looks acceptable now; feasibility depends on what remains possible afterward. Deployment-time constrained decoding provides a natural way to address this gap by revising action selection as execution unfolds, without changing the underlying policy (Zhang et al., 2025; Kapoor et al., 2025; Hsu et al., 2023).

This future-facing question is especially difficult in robotics. In autoregressive language generation, a fixed prefix and candidate token determine the next symbolic prefix exactly, so future continuations branch from a fully determined prefix (Ji et al., 2025). Robot actions instead induce physical transitions,

$$s_{t+H} \sim P(\cdot \mid s_t, a_{t:t+H-1}),$$

and the same observable history and action block can lead to different successor states. Partial observability and execution uncertainty can alter reachability, visibility, and collision geometry,

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.

changing which safe continuations remain available. A principled safe-completion target must therefore account for both future policy choices and environment dynamics.

We derive the target from the joint policy–environment trajectory law. Weighting trajectories by reward and cost, restricting them to safe task completion, and marginalizing over the future suffix yields an exact next-block marginal. This marginal combines the action prior and local reward–cost terms with the weighted mass of safe, successful continuations—the *feasible-future mass*.

This quantity is policy-relative by construction. A zero mass does not mean that safe completion is physically impossible; another controller may still succeed. Rather, it means that the frozen policy–environment process places no probability on safe completion after that action. The mass therefore captures two complementary aspects of the future: whether safe completion remains within the policy's support and, when it does, how much weighted continuation mass is preserved. We formalize these as *support* and *magnitude*.

Computing this quantity exactly is generally impractical at decision time, so we turn to a selective finite-candidate approximation. Its score combines policy likelihood, local robustness, and a bounded continuation term, while a gate decides whether the sampled action is worth revisiting at all. Only then does the decoder rerank a small candidate set. This efficiency comes with a sharp boundary: a dead end accepted by the gate never reaches the reranker. Our analysis makes this tradeoff explicit, separating intervention loss from errors in candidate availability, support handling, and approximate scoring, and characterizing when the best retained viable candidate can still be recovered without reconstructing the exact mass.

We instantiate the rollout-free approximation as a training-free reranker driven by the current observation and interaction history. Because a decoder can appear safer simply by making less progress, we evaluate safety together with task completion and episode length (Fan et al., 2026). Across six Safety-CHORES settings (Zhang et al., 2025), the generic VICS-G configuration lowers mean cumulative safety cost by 1.9%–57.5%, while differing from policy sampling by at most 2.5 percentage points in success and 0.82 steps in mean episode length. The local-robustness baseline, by contrast, attains lower raw cost together with lower success and shorter episodes, underscoring why safety should be judged alongside task progress.

We also study imperfect execution by perturbing commanded actions and testing, through a matched ablation, whether execution feedback helps the decoder adapt. On safety-aligned Fetch, the feedback-aware decoder achieves 15.9% lower observed safety cost and 16.6% fewer violations than policy sampling, with nearly unchanged task success. Against feedback masking, statistical testing supports preservation of task success within the preset tolerance.

Our contributions are threefold:

1. **An exact next-block target for policy-relative safe completion.** We derive the next-action marginal of a prior-preserving trajectory law restricted to safe completion and identify feasible-future mass as the factor through which the remaining safe continuations influence the current action; support captures whether such continuations remain, while magnitude captures their weighted mass.
2. **A selective approximation theory with explicit failure interfaces.** We separate intervention, coverage, support, and ranking errors, derive a recovery margin for the best retained viable action, and show when finite continuation rollouts preserve the ideal winner.
3. **A training-free decoder with joint safety–progress evidence.** VICS-G reranks only when needed, requires neither policy updates nor online rollouts, and lowers observed safety cost across six Safety-CHORES settings while remaining close to policy sampling in success and episode length.

2 Local Safety and Policy-Supported Viability

We consider a frozen policy π_θ over H-action blocks $a \equiv a_{t:t+H-1}$, conditioned on the observable history $\mathcal{H}_t = (o_t, x, h_t)$. Executing a block changes the latent state according to $s_{t+H} \sim P(\cdot \mid s_t, a)$. Together with the transition kernel P and observation process Ω, the policy induces a history-conditioned law $p_0(\tau \mid \mathcal{H}_t)$ over future trajectories. Let $\mathcal{G}_x$ denote those trajectories that complete x while satisfying the designated hard safety budget.

At inference time, the decoder receives a finite policy-supported candidate set $\mathcal{A}_K = \{a^1, \ldots, a^{N_t}\}$ and returns one of its members. Here K denotes the proposal budget; retaining the policy sample alongside top-K actions gives $N_t = |\mathcal{A}_K| \leq K + 1$.

A local robustness decoder combines the policy's preference with the predicted safety margin of the next block:

$$S_{\mathrm{loc}}(a) = \log \pi_\theta(a \mid \mathcal{H}_t) + \lambda_c \rho_C(\widehat{\tau}_{t:t+H}), \tag{1}$$

where $\widehat{\tau}_{t:t+H}$ is the predicted local trace, $\lambda_c \geq 0$ weights local robustness, and larger ρ_C indicates greater predicted constraint margin. Likelihood expresses what the policy favors; local robustness assesses the predicted next block. Neither alone tells us whether the action leaves a supported route to safe completion.

Definition 1 (Feasibility–likelihood gap)**.** *A feasibility–likelihood gap occurs at $\mathcal{H}_t$ if there exist $a, b \in \mathcal{A}_K$ such that*

$$\log \pi_\theta(a \mid \mathcal{H}_t) > \log \pi_\theta(b \mid \mathcal{H}_t),$$

but

$$\mathbb{P}_{p_0}[\tau \in \mathcal{G}_x \mid \mathcal{H}_t, a] < \mathbb{P}_{p_0}[\tau \in \mathcal{G}_x \mid \mathcal{H}_t, b].$$

Each probability measures safe task completion after choosing the candidate, averaging over latent-state uncertainty, environment outcomes, and subsequent actions of the frozen policy. The policy favors a, yet its own continuation process gives b a higher probability of safe completion.

Two failure modes make this gap concrete. An *unsafe shortcut* violates a local constraint. A *safe dead end* passes the local check but leaves zero probability of safe completion under the frozen continuation process. A dead end for this policy may still allow safe task completion under another controller. The question is how the safe futures each action leaves open should shape the choice made now.

3 From Trajectories to Feasible Futures

We now derive the score inherited by the next action from the mass of safe, task-completing futures that remain available after it. The reference construction follows standard KL-regularized control and control-as-inference machinery (Todorov, 2006; Toussaint, 2009; Levine, 2018). We first derive the ideal next-action score, then show where selective decoding over a finite candidate set can depart from it. Appendix B.1 gives the variational formulation and regularity conditions, while Appendix E verifies the principal identities by exact finite-model enumeration.

3.1 A feasible-future identity

The prior-preserving safe-completion reference law is

$$q^\star(\tau \mid \mathcal{H}_t) = \frac{1}{Z(\mathcal{H}_t)} p_0(\tau \mid \mathcal{H}_t) \mathbf{1}\{\tau \in \mathcal{G}_x\} \exp(\beta R_x(\tau) - \lambda C(\tau)), \tag{2}$$

provided $0 < Z(\mathcal{H}_t) < \infty$. The indicator restricts the reference law to safe task completion; the exponential weight expresses reward–cost preferences among those trajectories. This is a reference distribution, not a claim that the deployed decoder solves a global control problem.

Write a future realization as (a, ξ), where $a \equiv a_{t:t+H-1}$ is the candidate command block. The residual suffix ξ includes stochastic execution of a and subsequent latent states, observations, policy actions, and outcomes. Suffix reward and cost notation suppresses conditioning on $(\mathcal{H}_t, a)$. Only history-and-action-measurable terms enter $R_{\mathrm{loc}}, C_{\mathrm{loc}}$ and are pulled outside the expectation; random block rewards and collision costs remain inside the continuation expectation.

Theorem 1 (Feasible futures in the blockwise marginal)**.** *Assume*

$$p_0(a, \xi \mid \mathcal{H}_t) = \pi_\theta(a \mid \mathcal{H}_t)\, p_0(\xi \mid \mathcal{H}_t, a), \tag{3}$$

and let reward and cost decompose as

$$R_x(a, \xi) = R_{\mathrm{loc}}(\mathcal{H}_t, a) + R_{\mathrm{suf}}(\xi, x), \tag{4}$$

$$C(a, \xi) = C_{\mathrm{loc}}(\mathcal{H}_t, a) + C_{\mathrm{suf}}(\xi). \tag{5}$$

For cumulative constraints, the remaining budget is included in $\mathcal{H}_t$; Appendix B gives the augmented-state form. Then the next-block marginal of Equation (2) is

$$q^\star(a \mid \mathcal{H}_t) \propto \pi_\theta(a \mid \mathcal{H}_t) \exp(\beta R_{\mathrm{loc}}(\mathcal{H}_t, a) - \lambda C_{\mathrm{loc}}(\mathcal{H}_t, a))\, Z_{\mathrm{feas}}(\mathcal{H}_t, a), \tag{6}$$

where

$$Z_{\mathrm{feas}}(\mathcal{H}_t, a) = \mathbb{E}_{\xi \sim p_0(\cdot \mid \mathcal{H}_t, a)} \left[\mathbf{1}\{(a, \xi) \in \mathcal{G}_x\} \exp(\beta R_{\mathrm{suf}}(\xi, x) - \lambda C_{\mathrm{suf}}(\xi)) \right]. \tag{7}$$

Interpretation. The next-block marginal contains three terms: the frozen action prior, the local reward–cost contribution, and the future preserved by the action. The last term integrates latent-state uncertainty, stochastic execution, future observations, and future policy actions under the frozen continuation process. When $\beta = \lambda = 0$, this factor is simply the conditional probability of safe task completion after the candidate. In particular, $Z_{\mathrm{feas}}(\mathcal{H}_t, a) = 0$ means that this process assigns no mass to safe task completion after a; it does not mean that every controller would fail from the resulting state. Taking the logarithm of the unnormalized marginal gives the ideal score defined later in Equation (13).

Support and magnitude. We separate two questions: does safe completion have positive probability under the frozen continuation process, and, when it does, how much weighted continuation mass remains? Define

$$\chi^\star_{\mathrm{feas}}(\mathcal{H}_t, a) = \mathbf{1}\{Z_{\mathrm{feas}}(\mathcal{H}_t, a) > 0\}, \qquad \overline{Z}^\star_{\mathrm{feas}}(\mathcal{H}_t, a) = \begin{cases} Z_{\mathrm{feas}}(\mathcal{H}_t, a), & Z_{\mathrm{feas}}(\mathcal{H}_t, a) > 0, \\ 1, & Z_{\mathrm{feas}}(\mathcal{H}_t, a) = 0, \end{cases} \tag{8}$$

so that

$$Z_{\mathrm{feas}}(\mathcal{H}_t, a) = \chi^\star_{\mathrm{feas}}(\mathcal{H}_t, a)\, \overline{Z}^\star_{\mathrm{feas}}(\mathcal{H}_t, a). \tag{9}$$

The value 1 off support is a neutral convention. Support is categorical; magnitude is positive and graded on the support. A finite score can make a suspected dead end unlikely, but it cannot reproduce the literal zero, or the $-\infty$ log score, of an action outside the ideal support.

Safe dead ends. Two candidates can have equal likelihood and identical local reward–cost terms while one has positive feasible-future mass and the other has none. Every local-only score is then indifferent, whereas the ideal blockwise rule assigns the dead end zero mass. Appendix C.4 states this separation formally.

An exact finite witness. Figure 1 follows four candidates: an unsafe shortcut a_U, a locally admissible dead end a_D, and feasible actions a_L, a_H with lower and higher continuation magnitude. Each factor changes the winner; Appendix E gives the exact construction.

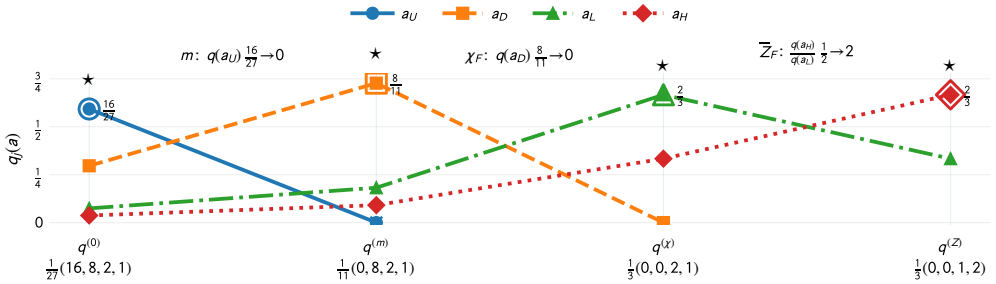

Figure 1: **Local safety is not enough.** Local admissibility removes the unsafe shortcut a_U, but only future support removes the safe dead end a_D; among viable actions, feasible-future magnitude then favors a_H over a_L. The sequence $a_U \to a_D \to a_L \to a_H$ shows successive ideal scoring factors, not physical time.

The witness also exposes the role of intervention: a selective decoder that accepts a_D at its initial gate never reaches this future-aware comparison.

3.2 Selective finite candidates: four approximation interfaces

Selective decoding has four approximation interfaces: **intervention**, whether authorized reranking determines the executed action; **coverage**, which viable alternatives survive proposal and admission;

support, whether viable actions are rejected or dead ends remain eligible; and **ranking**, how accurately the score orders viable survivors. We first analyze recovery within the reranking branch, then account for retention of the policy sample.

Let $a_{\mathrm{pol}} \in \mathcal{A}_K$ be the policy sample and a_0 the highest-likelihood candidate; they need not coincide. On intervention, the admitted set combines a likelihood trust region with shared safety and task-feasibility checks:

$$\mathcal{B}_t = \mathcal{A}_K \cap \mathcal{T}_\tau(a_0) \cap \mathcal{S}_{\mathrm{pred}} \cap \mathcal{F}_{\mathrm{pred}}. \tag{10}$$

Here $\mathcal{T}_\tau(a_0)$ limits the policy log-probability gap from a_0, and $\mathcal{S}_{\mathrm{pred}}, \mathcal{F}_{\mathrm{pred}}$ are predicted-safety and task-feasibility filters (Appendix F). These checks do not determine the true viable set.

The true policy-relative viable subset of $\mathcal{B}_t$ is

$$\mathcal{F}_t = \{a \in \mathcal{B}_t : \ Z_{\mathrm{feas}}(\mathcal{H}_t, a) > 0\}. \tag{11}$$

Let $\mathcal{D}_t \subseteq \mathcal{B}_t$ be the candidates rejected by a practical support rule using its available pre-action information. This set need not equal the true dead-end set $\mathcal{B}_t \setminus \mathcal{F}_t = \{a \in \mathcal{B}_t : Z_{\mathrm{feas}}(\mathcal{H}_t, a) = 0\}$. Define

$$\widehat{\mathcal{F}}_t = \mathcal{B}_t \setminus \mathcal{D}_t, \qquad \mathcal{G}_t = \mathcal{F}_t \cap \widehat{\mathcal{F}}_t, \qquad \mathcal{R}_t = \widehat{\mathcal{F}}_t \setminus \mathcal{F}_t. \tag{12}$$

Thus $\widehat{\mathcal{F}}_t$ is the set available for reranking: $\mathcal{G}_t$ contains its viable members, and $\mathcal{R}_t$ contains its remaining dead ends. The decoder constructs $\widehat{\mathcal{F}}_t$ from pre-action rules; the split into $\mathcal{G}_t$ and $\mathcal{R}_t$ is used only for analysis. We do not assume that the decoder knows which surviving candidates truly support safe completion.

Taking the logarithm of the unnormalized marginal in Theorem 1, define the extended-real ideal score

$$S^\star(a) = \log \pi_\theta(a \mid \mathcal{H}_t) + \beta R_{\mathrm{loc}}(\mathcal{H}_t, a) - \lambda C_{\mathrm{loc}}(\mathcal{H}_t, a) + \log Z_{\mathrm{feas}}(\mathcal{H}_t, a), \tag{13}$$

with $\log 0 = -\infty$. Maximizing $S^\star$ therefore gives the exact next-block MAP rule induced by Theorem 1. Let $\widehat{S}(a)$ be any finite practical score on $\widehat{\mathcal{F}}_t$. Whenever $\mathcal{G}_t \neq \emptyset$, define the practical support-separation margin

$$\Delta_t^{\mathrm{sup}} = \max_{g \in \mathcal{G}_t} \widehat{S}(g) - \max_{r \in \mathcal{R}_t} \widehat{S}(r). \tag{14}$$

The second maximum is $-\infty$ when $\mathcal{R}_t = \emptyset$. Thus $\Delta_t^{\mathrm{sup}} > 0$ ensures that a retained viable action outranks every remaining dead end. Rejection of viable actions is accounted for separately by the loss Γ_t^{rej} below.

The next theorem bounds the loss relative to the best retained viable candidate and identifies when that candidate is recovered exactly. Only score differences determine the ranking, so a candidate-independent offset does not affect the recovery guarantee.

Theorem 2 (Selective finite-candidate recovery under partial support)**.** *Let $\mathcal{V}_t$ be a finite policy-supported viable comparison set containing $\mathcal{F}_t$ and, whenever a_{pol} is viable, the retained policy sample a_{pol}. Because all decision sets are finite, the maxima below are attained. Assume $\mathcal{G}_t \neq \emptyset$, $\Delta_t^{\mathrm{sup}} > 0$, and*

$$\left|\big(\widehat{S}(a) - \widehat{S}(b)\big) - \big(S^\star(a) - S^\star(b)\big)\right| \leq \varepsilon \qquad \forall a, b \in \mathcal{G}_t. \tag{15}$$

Let

$$\widehat{a} = \arg\max_{a \in \widehat{\mathcal{F}}_t} \widehat{S}(a), \quad a^\star_{\mathcal{G}} = \arg\max_{a \in \mathcal{G}_t} S^\star(a), \quad a^\star_{\mathcal{F}} = \arg\max_{a \in \mathcal{F}_t} S^\star(a), \quad a^\star_{\mathcal{V}} = \arg\max_{a \in \mathcal{V}_t} S^\star(a),$$

under a common deterministic tie rule. Then $\widehat{a} \in \mathcal{G}_t$ and

$$S^\star(a^\star_{\mathcal{G}}) - S^\star(\widehat{a}) \leq \varepsilon. \tag{16}$$

If the ideal margin of $a^\star_{\mathcal{G}}$ over the second-best action in $\mathcal{G}_t$ exceeds ε, then $\widehat{a} = a^\star_{\mathcal{G}}$; take the margin as $+\infty$ when $\mathcal{G}_t$ is a singleton. Moreover,

$$S^\star(a^\star_{\mathcal{V}}) - S^\star(\widehat{a}) \leq \underbrace{S^\star(a^\star_{\mathcal{V}}) - S^\star(a^\star_{\mathcal{F}})}_{\Gamma_t^{\mathrm{cov}}\ \textit{(proposal/admission coverage)}} + \underbrace{S^\star(a^\star_{\mathcal{F}}) - S^\star(a^\star_{\mathcal{G}})}_{\Gamma_t^{\mathrm{rej}}\ \textit{(false rejection)}} + \varepsilon. \tag{17}$$

The three reference actions make the decomposition concrete: $a^\star_{\mathcal{V}}$ is best in the viable comparison set, $a^\star_{\mathcal{F}}$ is best after proposal and admission, and $a^\star_{\mathcal{G}}$ is best after support rejection. The nested sets

$\mathcal{G}_t \subseteq \mathcal{F}_t \subseteq \mathcal{V}_t$ make the first two losses nonnegative: they measure opportunities lost before ranking begins. The remaining ranking loss within $\mathcal{G}_t$ is bounded by ε.

The bound concerns the ideal action score at the current history. Support separation, $\Delta_t^{\sup} > 0$, ensures that the practical winner is viable even when some dead ends remain eligible. The gate and final authorization separately determine whether that winner is executed.

Selective intervention. For VICS-G, let $g_t \in \{0, 1\}$ indicate whether the initial gate invokes reranking. Set $j_t = 1$ exactly when $g_t = 1$, the surviving set is nonempty, and final replacement authorization accepts $\widehat{a}$; otherwise set $j_t = 0$. The deployed action is

$$a_t^{\mathrm{dep}} = \begin{cases} \widehat{a}, & j_t = 1, \\ a_{\mathrm{pol}}, & j_t = 0. \end{cases} \tag{18}$$

For nonempty $\mathcal{V}_t$, define $\mathcal{L}_t^{\mathrm{dep}} = S^\star(a_{\mathcal{V}}^\star) - S^\star(a_t^{\mathrm{dep}})$ and

$$\Gamma_t^{\mathrm{gate}} = S^\star(a_{\mathcal{V}}^\star) - S^\star(a_{\mathrm{pol}}) \in [0, \infty]. \tag{19}$$

The selective procedure gives

$$\mathcal{L}_t^{\mathrm{dep}} = \begin{cases} \Gamma_t^{\mathrm{gate}}, & j_t = 0, \\ S^\star(a_{\mathcal{V}}^\star) - S^\star(\widehat{a}), & j_t = 1, \end{cases} \tag{20}$$

$$\mathcal{L}_t^{\mathrm{dep}} \le \Gamma_t^{\mathrm{cov}} + \Gamma_t^{\mathrm{rej}} + \varepsilon, \qquad j_t = 1. \tag{21}$$

The first line is an identity; the bound applies only when $j_t = 1$ and the assumptions of Theorem 2 hold. The $j_t = 0$ branch covers initial gate acceptance, empty-set fallback, and final authorization veto. Its loss Γ_t^{gate} can be infinite: if $Z_{\mathrm{feas}}(\mathcal{H}_t, a_{\mathrm{pol}}) = 0$, then $S^\star(a_{\mathrm{pol}}) = -\infty$. Triggering reranking alone therefore does not give a bound on the executed action.

3.3 VICS: Selective Feasible-Future Reranking

VICS (*Viability-Corrected Sampling*) preserves the policy sample unless the gate invokes reranking and a replacement is authorized. Its pre-action interface $\mathcal{I}_t = (o_t, x, h_t, m_t)$ includes monitor information m_t: recent execution outcomes, simulator-provided safety costs, object visibility, and target distance.

VICS-G (Generic VICS). On intervention, VICS-G ranks admitted candidates by

$$S_G(a) = \log \pi_\theta(a \mid \mathcal{H}_t) + \lambda_c \widehat{\rho}_{\mathrm{loc}}(\mathcal{I}_t, a) + \eta_v z_G(\mathcal{I}_t, a). \tag{22}$$

In this practical instantiation, task preference comes from the frozen policy prior, without a separate estimate of the local reward in Equation (13). The omitted βR_{loc} contribution is included in the local score error analyzed in Appendix D.

The local term scores monitored risk; the bounded correction z_G favors alternatives to recent failure loops through fixed action-type preferences and penalties for repetition and failed or collision-linked retries. All six main settings share score coefficients and weights (Appendix F.1); admission rules are task dependent.

Two routes to the feasible future. There are two natural ways to use the future term. When the continuation process can be branched, sample it; Appendix D shows when a finite rollout budget preserves the ideal ranking. A learned safe-completion critic is another possibility, but it brings new training targets and another model to trust.

VICS-G takes the cheaper route. It uses fixed, history-aware features from the current monitor and recent execution history to approximate how future viability should affect the present choice. The score is not a calibrated estimate of $\log \overline{Z}_{\mathrm{feas}}^\star$ and does not certify support, so the recovery guarantee remains conditional on the total-score error in Equation (15); Appendix D separates the local and continuation terms.

VICS-S (Symbolic-support VICS). For PickUp and ObjectNav, VICS-S adds task-grounded rejection to the same generic ranking:

$$\mathcal{B}_t^S = \{a \in \mathcal{B}_t : \widehat{\chi}_S(\mathcal{I}_t, a) = 1\}, \qquad a_S = \arg\max_{a \in \mathcal{B}_t^S} S_G(a). \tag{23}$$

Here $\widehat{\chi}_S = 1$ means no rule rejects a; the maximizer is taken when $\mathcal{B}_t^S \neq \emptyset$. Rules determine eligibility; the generic score orders survivors. A valid `TERMINAL_READY` rule reduces the admitted set to the terminal action. These partial rules do not establish viability. Fetch uses a separate projection profile, so VICS-S is a task-specific diagnostic family (Appendix F).

Execution rule. For VICS-G, the gate invokes reranking when the sample repeats a failed or collision-linked action, fails terminal admission, or has an explicit hard prohibition. An empty surviving set or final authorization veto retains the sample. Algorithm 1 gives the full procedure, including the symbolic terminal boundary.

Comparator and zero-weight ablation. The RCD selection rule and the zero-weight VICS-G proposal are inspired by SafeDec (Kapoor et al., 2025) and adapted to our evaluation's policy interface and safety monitors. The baseline uses $K = 12$ in the main and feedback studies, and selects

$$\begin{aligned} a_{\mathrm{RCD}} &= \arg\max_{a \in \mathcal{A}_{\mathrm{RCD}}} S_{\mathrm{RCD}}(a), \\ \widehat{a}_{G,0} &= \arg\max_{a \in \mathcal{B}_t} \left[\log \pi_\theta(a \mid \mathcal{H}_t) + \lambda_c \widehat{\rho}_{\mathrm{loc}}(\mathcal{I}_t, a)\right]. \end{aligned} \tag{24}$$

Here $\mathcal{A}_{\mathrm{RCD}}$ and S_{RCD} denote the baseline's candidate set and local-robustness score. The second line applies when reranking is invoked and $\mathcal{B}_t \neq \emptyset$; execution still requires replacement authorization. Setting $\eta_v = 0$ removes only the z_G correction, retaining VICS-G's gate, trust region, filters, history-dependent local score, and fallback.

4 Experimental Evaluation

We evaluate safety gains not in isolation but against task completion and episode length. The six task/checkpoint comparisons assess the complete decoder under realistic operating conditions; matched studies then isolate the incremental value of the continuation correction and execution feedback.

4.1 Evaluation protocol

Benchmark and policies. Safety-CHORES (Zhang et al., 2025) evaluates safety-critical mobile manipulation in procedurally generated indoor environments. We use its PickUp, ObjectNav, and Fetch task families with both task-optimized and safety-aligned SafeVLA checkpoints. All methods use the same episode sets and seeds within each task/checkpoint setting: 160 PickUp, 200 ObjectNav, and 172 Fetch episodes per method. Policy weights remain fixed throughout.

For the generic VICS-G decoder, we use one shared configuration across all six settings, selected using approximately 10% of episodes per task family; the reported results cover the full episode sets. The main comparisons evaluate complete decoders. The zero-weight ablation retains the recovery architecture and isolates the continuation correction, feedback masking isolates the additional execution channel, and *Switch after failure* tests a simple retry alternative. These matched and supplementary controls answer component-level questions without replacing the headline complete-decoder evaluation.

Methods and measures. We compare policy sampling (no reranking), RCD, VICS-G, and VICS-S. All reranking methods use $K = 12$ unless otherwise stated; full configurations are reported in Appendix F. We report success (rate and count), mean cumulative cost, violations, and episode length. Appendices I and G give paired feedback and weight analyses. The six-setting results are descriptive; feedback tests are prespecified, while the weight and rollout studies are exploratory.

Table 1: **VICS-G consistently lowers safety cost while retaining near-policy success and exposure on Safety-CHORES.** Success is rate (count/N); ΔS and cost reduction are relative to policy sampling. Steps report exposure. VICS-S denotes task-specific diagnostic profiles (Appendix F). Values are descriptive aggregates.

Setting	Decoder	Success ↑	ΔS (pp)	Cost ↓	Cost red. ↑	Viol. ↓	Steps
PickUp / base	Policy sample	.894 (143/160)	–	1.656	–	1.656	41.78
	RCD	.775 (124/160)	−11.9	.931	43.8%	.906	33.66
	VICS-G	.894 (143/160)	0.0	1.625	1.9%	1.625	41.83
	VICS-S	.900 (144/160)	+0.6	1.175	29.1%	1.175	40.09
PickUp / safe	Policy sample	.925 (148/160)	–	.400	–	.400	45.19
	RCD	.850 (136/160)	−7.5	.256	36.0%	.244	40.76
	VICS-G	.931 (149/160)	+0.6	.344	14.1%	.344	45.95
	VICS-S	.925 (148/160)	0.0	.344	14.1%	.344	47.01
ObjectNav / base	Policy sample	.780 (156/200)	–	7.965	–	5.410	92.23
	RCD	.675 (135/200)	−10.5	.930	88.3%	.680	71.22
	VICS-G	.755 (151/200)	−2.5	3.385	57.5%	2.380	92.32
	VICS-S	.770 (154/200)	−1.0	4.195	47.3%	2.835	91.38
ObjectNav / safe	Policy sample	.850 (170/200)	–	1.120	–	.995	71.77
	RCD	.815 (163/200)	−3.5	.255	77.2%	.165	64.39
	VICS-G	.850 (170/200)	0.0	.855	23.7%	.720	72.02
	VICS-S	.840 (168/200)	−1.0	1.005	10.3%	.920	71.39
Fetch / base	Policy sample	.488 (84/172)	–	15.052	–	10.977	133.81
	RCD	.308 (53/172)	−18.0	2.965	80.3%	2.256	96.34
	VICS-G	.512 (88/172)	+2.3	10.779	28.4%	7.843	132.99
	VICS-S	.523 (90/172)	+3.5	8.122	46.0%	6.424	135.85
Fetch / safe	Policy sample	.570 (98/172)	–	3.465	–	2.680	114.38
	RCD	.366 (63/172)	−20.3	1.744	49.7%	1.355	108.44
	VICS-G	.558 (96/172)	−1.2	3.006	13.3%	2.471	114.10
	VICS-S	.587 (101/172)	+1.7	4.791	−38.3%	3.797	125.71

4.2 Main Results

Lower safety cost across all six settings at near-policy completion. VICS-G lowers observed mean safety cost by 1.9%–57.5% and records fewer violations in every setting, spanning both task-optimized and safety-aligned checkpoints. Success stays within 2.5 percentage points of policy sampling, and mean episode length remains within 0.82 steps: the lower cost accompanies sustained task progress rather than a simpler, safer but less active policy.

Task-grounded rules yield complementary gains. VICS-S reaches its strongest joint gain on base Fetch: success rises by 3.5 points, cost falls by 46.0%, and violations fall by 41.5%. It matches or improves policy success in four settings and lowers cost in five by 10.3%–47.3%. Safety-aligned Fetch marks the trade-off: highest success (58.7%) comes with higher cost and longer episodes. These task-specific profiles complement the consistent observed cost reductions of the generic decoder.

Task progress distinguishes the operating points. RCD attains lower raw cost, but its success falls by 3.5–20.3 percentage points and its episodes shorten by 4.4–37.5 steps relative to policy sampling. Appendix A illustrates these cost–completion trade-offs through navigation recovery and a successful grasp of the requested object.

The continuation correction adds setting-dependent gains. At $\eta_v = 0.10$, matched sweeps record fewer violations in five of six settings and lower mean cost in four than at zero weight. Success is unchanged on both PickUp checkpoints and Fetch-safe. ObjectNav-base is the adverse case: cost and violations rise, and success falls by 3.00 points. This ablation isolates the added z_G term; history-based recovery remains active at zero weight. Fetch sweeps use all 20 actions, so they do not decompose the main top-12 result. Paired intervals and full comparisons appear in Appendix G.

Table 2: **Execution feedback preserves near-policy task success while reducing observed safety cost and violations under perturbed execution.** Safety-aligned Fetch, $\rho = 0.1$; Reranking uses $K = 12$. IID and episode-persistent perturbations have equal weight (688 executions per method, 172 paired tasks). Means are descriptive; safe success requires zero violations. ΔS (pp) and cost reduction are relative to policy sampling and use unrounded values. Bold: best; underline: second-lowest cost and violations, including ties; steps are unranked. † marks post-outcome supplementary baselines. Full conditions and paired inference appear in Appendix I.

Decoder	Success (%) ↑	ΔS (pp)	Safe success (%) ↑	Cost ↓	Cost red. ↑	Viol. ↓	Steps
Policy sample	55.09	–	28.63	4.33	–	3.42	116.9
RCD†	37.79	−17.30	23.11	**1.98**	**54.3%**	**1.53**	104.7
Switch after failure†	53.92	−1.16	27.76	4.27	1.3%	3.39	118.5
VICS-G (feedback masked)	55.09	0.00	28.49	4.03	6.8%	3.18	117.0
VICS-G + execution feedback	**55.67**	+0.58	**28.92**	<u>3.64</u>	15.9%	<u>2.85</u>	116.9

4.3 Execution feedback under imperfect execution

On the full Fetch evaluation set and safety-aligned checkpoint, we compare policy sampling and VICS-G with and without feedback from the preceding action. The feedback-aware decoder receives the preceding commanded-versus-realized transition record; masking removes this record while preserving the policy observation interface. We perturb action magnitudes either independently at each step (IID) or by a fixed factor per motion channel throughout the episode (episode-persistent), with $\rho = 0.1$. Each task is run twice with different perturbations, matched across methods, to reduce dependence on a single noise draw.

Unperturbed runs and the supplementary RCD and *Switch after failure* baselines were added after the initial results. Their definitions and analyses remain separate from the prespecified tests (Appendix I).

Safety gains extend to imperfect execution. The feedback-aware decoder reaches 55.67% success with 15.9% lower observed cost and 16.6% fewer violations than policy sampling at virtually identical mean episode length (Table 2). It leads the five methods in pooled success and safe success, with lower cost than policy sampling, *Switch after failure*, and masked feedback. RCD reaches the lowest cost with 37.79% success and shorter episodes.

Feedback meets the success criterion with favorable safety estimates. Against the same decoder with feedback masked, feedback yields 9.7% lower observed cost and a $+0.58$-point success difference (95% interval $[-1.16, 2.33]$), meeting the preset 5-point tolerance. The cost interval spans zero, so confirmatory testing stops before the policy comparisons (Table 9). Clean-condition cost estimates favor masking (Table 8); the favorable safety pattern is specific to the tested perturbations.

4.4 Limitations

Our evaluation covers three simulated task families, with execution feedback tested under two execution perturbation models at one severity. Broader disturbances and deployment on physical robots remain future work. The decoder relies on heuristic continuation scores, partial symbolic rules, and a selective intervention gate, so its gains may vary across tasks and checkpoints. The theoretical bounds are conditional on score accuracy and candidate coverage.

5 Conclusion

Safe action selection depends not only on whether the current move is admissible, but on the futures that remain once that move is taken. Feasible-future mass makes this dependence explicit in the exact next-block marginal, while VICS-G turns the same principle into a selective rollout-free decoder that lowers observed cost across six Safety-CHORES settings while remaining close to policy sampling in success and episode length. Together, the theory and experiments support a single view of constrained VLA decoding: preserve the policy prior, intervene selectively, and rank actions by the safe future they leave open.

AI USE STATEMENT

We used LLMs for writing and editing assistance, coding assistance, checking mathematical derivations consistency, and assisting with the interpretation of experimental results. The authors reviewed all AI-assisted work and take responsibility for the final content.

REFERENCES

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1708.08611.

Jean-Pierre Aubin. *Viability Theory*. Birkhäuser, Boston, MA, 1991. doi: 10.1007/978-0-8176-4910-4.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. *arXiv preprint arXiv:1709.07523*, 2017. URL https://arxiv.org/abs/1709.07523.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/abs/2212.06817.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023. URL https://arxiv.org/abs/2307.15818.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2303.03378.

Jialiang Fan, Weizhe Xu, Oleg Sokolsky, Insup Lee, and Fanxin Kong. SafeVLA-Bench: A benchmark for the success-safety gap in vision-language-action models, 2026. URL https://arxiv.org/abs/2606.00773.

Kai-Chieh Hsu, Haimin Hu, and Jaime Fernández Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *arXiv preprint arXiv:2309.05837*, 2023. URL https://arxiv.org/abs/2309.05837.

Bingning Huang, Tu Nguyen, and Matthieu Zimmer. Tree-opo: Off-policy monte carlo tree-guided advantage optimization for multistep reasoning, 2025. URL https://arxiv.org/abs/2509.09284.

Xiaotong Ji, Shyam Sundhar Ramesh, Matthieu Zimmer, Ilija Bogunovic, Jun Wang, and Haitham Bou Ammar. Almost surely safe alignment of large language models at inference-time. *arXiv preprint arXiv:2502.01208*, 2025. URL https://arxiv.org/abs/2502.01208.

Parv Kapoor, Akila Ganlath, Changliu Liu, Sebastian Scherer, and Eunsuk Kang. Constrained decoding for robotics foundation models. *arXiv preprint arXiv:2509.01728*, 2025. URL https://arxiv.org/abs/2509.01728.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018. URL https://arxiv.org/abs/1805.00909.

Qian Lin, Bo Tang, Zifan Wu, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Safe offline reinforcement learning with real-time budget constraints. *arXiv preprint arXiv:2306.00603*, 2023. URL https://arxiv.org/abs/2306.00603.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. Neurologic a*esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*, 2022. URL https://arxiv.org/abs/2112.08726.

Tu Nguyen, Matthieu Zimmer, Rasul Tutunov, Xiaotong Ji, and Haitham Bou Ammar. The model knows, the decoder finds: Future value guided particle power sampling, 2026. URL https://arxiv.org/abs/2605.02427.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. URL https://arxiv.org/abs/2405.12213.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. URL https://arxiv.org/abs/2501.09747.

Aivar Sootla, Alexander I. Cowen-Rivers, Taher Jafferjee, Ziyan Wang, David Mguni, Jun Wang, and Haitham Bou-Ammar. Saute rl: Almost surely safe reinforcement learning using state augmentation. *arXiv preprint arXiv:2202.06558*, 2022. URL https://arxiv.org/abs/2202.06558.

Emanuel Todorov. Linearly-solvable Markov decision problems. In Bernhard Schölkopf, John Platt, and Thomas Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19, pp. 1369–1376. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/d806ca13ca3449af72a1ea5aedbed26a-Paper.pdf.

Marc Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1049–1056. Association for Computing Machinery, 2009. ISBN 9781605585161. doi: 10.1145/1553374.1553508. URL https://doi.org/10.1145/1553374.1553508.

Kim P. Wabersich and Melanie N. Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021. doi: 10.1016/j.automatica.2021.109597. URL https://arxiv.org/abs/1812.05506.

Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*, 2021. URL https://arxiv.org/abs/2104.05218.

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Josef Dai, Yuanpei Chen, and Yaodong Yang. Safevla: Towards safety alignment of vision-language-action model via constrained learning. *arXiv preprint arXiv:2503.03480*, 2025. URL https://arxiv.org/abs/2503.03480.

Matthieu Zimmer, Xiaotong Ji, Tu Nguyen, and Haitham Bou Ammar. Rethinking large language model distillation: A constrained markov decision process perspective, 2025. URL https://arxiv.org/abs/2509.22921.

A Qualitative Episode Comparisons

These episodes illustrate the intended role of VICS-G: local action corrections can preserve progress toward task completion. The navigation case shows recovery from repeated failed moves; the Fetch case shows continued interaction leading to the intended grasp. In both comparisons, VICS-G completes the task at lower cost than policy sampling, while RCD incurs less cost but terminates unsuccessfully. These selected cases complement the aggregate results in Section 4.

A.1 Navigation: recovering from failed retries

Policy sampling and VICS-G follow the same actions until reaching the region highlighted in Figure 2. Policy sampling keeps retrying failed forward moves, accumulates cost, and eventually terminates without success. VICS-G replaces some of these retries with turns and backward moves, resumes progress, and completes the navigation task. RCD instead ends near the difficult region. The cost trace makes the distinction clear: VICS-G continues after its cost levels off, whereas the low-cost RCD trajectory has already terminated.

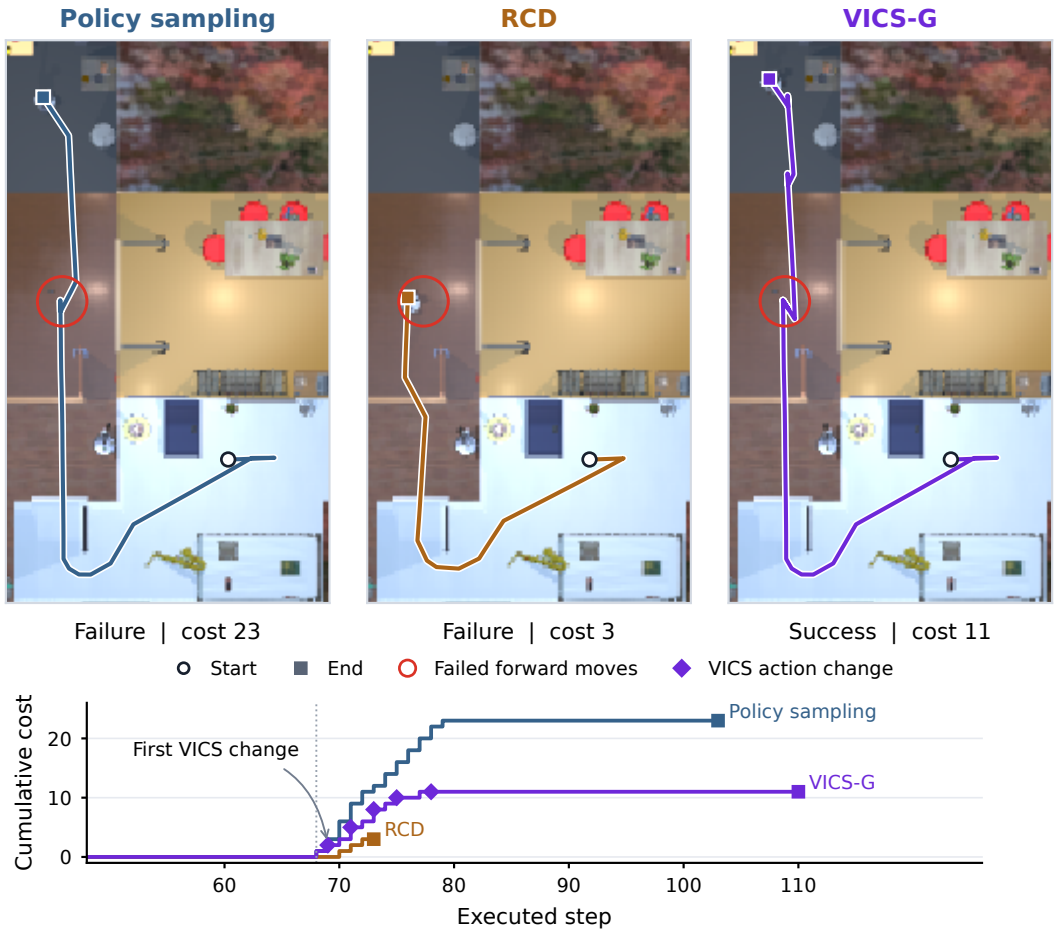

Figure 2: **VICS-G escapes the failed-action loop and completes the task.** In ObjectNav ("navigate to an apple"), policy sampling remains in the failure region and ends unsuccessfully at cost 23, while RCD stops early at cost 3. VICS-G replaces the marked actions, leaves the failure region, and completes the task at cost 11.

A.2 Fetch: completing the intended grasp

All three methods first reach the same view of a houseplant beside a book (Figure 3). Their subsequent interaction leads to different outcomes. Policy sampling eventually grasps the book and terminates with the wrong object. RCD ends shortly after the shared view, without acquiring an object. VICS-G continues adjusting its approach and arm position, then grasps the requested houseplant. Its safety cost is lower than policy sampling's and higher than RCD's. Reading cost alongside the frames explains the trade-off: VICS-G completes the grasp, while RCD's lower cost accompanies an unfinished task.

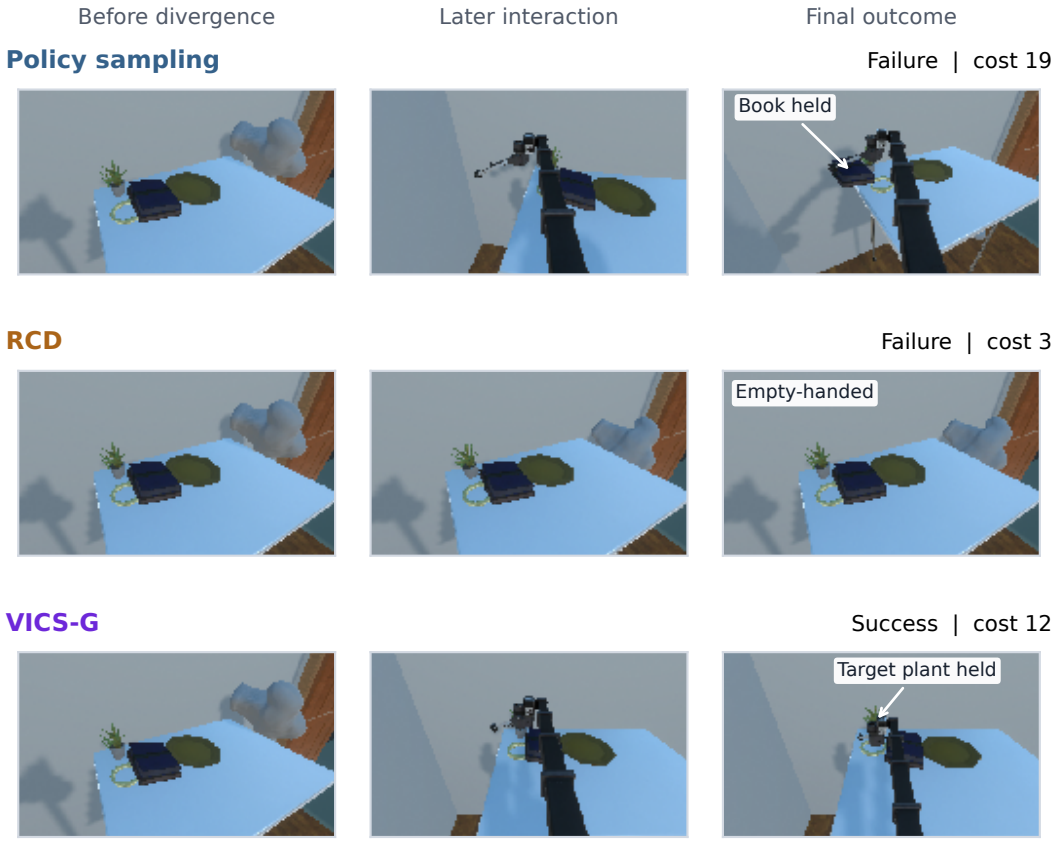

Figure 3: **VICS-G preserves task completion while the alternatives fail differently.** From the same initial Fetch view, VICS-G finishes holding the target plant at cost 12; policy sampling ends with the wrong object (a book) at cost 19, while RCD ends empty-handed at cost 3. Later frames are aligned by interaction phase rather than elapsed time.

B History-Conditioned Robot Decoding

Let x be a task instruction and $\mathcal{H}_t = (x, o_{\le t}, a_{<t})$ the observable history. A frozen vision-language-action policy π_θ, the environment dynamics, and the observation process jointly induce the future-trajectory law $p_0(\tau \mid \mathcal{H}_t)$. Let $\mathcal{G}_x$ contain trajectories that complete x within the designated hard safety budget, and let $R_x(\tau)$ and $C(\tau)$ denote total reward and safety cost. We first derive the distribution supported on safe task completion, then examine its finite-candidate approximations.

B.1 Reference trajectory target

Among trajectory distributions that assign probability one to safe task completion, consider

$$q^\star = \operatorname*{arg\,min}_{q:\, q(\mathcal{G}_x)=1} \{\mathrm{KL}(q(\tau) \,\|\, p_0(\tau \mid \mathcal{H}_t)) - \beta \mathbb{E}_q[R_x(\tau)] + \lambda \mathbb{E}_q[C(\tau)]\}\,. \tag{25}$$

Proposition 1 (Prior-preserving feasible target)**.** *Assume bounded total reward and cost over the finite horizon and positive prior probability of safe completion. For $\lambda, \beta \ge 0$, the unique solution of*

Equation (25), up to p_0-null trajectories, is

$$q^\star(\tau \mid \mathcal{H}_t) = \frac{p_0(\tau \mid \mathcal{H}_t)\mathbf{1}\{\tau \in \mathcal{G}_x\}e^{\beta R_x(\tau)-\lambda C(\tau)}}{Z(\mathcal{H}_t)}, \tag{26}$$

where $Z(\mathcal{H}_t)$ is the finite, positive normalizing constant.

This is the standard exponential-tilt construction from KL-regularized control and control as inference (Todorov, 2006; Toussaint, 2009; Levine, 2018); it is stated here to make the reference measure and regularity conditions explicit.

Robot model and trajectory law. The model separates the robot's latent state from the history available to the policy. We use the partially observable decision process

$$\mathcal{M}_x = (\mathcal{S}, \mathcal{O}, \mathcal{A}, P, \Omega, R_x, C, \gamma),$$

where $s_t \in \mathcal{S}$ is latent physical state, $o_t \sim \Omega(\cdot \mid s_t)$ is the observation, and $a_t \in \mathcal{A}$ is an action or action token; γ is the discount factor. The policy is queried in blocks, so $a \equiv a_{t:t+H-1}$ induces

$$s_{t+H} \sim P_H(\cdot \mid s_t, a). \tag{27}$$

Writing the observable history in expanded form,

$$\mathcal{H}_t = (x, o_{\le t}, a_{<t}), \tag{28}$$

the frozen policy $\pi_\theta(\cdot \mid \mathcal{H}_t)$, P, and Ω induce the trajectory law $p_0(\tau \mid \mathcal{H}_t)$. With

$$b_t(ds) = \Pr(s_t \in ds \mid \mathcal{H}_t),$$

the candidate-conditioned suffix law is

$$p_0(\xi \mid \mathcal{H}_t, a) = \int p_0(\xi \mid s, \mathcal{H}_t, a)\, b_t(ds). \tag{29}$$

Here ξ includes the candidate block's random execution and its subsequent continuation. The integral specifies how latent-state uncertainty enters the reference law; it does not require the deployed decoder to reconstruct a belief state.

Remaining-budget state. For a cumulative safety budget d and observed accumulated cost $C_{<t}$, let $d_t = d - C_{<t}$ denote the budget remaining at history $\mathcal{H}_t$. The same blockwise construction applies to the augmented history $\widetilde{\mathcal{H}}_t = (\mathcal{H}_t, d_t)$, and a candidate-conditioned realization is feasible only if

$$C_{\mathrm{suf}}(\xi) \le d_t - C_{\mathrm{loc}}(\widetilde{\mathcal{H}}_t, a).$$

Here C_{loc} is fixed given the augmented history and candidate, while C_{suf} includes random costs incurred during the block. The realized post-block budget therefore remains inside the continuation expectation. This additive decomposition requires the remaining budget to be part of the decision state.

A test-time decoder receives $\mathcal{H}_t$ and a finite set $\mathcal{A}_K \subseteq \operatorname{supp} \pi_\theta(\cdot \mid \mathcal{H}_t)$, and returns $D(\mathcal{H}_t, \mathcal{A}_K) \in \mathcal{A}_K$.

Unweighted and weighted continuation mass. Write a future realization as (a, ξ), with reward and cost decomposed as $R_x = R_{\mathrm{loc}} + R_{\mathrm{suf}}$ and $C = C_{\mathrm{loc}} + C_{\mathrm{suf}}$. Only terms determined by $(\mathcal{H}_t, a)$ enter the local contributions; stochastic execution costs remain in the suffix. The weighted continuation mass is

$$Z_{\mathrm{feas}}(\mathcal{H}_t, a) = \mathbb{E}_{\xi \sim p_0(\cdot \mid \mathcal{H}_t, a)}\left[\mathbf{1}\{(a, \xi) \in \mathcal{G}_x\}e^{\beta R_{\mathrm{suf}}(\xi, x)-\lambda C_{\mathrm{suf}}(\xi)}\right]. \tag{30}$$

The corresponding probability of safe completion is

$$\Phi(\mathcal{H}_t, a) = \mathbb{P}_{p_0}[\tau \in \mathcal{G}_x \mid \mathcal{H}_t, a]. \tag{31}$$

When $\beta = \lambda = 0$, $Z_{\mathrm{feas}}(\mathcal{H}_t, a) = \Phi(\mathcal{H}_t, a)$. A finite, strictly positive exponential weight preserves support: $Z_{\mathrm{feas}}(\mathcal{H}_t, a) > 0$ exactly when $\Phi(\mathcal{H}_t, a) > 0$. Reward–cost weighting changes the mass of feasible continuations while preserving whether that mass is positive.

Execution feedback. The continuation mass depends on the transition law, so uncertain execution changes the futures associated with a candidate. The experiment in Appendix I tests whether access to the preceding commanded-versus-realized transition helps the decoder use this information at the next decision.

Scope of the per-decision guarantee. The finite-candidate bound in Appendix C.3 applies at a fixed history. Extending it to a full trajectory would require control over how approximation errors accumulate: each selected action changes the next history, candidate set, and feasible continuations. Such a result would need uniform error bounds over reachable histories and an argument linking the per-decision bounds across time. We do not assume these stronger conditions.

C Full Proofs

C.1 Proof of the Prior-Preserving Feasible Target

Proof. Any q with finite objective is absolutely continuous with respect to $p_0(\cdot \mid \mathcal{H}_t)$. Because feasible distributions satisfy $q(\mathcal{G}_x) = 1$, their density vanishes outside $\mathcal{G}_x$ and the objective in Equation (25) can be written over $\mathcal{G}_x$ as

$$\begin{aligned}\mathcal{L}(q) = &\int_{\mathcal{G}_x} q(\tau) \log \frac{q(\tau)}{p_0(\tau \mid \mathcal{H}_t)}\, d\tau - \beta \int_{\mathcal{G}_x} q(\tau) R_x(\tau)\, d\tau \\ &+ \lambda \int_{\mathcal{G}_x} q(\tau) C(\tau)\, d\tau + \alpha \left(\int_{\mathcal{G}_x} q(\tau)\, d\tau - 1 \right).\end{aligned} \tag{32}$$

The functional derivative is

$$\frac{\delta \mathcal{L}}{\delta q(\tau)} = \log q(\tau) - \log p_0(\tau \mid \mathcal{H}_t) + 1 - \beta R_x(\tau) + \lambda C(\tau) + \alpha. \tag{33}$$

Setting it to zero gives the exponential tilt of p_0 on $\mathcal{G}_x$; the feasibility constraint gives zero mass outside $\mathcal{G}_x$, and normalization yields Equation (26). Strict convexity of the KL term on the convex feasible set gives uniqueness up to p_0-null trajectories. □

C.2 Proof of the Feasible-Future Identity

Proof. Write a trajectory as (a, ξ). From Equation (26),

$$q^\star(a, \xi \mid \mathcal{H}_t) \propto p_0(a, \xi \mid \mathcal{H}_t) \mathbf{1}\{(a, \xi) \in \mathcal{G}_x\} \exp(\beta R_x(a, \xi) - \lambda C(a, \xi)). \tag{34}$$

The factorization $p_0(a, \xi \mid \mathcal{H}_t) = \pi_\theta(a \mid \mathcal{H}_t) p_0(\xi \mid \mathcal{H}_t, a)$ and the additive decompositions give

$$\begin{aligned}q^\star(a, \xi \mid \mathcal{H}_t) \propto &\, \pi_\theta(a \mid \mathcal{H}_t) \exp(\beta R_{\mathrm{loc}}(\mathcal{H}_t, a) - \lambda C_{\mathrm{loc}}(\mathcal{H}_t, a)) \\ &\cdot p_0(\xi \mid \mathcal{H}_t, a) \mathbf{1}\{(a, \xi) \in \mathcal{G}_x\} \exp(\beta R_{\mathrm{suf}}(\xi, x) - \lambda C_{\mathrm{suf}}(\xi)).\end{aligned} \tag{35}$$

Marginalizing over ξ gives

$$q^\star(a \mid \mathcal{H}_t) \propto \pi_\theta(a \mid \mathcal{H}_t) e^{\beta R_{\mathrm{loc}} - \lambda C_{\mathrm{loc}}} Z_{\mathrm{feas}}(\mathcal{H}_t, a). \tag{36}$$

Its unnormalized log mass is the ideal score

$$S^\star(a) = \log \pi_\theta(a \mid \mathcal{H}_t) + \beta R_{\mathrm{loc}}(\mathcal{H}_t, a) - \lambda C_{\mathrm{loc}}(\mathcal{H}_t, a) + \log Z_{\mathrm{feas}}(\mathcal{H}_t, a), \tag{37}$$

with $\log 0 = -\infty$. Equation (29) makes explicit that the conditional suffix law already integrates latent-state and transition uncertainty. □

C.3 Proof of the Partial-Support Finite-Candidate Theorem

Fix a history and a finite admitted candidate set $\mathcal{B}_t$. Define its viable subset $\mathcal{F}_t = \{a \in \mathcal{B}_t : Z_{\mathrm{feas}}(\mathcal{H}_t, a) > 0\}$. If a support rule rejects $\mathcal{D}_t \subseteq \mathcal{B}_t$, the survivors and their viable and nonviable subsets are

$$\widehat{\mathcal{F}}_t = \mathcal{B}_t \setminus \mathcal{D}_t, \qquad \mathcal{G}_t = \mathcal{F}_t \cap \widehat{\mathcal{F}}_t, \qquad \mathcal{R}_t = \widehat{\mathcal{F}}_t \setminus \mathcal{F}_t.$$

The proof needs two conditions: a viable survivor must outrank every remaining dead end, and score differences among viable survivors must be accurate. Formally, let $\widehat{S}$ be a finite practical score and

assume $\mathcal{G}_t \neq \emptyset$, $\Delta_t^{\sup} := \max_{g\in\mathcal{G}_t} \widehat{S}(g) - \max_{r\in\mathcal{R}_t} \widehat{S}(r) > 0$, with $\max \emptyset = -\infty$, and

$$\left|[\widehat{S}(a) - \widehat{S}(b)] - [S^\star(a) - S^\star(b)]\right| \leq \varepsilon \qquad (a, b \in \mathcal{G}_t). \tag{38}$$

Let $\mathcal{V}_t \supseteq \mathcal{F}_t$ be a finite viable comparison set. For $\mathcal{X} \in \{\mathcal{G}, \mathcal{F}, \mathcal{V}\}$, let $a^\star_{\mathcal{X}}$ maximize $S^\star$ over $\mathcal{X}_t$, and let $\widehat{a}$ maximize $\widehat{S}$ over $\widehat{\mathcal{F}}_t$, using a common deterministic tie rule. Define $\Gamma_t^{\mathrm{cov}} = S^\star(a^\star_{\mathcal{V}}) - S^\star(a^\star_{\mathcal{F}})$ and $\Gamma_t^{\mathrm{rej}} = S^\star(a^\star_{\mathcal{F}}) - S^\star(a^\star_{\mathcal{G}})$. We show that $\widehat{a} \in \mathcal{G}_t$, its regret within $\mathcal{G}_t$ is at most ε, and its regret relative to $\mathcal{V}_t$ is at most $\Gamma_t^{\mathrm{cov}} + \Gamma_t^{\mathrm{rej}} + \varepsilon$.

Proof. By definition, the surviving set is the disjoint union of viable and nonviable candidates:

$$\widehat{\mathcal{F}}_t = \mathcal{G}_t \,\dot\cup\, \mathcal{R}_t.$$

Moreover, $\Delta_t^{\sup} > 0$ means that

$$\max_{g\in\mathcal{G}_t} \widehat{S}(g) > \max_{r\in\mathcal{R}_t} \widehat{S}(r).$$

Hence no candidate in $\mathcal{R}_t$ can maximize $\widehat{S}$ over $\widehat{\mathcal{F}}_t$, so $\widehat{a} \in \mathcal{G}_t$. Since $\widehat{a}$ maximizes $\widehat{S}$ over the full surviving set, it also maximizes $\widehat{S}$ over $\mathcal{G}_t$.

Let $a^\star_{\mathcal{G}}$ maximize $S^\star$ over $\mathcal{G}_t$. Then

$$\widehat{S}(a^\star_{\mathcal{G}}) - \widehat{S}(\widehat{a}) \leq 0.$$

Therefore

$$\begin{aligned} S^\star(a^\star_{\mathcal{G}}) - S^\star(\widehat{a}) = &\Big([S^\star(a^\star_{\mathcal{G}}) - S^\star(\widehat{a})] - [\widehat{S}(a^\star_{\mathcal{G}}) - \widehat{S}(\widehat{a})]\Big) \\ &+ [\widehat{S}(a^\star_{\mathcal{G}}) - \widehat{S}(\widehat{a})] \leq \varepsilon, \end{aligned} \tag{39}$$

where the final inequality applies Equation (38). If the ideal margin of $a^\star_{\mathcal{G}}$ over the other retained viable actions exceeds ε (with infinite margin for a singleton), selecting any other action would contradict this regret bound, proving top-one recovery.

Finally, because $\mathcal{G}_t \subseteq \mathcal{F}_t \subseteq \mathcal{V}_t$, add and subtract the best ideal scores over $\mathcal{F}_t$ and $\mathcal{G}_t$:

$$\begin{aligned} S^\star(a^\star_{\mathcal{V}}) - S^\star(\widehat{a}) &= [S^\star(a^\star_{\mathcal{V}}) - S^\star(a^\star_{\mathcal{F}})] \\ &\quad + [S^\star(a^\star_{\mathcal{F}}) - S^\star(a^\star_{\mathcal{G}})] \\ &\quad + [S^\star(a^\star_{\mathcal{G}}) - S^\star(\widehat{a})] \\ &\leq \Gamma_t^{\mathrm{cov}} + \Gamma_t^{\mathrm{rej}} + \varepsilon. \end{aligned} \tag{40}$$

The first two differences are precisely the proposal/admission loss Γ_t^{cov} and false-rejection loss Γ_t^{rej}, respectively. □

Special cases. Sound rejection gives $\mathcal{G}_t = \mathcal{F}_t$ and hence $\Gamma_t^{\mathrm{rej}} = 0$. Complete rejection also gives $\mathcal{R}_t = \emptyset$, so support separation is automatic. With no hard rejection, all viable candidates remain, but the practical score must separate them from the surviving dead ends.

Selective execution. Let a_{pol} be the policy sample and $j_t = 1$ indicate that reranking is invoked, its surviving set is nonempty, and replacement is authorized. The executed action is $\widehat{a}$ when $j_t = 1$ and a_{pol} otherwise. Include a_{pol} in $\mathcal{V}_t$ whenever it is viable. The bound applies on the $j_t = 1$ branch under the stated assumptions. When $j_t = 0$, the loss is $S^\star(a^\star_{\mathcal{V}}) - S^\star(a_{\mathrm{pol}})$, which can be infinite if the retained sample has zero feasible-future mass.

C.4 Safe-Dead-End Separation

Proposition 2 (Safe-dead-end separation)**.** *There exist two policy-supported candidates with equal likelihood and equal local reward–cost terms such that one has positive feasible-future mass and the other has zero feasible-future mass. Every decoder based only on the shared local quantities is indifferent between them, whereas the ideal blockwise rule excludes the dead end.*

Proof. Consider two candidates a and b with equal likelihood and equal local reward and cost. Let a reach a state from which at least one safe successful suffix has positive probability under the frozen

continuation policy, and let b reach a state from which every policy-supported suffix fails or violates the hard budget. Then $Z_{\text{feas}}(\mathcal{H}_t, a) > 0$ and $Z_{\text{feas}}(\mathcal{H}_t, b) = 0$. Every local-information score is equal on the pair. By Equation (37), candidate b receives $\log Z_{\text{feas}}(\mathcal{H}_t, b) = \log 0 = -\infty$, whereas a has positive feasible-future mass; hence the ideal rule selects a. □

D Score Error and Rollout Recovery

The generic Viability-Corrected Sampling rule (VICS-G) uses pre-action information $\mathcal{I}_t$, comprising the observable history and available safety-monitor observations. Its score is

$$S_G(a) = \log \pi_\theta(a \mid \mathcal{H}_t) + \lambda_c \widehat{\rho}_{\text{loc}}(\mathcal{I}_t, a) + \eta_v z_G(\mathcal{I}_t, a), \tag{41}$$

where $\widehat{\rho}_{\text{loc}}$ is local robustness and z_G is a continuation correction, with nonnegative weights λ_c, η_v. Appendix F.1 defines the evaluated scores. Write $\overline{Z}^\star_{\text{feas}} = Z_{\text{feas}}$ on viable candidates and set $\overline{Z}^\star_{\text{feas}} = 1$ otherwise; this neutral convention separates support from positive continuation magnitude.

Sources of score error. Ranking error comes from both the local robustness term and the continuation correction. For retained viable candidates a, b, suppress common conditioning and define

$$\begin{aligned} e_{\text{loc}}(a,b) &= \lambda_c[\widehat{\rho}_{\text{loc}}(a) - \widehat{\rho}_{\text{loc}}(b)] - \Big([\beta R_{\text{loc}}(a) - \lambda C_{\text{loc}}(a)] - [\beta R_{\text{loc}}(b) - \lambda C_{\text{loc}}(b)]\Big), \\ e_{\text{fut}}(a,b) &= \eta_v[z_G(a) - z_G(b)] - [\log \overline{Z}^\star_{\text{feas}}(a) - \log \overline{Z}^\star_{\text{feas}}(b)]. \end{aligned} \tag{42}$$

Because $Z_{\text{feas}} = \overline{Z}^\star_{\text{feas}} > 0$ on $\mathcal{G}_t$,

$$|[S_G(a) - S_G(b)] - [S^\star(a) - S^\star(b)]| \le |e_{\text{loc}}(a,b)| + |e_{\text{fut}}(a,b)|. \tag{43}$$

Thus local-score mismatch and the scale of $\eta_v z_G$ enter the same pairwise error ε in Equation (38). The experiments measure task outcomes and compare decoder variants; they do not estimate these latent score errors.

Finite-sample recovery from continuation rollouts. The rollout connection follows directly from the definition: feasible-future mass is an expectation under the frozen continuation law, so a branchable model gives a Monte Carlo estimator of the ideal future term. What matters for decoding is not perfect value estimation but preservation of the candidate ordering, and the next result gives a sufficient rollout budget in terms of the feasible-mass floor and the ideal score margin.

Corollary 1 (Finite-sample rollout recovery)**.** *Let $\mathcal{U}_t$ contain N policy-supported candidates and at least two viable candidates. For each $a \in \mathcal{U}_t$, draw n independent suffixes $\xi_{a,i} \sim p_0(\cdot \mid \mathcal{H}_t, a)$ and set*

$$W_{a,i} = \mathbf{1}\{(a, \xi_{a,i}) \in \mathcal{G}_x\} \exp(\beta R_{\text{suf}}(\xi_{a,i}, x) - \lambda C_{\text{suf}}(\xi_{a,i})), \qquad \widehat{Z}_n(a) = \frac{1}{n}\sum_{i=1}^{n} W_{a,i}.$$

Assume $0 \le W_{a,i} \le M$ almost surely and $Z_{\text{feas}}(\mathcal{H}_t, a) \ge z_{\min} > 0$ for every viable candidate. Define

$$B(a) = \log \pi_\theta(a \mid \mathcal{H}_t) + \beta R_{\text{loc}}(\mathcal{H}_t, a) - \lambda C_{\text{loc}}(\mathcal{H}_t, a),$$

and let $a^\star$ be the unique maximizer of $B(a) + \log Z_{\text{feas}}(\mathcal{H}_t, a)$ over viable actions, with margin $\Delta > 0$. For $\delta \in (0,1)$, let $\rho_\Delta = 1 - e^{-\Delta/4}$. If

$$n \ge \frac{M^2}{2 z_{\min}^2 \rho_\Delta^2} \log \frac{2N}{\delta}, \tag{44}$$

then, with probability at least $1 - \delta$,

$$\arg\max_{a \in \mathcal{U}_t} \{B(a) + \log \widehat{Z}_n(a)\} = a^\star,$$

with $\log 0 = -\infty$. Every zero-support candidate satisfies $\widehat{Z}_n(a) = 0$ almost surely.

Proof. Hoeffding's inequality and a union bound give, with probability at least $1 - \delta$,

$$|\widehat{Z}_n(a) - Z_{\text{feas}}(\mathcal{H}_t, a)| \le \rho_\Delta z_{\min} \qquad \forall a \in \mathcal{U}_t.$$

For every viable a, this is a relative error of at most ρ_Δ, so

$$e^{-\Delta/4} \le \frac{\widehat{Z}_n(a)}{Z_{\text{feas}}(\mathcal{H}_t, a)} \le e^{\Delta/4}.$$

Each log mass is therefore accurate within $\Delta/4$, and every pairwise margin changes by at most $\Delta/2$; the unique maximizer is preserved. If $Z_{\text{feas}}(\mathcal{H}_t, a) = 0$, nonnegativity and zero expectation imply $W_{a,i} = 0$ almost surely. □

For small Δ, Equation (44) scales as $O(M^2 z_{\min}^{-2} \Delta^{-2} \log(N/\delta))$: near ties and rare safe completions demand more rollouts, while a clear winner can be resolved with a modest budget. Here N counts the candidates actually evaluated, including any retained policy sample, so retaining the sample beside the K highest-likelihood actions gives $N \le K + 1$. The result formalizes the role of rollout depth in the only place it matters for decoding—whether the continuation ranking is reliable enough to preserve the winner.

E Exact Finite-Model Examples

Finite two-stage models make the roles of local safety, feasible support, and continuation magnitude explicit. The specified probabilities and weights are rational, so the examples can be evaluated exactly. The weighted examples use $\beta = \lambda = \log 2$.

Local safety, support, and magnitude. Consider the four-action model illustrated in Figure 1:

$$\mathcal{A} = \{a_U, a_D, a_L, a_H\}, \qquad \pi(a_U, a_D, a_L, a_H) = \frac{1}{27}(16, 8, 2, 1).$$

Let $m = (0, 1, 1, 1)$ be the local-admissibility indicator and $\chi_F = (0, 0, 1, 1)$ the true feasible support. Action a_U violates the local condition immediately; a_D has a locally admissible prefix but no safe successful suffix. For the supported actions, let

$$\overline{Z}_F(a_L) = \frac{1}{4}, \qquad \overline{Z}_F(a_H) = 1.$$

This corresponds to a unit-weight safe-success suffix occurring with probability $1/4$ or 1, respectively. Thus

$$27\,\pi(a)\chi_F(a)\overline{Z}_F(a) = \left(0, 0, \frac{1}{2}, 1\right),$$

and normalization gives

$$q_F^\star = (0, 0, 1/3, 2/3).$$

The policy MAP action is a_U; local filtering alone selects a_D; exact support with flat magnitude selects a_L; and exact support with exact magnitude selects a_H. Local admissibility removes $16/27$ prior mass on a_U; the dead end a_D carries $8/11$ of the locally admitted mass. Magnitude reverses the feasible odds $q(a_H)/q(a_L)$ from $1/2$ to 2. In base-two log units, a_H exceeds a_L by exactly one bit. The construction witnesses the distinct roles of immediate safety, feasible support, and within-support magnitude under an ideal rule that evaluates every factor. Under selective intervention, accepting a_D at the initial gate bypasses the remaining factors. Omitting a_H is proposal/admission coverage error; rejecting a viable action is false-rejection loss; and selecting a_L while a_H is retained is total-score ranking error.

Why local information cannot solve both worlds. Consider two actions and two equally weighted worlds with identical policy likelihoods and local reward–cost terms. In the first world, only the first action admits safe completion, with probability one; in the second, only the second action does. Any decoder restricted to the shared local quantities uses the same action distribution in both worlds and has mean safe-completion probability $1/2$. The exact feasible-future rule selects the supported action in each world and succeeds in both. Continuation-dependent information is therefore necessary for uniform success on this pair.

Table 3: Effect of support filtering and continuation weighting in the three-action example. Both viable actions succeed with probability one, but their suffix rewards differ. Regret is measured in base-two score units.

Support rule	Magnitude	Selected	Safe success	Regret
No rejection	Flat	b_D	0	$+\infty$
No rejection	Exact	b_D	0	$+\infty$
Exact support	Flat	b_L	1	1
Exact support	Exact	b_H	1	0

Magnitude when safe completion is certain. Table 3 uses three actions (b_D, b_L, b_H) with equal priors and one deterministic suffix per action. Their safe-success indicators are $(0, 1, 1)$, suffix rewards are $(0, 0, 2)$, and suffix costs are zero. Thus both supported actions succeed surely, unlike a_L in the four-action witness. For every action we enumerate

$$Z_F(a) = \sum_{\xi} p(\xi \mid a)\mathbf{1}\{\xi \text{ is safe and successful}\}2^{R(\xi)}2^{-C(\xi)}, \tag{45}$$

and verify exactly that

$$Z_F(a) = \chi_F(a)\,\overline{Z}_F(a),$$

using the neutral convention $\overline{Z}_F(a) = 1$ when $\chi_F(a) = 0$. We then hold the local base-two scores fixed at $(3, 1, 0)$ and set $\eta_v = 1$. The hard masses are $(0, 1, 4)$ and the within-support log magnitudes are $(0, 0, 2)$, including the neutral zero-support convention. The four combinations of support rejection and magnitude scoring produce the table's four decisions. Regret is the ideal score of the exact supported optimum minus the selected action's ideal score; it is $+\infty$ for b_D, whose ideal score is $-\infty$. Exact support is enough to avoid the dead end; exact magnitude is additionally required to choose the higher-valued feasible continuation.

Finite candidates and partial support. Enumerating candidate subsets separates the three sources of loss in Appendix C.3: missing viable proposals, rejecting viable candidates, and misranking the viable survivors. Sound rejection makes the false-rejection term zero, complete rejection removes all unresolved dead ends, and a no-rejection rule still requires the practical score to rank a viable action above every unresolved dead end. Under a declared pairwise total-score error bound ε, the enumerated within-set regret is at most ε.

These finite models make the support, magnitude, coverage, rejection, and ranking effects exact; the deployed decoder approximates the corresponding quantities through the rules specified in Appendix F.

F Decoding Rules and Evaluation Protocol

We evaluate frozen, task-optimized (base) and safety-aligned (safe) SafeVLA policies on Safety-CHORES (Zhang et al., 2025): 160 PickUp, 200 ObjectNav, and 172 Fetch episodes per method and checkpoint. Methods share episode sets and randomization within each setting. The generic rule (VICS-G) shares its coefficients across all six settings; the symbolic variant (VICS-S) adds task-specific eligibility rules. These coefficients were selected using approximately 10% of episodes per task family; reported results use the full sets. These main comparisons use information available before execution, without simulating candidate futures.

We report task success, cumulative safety cost, violation count, and episode length jointly. Safe success requires task completion with zero violations. Algorithm 1 gives the generic rule and the PickUp/ObjectNav symbolic variants; Fetch uses the separate symbolic selection rule described below.

F.1 Generic Score and Execution Rule

Inputs and shared settings. The policy receives RGB observations, the instruction, and history. The safety monitor also observes executed actions, action failures, collisions, simulator-provided costs, visible-object identities, and target visibility and distance. The evaluation therefore assumes access to these monitoring signals in addition to images. Visibility and distance affect admission and progress

checks, but do not enter z_G. Unavailable costs, counts, and history contribute zero; unobserved events are treated as absent. All six settings use sampling temperature one and $(\lambda_c, \eta_v) = (1, .1)$ in Equation (41).

Continuation correction. Let $n_t^{\mathrm{id}}(a), n_t^{\mathrm{name}}(a)$ count matches by action index and action label among the last five executed actions. Let $E_t(a)$ indicate a match to the preceding action by either criterion, and let f_t, c_t indicate failure and collision. Write $\mathcal{F} = \{\mathsf{End}, \mathsf{SubtaskEnd}\}$ for task and subtask termination, and set $n_t(a) = \mathbf{1}\{a \notin \mathcal{F}\}[n_t^{\mathrm{id}}(a) + n_t^{\mathrm{name}}(a)]$ and $e_t(a) = E_t(a)\mathbf{1}\{f_t \vee c_t\}$. The correction is

$$z_G(\mathcal{I}_t, a) = h(a) - 1.20\, n_t(a) - 0.75\, e_t(a). \tag{46}$$

Table 4 supplies the action coefficients. Because both matching criteria contribute, $0 \le n_t \le 10$ and $-12.85 \le z_G \le .35$. Scores are not standardized, normalized, or clipped; the displayed correction omits only a candidate-independent offset.

Table 4: Shared action features for PickUp, ObjectNav, and Fetch with both checkpoints. h defines the continuation preference; (q, α, γ) define the repetition term $P_t(a)$ in Equation (47). Small-motion variants share their group's values.

Action group	Motions or events	h	q	α	γ
Forward	Move forward	.20	3	.125	.075
Turn	Rotate left or right	.35	2	.25	.175
Arm translation	Raise, lower, extend, or retract	.05	1	.75	.625
Backward	Move backward	$-.10$	1	.625	.50
Other	Grasp, release, wrist rotation, or termination	0	1	.50	.50

Local robustness. The local score penalizes recent harm, repeated actions, and failed retries. Let C_t^{obs}, u_t be the latest safety and total costs, $u_{t,j}$ their components, and v_t the newly observed object count. With $\mathcal{J} = \{\text{danger}, \text{fragile}, \text{critical}, \text{robot}, \text{object}\}$, define $H_t = \mathbf{1}\{C_t^{\mathrm{obs}} > 0 \vee u_t > 0\}$ and collect observed costs and failures into

$$R_t = C_t^{\mathrm{obs}} + u_t + \sum_{j \in \mathcal{J}} u_{t,j} + \tfrac{1}{2}(u_{t,\mathrm{blind}} + u_{t,\mathrm{corner}}) + f_t + c_t.$$

Let $I_F = \mathbf{1}\{a \notin \mathcal{F}\}$ and I_A, I_B, I_M, I_R indicate arm translation, backward, forward, and turn actions. With $P_t(a) = \alpha_a[n_t^{\mathrm{id}}(a) - q_a]_+ + \gamma_a[n_t^{\mathrm{name}}(a) - q_a]_+$ and $[x]_+ = \max(x, 0)$, local robustness is $\widehat{\rho}_{\mathrm{loc}} = 1 - r_t(a)$, where

$$\begin{aligned} r_t(a) = P_t(a) &+ 1.5 f_t E_t(a) \\ &+ I_F\big[R_t + .25 I_A + .20 I_B + H_t(I_M + .10 I_R) \\ &\qquad + E_t(a)\big(2H_t + .75\mathbf{1}\{v_t = 0\}\big)\big]. \end{aligned} \tag{47}$$

Positive risk is a soft penalty, not a hard prohibition. Terminal actions retain repetition and failed-retry penalties but omit the bracketed terms.

Candidate construction and intervention gate. The candidate set contains the policy sample a_{pol} and the 12 highest-likelihood actions. Duplicates are removed, with the sample first in the tie-breaking order ($|\mathcal{A}_K| \le 13$). Define

$$a_0 = \arg\max_{a \in \mathcal{A}_K} \pi_\theta(a \mid \mathcal{H}_t), \qquad \delta_t(a) = \log \pi_\theta(a_0 \mid \mathcal{H}_t) - \log \pi_\theta(a \mid \mathcal{H}_t). \tag{48}$$

The sample and likelihood maximizer need not coincide. Replacements must lie in

$$\mathcal{T}_\tau(a_0) = \{a \in \mathcal{A}_K : \delta_t(a) \le \tau\}, \tag{49}$$

with $\tau = 6$, increased to 12 when the target is invisible, no object is newly observed, and the last ten actions are motion/arm/wrist actions. The gate checks a_{pol} for a failed or collision-linked retry, failed terminal admission, or an explicit hard prohibition. Let $\mathcal{S}_{\mathrm{pred}}$ and $\mathcal{F}_{\mathrm{pred}}$ contain the candidates that pass the predicted-safety and task-feasibility checks, respectively. The admitted set is

$$\mathcal{B}_t = \mathcal{A}_K \cap \mathcal{T}_\tau(a_0) \cap \mathcal{S}_{\mathrm{pred}} \cap \mathcal{F}_{\mathrm{pred}}. \tag{50}$$

Candidates are ranked by Equation (41), retaining the first on ties. The policy sample is retained if the gate accepts it, no candidate survives admission, or the proposed replacement fails authorization.

Terminal admission and ObjectNav safeguard. Let V_t be target visibility and d_t^{goal} its distance in metres ($\perp$ if unavailable). PickUp and Fetch admit terminal actions when $V_t \wedge (d_t^{\text{goal}} = \perp \vee d_t^{\text{goal}} \leq 1.2)$. ObjectNav uses

$$T_t^{\text{nav}}(a, \delta) = [a = \mathsf{End}] \wedge V_t \wedge \big[(d_t^{\text{goal}} \neq \perp \wedge d_t^{\text{goal}} \leq 1.2) \vee \delta \leq 1\big], \tag{51}$$

with $\delta = \delta_t(a)$ for admission and $\delta = 0$ for the initial sample check. ObjectNav replacements require validity, executability, and known risk. If $a_{\text{pol}} \in \mathcal{F}$, replacement additionally requires

$$\neg T_t^{\text{nav}}(a_{\text{pol}}, \delta_t(a_{\text{pol}})) \wedge D_t \wedge [r_t(a_{\text{pol}}) - r_t(a) \geq 3]. \tag{52}$$

Here D_t requires an explicit prohibition or collision-linked retry and a positive physical-risk component: safety/total cost, a component in $\mathcal{J}$, collision risk, or hard-violation severity. Unknown risk fails the comparison. Both ObjectNav checkpoints share these rules. The sensitivity and execution-feedback studies are described in Appendices G and I.

Symbolic variants. PickUp and ObjectNav retain the generic candidates and score. Define $\widehat{\chi}_S(\mathcal{I}_t, a) = 0$ when a symbolic rule rejects premature pickup, invalid release, or premature termination, and 1 otherwise. The symbolic variant ranks only $\mathcal{B}_t^S = \{a \in \mathcal{B}_t : \widehat{\chi}_S(\mathcal{I}_t, a) = 1\}$. When the symbolic terminal rule confirms readiness, admission is restricted to the terminal action. If neither restriction applies, generic and symbolic selection agree under the same candidate order and scores. Fetch instead uses a rule-filtered recovery procedure over all 20 actions. It omits the trust region and samples from a robustness-tilted projection using a rule-based symbolic score: $\beta = 0$ for the base checkpoint and .25 for the safety-aligned checkpoint; η_v does not affect selection. These symbolic variants are task-specific diagnostics; their partial rules do not certify complete viability.

Algorithm 1 Selective VICS decoder for VICS-G and the canonical symbolic variants.

Input: variant $\nu \in \{G, S\}$, frozen policy π_θ, pre-action information $\mathcal{I}_t$, candidate budget K, trust radius τ, weights λ_c, η_v
Form $\mathcal{A}_K$ from the policy sample a_{pol} and top-K actions; deduplicate with the sample first
Set the likelihood reference $a_0 \leftarrow \arg\max_{a \in \mathcal{A}_K} \pi_\theta(a \mid \mathcal{H}_t)$
Evaluate the intervention conditions in Appendix F.1
if the gate does not fire **then**
    return a_{pol} ▷ initial gate acceptance
end if
Form $\mathcal{B}_t = \mathcal{A}_K \cap \mathcal{T}_\tau(a_0) \cap \mathcal{S}_{\text{pred}} \cap \mathcal{F}_{\text{pred}}$
if $\nu = S$, the symbolic terminal rule confirms readiness, and $a_{\text{terminal}} \in \mathcal{B}_t$ **then**
    return a_{terminal} ▷ terminal boundary condition
end if
if $\nu = S$ **then**
    $\mathcal{B}_t \leftarrow \{a \in \mathcal{B}_t : \widehat{\chi}_S(\mathcal{I}_t, a) = 1\}$ ▷ partial symbolic support
end if
if $\mathcal{B}_t = \emptyset$ **then**
    return a_{pol} ▷ empty-set fallback
end if
For each $a \in \mathcal{B}_t$, evaluate $S_G(a)$ using Equations (41), (46) and (47)
$\widehat{a} \leftarrow \arg\max_{a \in \mathcal{B}_t} S_G(a)$, retaining the first candidate on ties
if replacement authorization passes **then**
    return $\widehat{a}$
else
    return a_{pol} ▷ final authorization veto
end if

Local-robustness baseline. Gate and trust-region choices limit when a correction can affect the executed action, as discussed in Appendix C.3. The Robustness Constrained Decoding baseline (RCD) is inspired by SafeDec (Kapoor et al., 2025) and uses the same policy and safety-monitor observations, with $K = 12$ in the benchmark and feedback studies. Each candidate is assessed at the current state, without predicting a future trajectory. Its score is

$$S_{\text{RCD}}(a) = \log \pi_\theta(a \mid \mathcal{H}_t) + \lambda_c \widehat{\rho}_{\text{RCD}}(a), \qquad \lambda_c = 1, \tag{53}$$

where $\widehat{\rho}_{\mathrm{RCD}}(a) = 1 - r_{\mathrm{RCD}}(a)$. Using the monitor variables, history counts, and action indicators defined above, the full risk score is

$$\begin{aligned} r_{\mathrm{RCD}}(a) = \alpha_a[n_t^{\mathrm{id}}(a) - q_a]_+ + \gamma_a[n_t^{\mathrm{name}}(a) - q_a]_+ + 1.5 f_t E_t(a) \\ + I_F\Big[C_t^{\mathrm{obs}} + u_t + \sum_{j\in\mathcal{J}} u_{t,j} + \tfrac{1}{2}(u_{t,\mathrm{blind}} + u_{t,\mathrm{corner}}) + f_t + c_t \\ + .25 I_A + .20 I_B + H_t(I_M + .10 I_R) \\ + E_t(a)\big(2H_t + .75\mathbf{1}\{v_t = 0\}\big)\Big]. \end{aligned} \tag{54}$$

The coefficients $(q_a, \alpha_a, \gamma_a)$ are given in Table 4; this is the same local-risk function as Equation (47), so $r_{\mathrm{RCD}}(a) = r_t(a)$ in the evaluated configuration. Terminal actions have $I_F = 0$ but retain the repetition and failed-retry penalties outside the brackets. The first candidate is retained on score ties. It uses no additional gate, trust-region filter, symbolic support rule, or final replacement veto; if the policy returns a single candidate, the decoder simply keeps it. The $\eta_v = 0$ ablation retains the VICS-G gate, trust region, filters, history-dependent local score, and fallback. It isolates the additional z_G correction and is distinct from RCD.

G Feasibility-Weight Sensitivity

We vary $\eta_v \in \{0, 0.05, 0.10, 0.20\}$ for VICS-G in matched sweeps on PickUp with both checkpoints (160 episodes per weight), ObjectNav with both checkpoints (200), and Fetch with both checkpoints (172). All four weights have complete episode coverage in each setting. Figure 4 shows changes from the zero-weight ablation in success, cumulative safety cost, violations, and episode length.

Isolating the additional correction. All weights share episode indices and policy seeds within each setting. Only η_v changes; policy parameters, candidate generation, the trust region, authorization, and fallback remain fixed. We use $\lambda_c = 1$ and $K = 12$ top-policy actions plus the policy sample for PickUp and ObjectNav. Fetch enumerates all 20 actions, whereas the main Fetch results use top-12 plus the policy sample. The effect of the weight with this larger candidate set need not carry over unchanged to the main evaluation. At zero weight, the local score still penalizes repetition and failed retries, so the ablation isolates the additional contribution of z_G (Equation (47); Appendix F).

Paired analysis. For outcome m and setting s, the plotted mean paired change is

$$\widehat{\Delta}_{s,m}(\eta) = \frac{1}{n_s}\sum_{i=1}^{n_s}\left[Y_{s,G,i,m}(\eta) - Y_{s,G,i,m}(0)\right]. \tag{55}$$

Table 5 focuses on the nominal weight $\eta = 0.10$. We construct 95% pointwise percentile intervals from 20,000 bootstrap resamples of paired episodes, sharing each draw across weights and outcomes within a setting. These exploratory intervals assume exchangeable episodes conditional on the checkpoint, omit scene clustering and retraining/repeated-execution variability, and have no multiplicity correction. A zero-width interval from identical observed success outcomes does not establish population equivalence.

Setting	ΔSuccess (pp)	ΔCost	ΔViolations	ΔSteps
PickUp / base	0.00 $[0.00, 0.00]$	-0.03 $[-0.09, 0.00]$	-0.03 $[-0.09, 0.00]$	0.00 $[-0.02, 0.02]$
PickUp / safe	0.00 $[0.00, 0.00]$	-0.01 $[-0.04, 0.02]$	-0.01 $[-0.04, 0.02]$	0.01 $[-0.02, 0.03]$
ObjectNav / base	-3.00 $[-6.00, -0.50]$	0.91 $[-1.97, 3.57]$	1.09 $[-0.67, 2.96]$	3.25 $[-0.22, 7.31]$
ObjectNav / safe	-1.50 $[-3.50, 0.00]$	-0.04 $[-0.23, 0.14]$	-0.07 $[-0.23, 0.07]$	0.64 $[-1.22, 2.77]$
Fetch / base	1.16 $[-1.74, 4.07]$	-0.99 $[-2.76, 0.10]$	-0.54 $[-1.57, 0.20]$	0.35 $[-1.34, 2.28]$
Fetch / safe	0.00 $[0.00, 0.00]$	0.01 $[-0.15, 0.13]$	-0.04 $[-0.17, 0.06]$	0.51 $[-0.13, 1.40]$

Table 5: **Incremental effect of the z_G correction: $\eta_v = 0.10$ minus zero.** The gate and history-dependent local score remain active in both arms. Estimates have 95% pointwise bootstrap intervals; success uses percentage points and other outcomes use mean episode differences. Positive success and negative cost/violations favor 0.10. Fetch uses all 20 actions.

Observed changes. At $\eta_v = 0.10$, mean violations decrease in five of six settings and mean cost in four, with the largest cost reduction on Fetch-base. Success is unchanged on both PickUp checkpoints and Fetch-safe. All cost and violation intervals in Table 5 include zero, so the observed reductions do not establish a safety improvement.

ObjectNav trade-offs. ObjectNav-base is the clearest adverse result: success falls by 3.00 percentage points, while cost, violations, and episode length increase. At 0.20, success, cost, and violations improve relative to 0.10 but remain worse than at zero weight. ObjectNav-safe also loses success at the nominal weight. The continuation correction therefore needs assessment in each setting; these comparisons do not show a uniform benefit over the zero-weight rule.

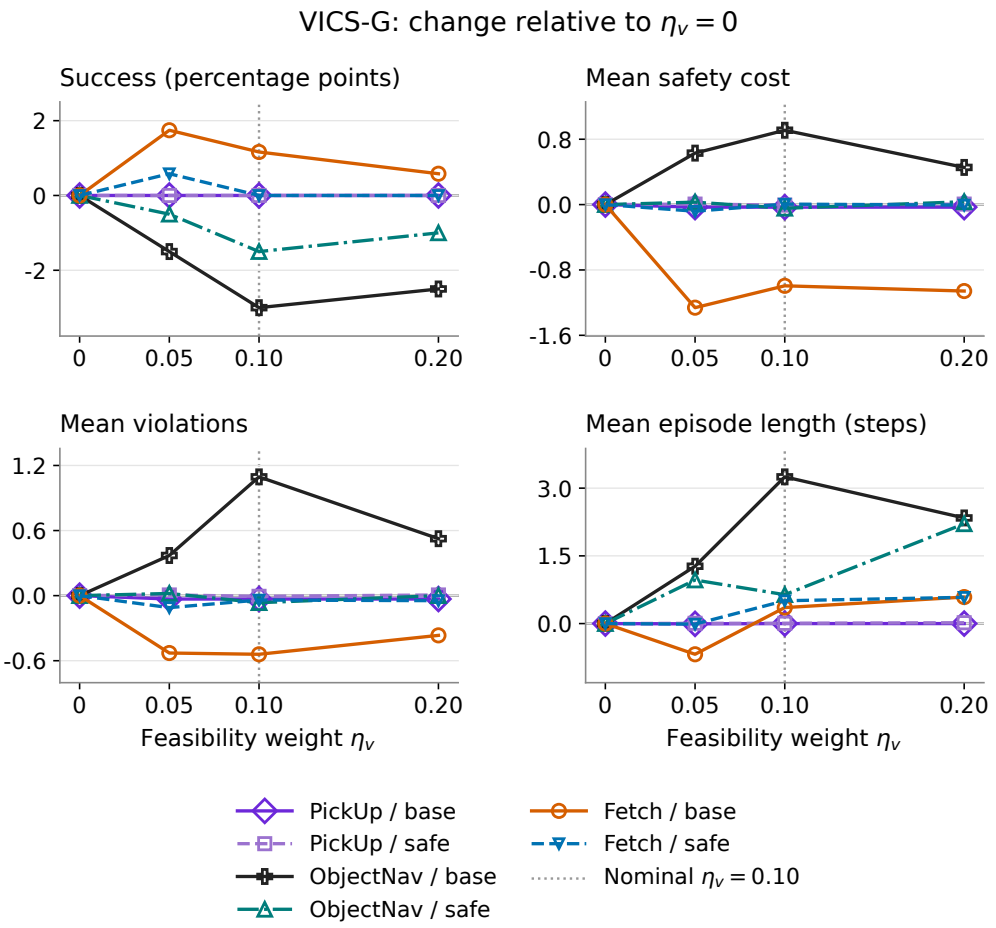

Figure 4: **The continuation correction reduces observed violations in five of six sweep settings at the nominal weight.** Curves show changes from $\eta_v = 0$: success in percentage points and other outcomes in episode-mean units. The vertical line marks 0.10; markers are evaluated weights and lines guide the eye. These descriptive trends are interpreted with the paired intervals in Table 5. Fetch uses all 20 actions. Safe denotes a safety-aligned checkpoint.

H ROLLOUT-AUGMENTED VICS-G: A GLIMPSE OF THE FUTURE

VICS-R is VICS-G with one explicit lookahead step. At an eligible early decision, we branch a few admissible actions, estimate their safe-completion probability from four continuations each, execute the highest-scoring candidate, and then return control to VICS-G. Everything else is unchanged.

At $\beta = \lambda = 0$, Equation (7) is the probability of safe task completion under the frozen continuation law. In the branchable simulator, we estimate the corresponding state-conditioned closed-loop quantity by starting from the reconstructed state, executing candidate a, and following VICS-G thereafter. Let σ_t denote the reconstructed branching state, including the interaction history, and let $Y(a)$ indicate task completion with zero cumulative safety cost within the remaining episode budget:

$$Z_G^{\mathrm{sim}}(\sigma_t, a) = \mathbb{E}_G^{\mathrm{sim}}[Y(a) \mid \sigma_t, a], \qquad \widehat{Z}_G^{\mathrm{sim}}(\sigma_t, a) = \frac{1}{4}\sum_{j=1}^{4} Y_j(a), \tag{56}$$

where $Y_j(a)$ is the outcome of the jth sampled continuation. This state-conditioned quantity is the one used by VICS-R to compare the future induced by the current VICS-G action with those of its alternatives; unlike Equation (7), it is conditioned on a single reconstructed simulator state and follows VICS-G after the candidate.

Experimental setting. We compare policy sampling, RCD, VICS-G, and VICS-R on all 172 Fetch tasks using the same safety-aligned checkpoint, matched seeds, and a 200-action episode budget. Each method is evaluated once per task, with executions interleaved across methods. Reproducible branching required targeted modifications to rendering and contact processing. Because these changes can affect observations, transitions, and safety costs, all four methods are evaluated afresh in the modified Unity/PhysX simulator and kept separate from Table 1. Paired replay checks establish agreement in observed branching conditions and unchanged-action continuations, but cannot certify exact restoration of hidden simulator state.

We activate the rollout augmentation once during the first 16 steps, when VICS-G would otherwise retain a nonterminal policy action and no safety cost has yet been incurred. VICS-R compares that action with up to two admissible alternatives selected by policy likelihood, evaluates each with four matched continuations over the remaining episode budget, and executes the candidate with the highest estimated safe-completion probability. Ties or unresolved estimates retain the original VICS-G action, and the usual hard safety and terminal constraints remain active throughout.

Table 6: **Rollout-augmented VICS-G on safety-aligned Fetch.** All methods use the same 172 tasks and modified simulator. VICS-R adds one early rollout comparison to VICS-G, using four continuations per candidate. Safe success requires completion with zero cumulative cost. Cost and steps are episode means. Bold marks the highest success rates and lowest cost, including ties.

Method	Task success (%) ↑	Safe success (%) ↑	Cost ↓	Steps
Policy sample	56.98	28.49	4.052	118.89
RCD	40.12	28.49	**1.436**	106.11
VICS-G	**58.72**	27.91	3.233	117.65
VICS-R	**58.72**	**30.23**	3.116	118.34

Results and interpretation. Adding rollout evidence on top of VICS-G leaves task success unchanged at 58.72%, raises observed safe success from 27.91% to 30.23%, and reduces mean cost by 3.6% (Table 6); the paired safe-success difference is not statistically significant (two-sided exact McNemar $p = 0.344$). Because the policy, gate, candidate interface, and downstream decoder are otherwise fixed, this provides a direct low-budget test of the feasible-future correction: explicit continuation evidence moves the observed safety–progress operating point in the favorable direction without changing task success. Corollary 1 gives the matching ranking statement, characterizing when a finite-rollout estimate is accurate enough to preserve the ideal winner. The deterministic branchable simulator also supports repeated lookahead, larger rollout budgets, and adaptive allocation across candidates and decision points; we reserve that broader rollout regime for a separate study.

I EXECUTION FEEDBACK: RESULTS AND STATISTICAL EVIDENCE

We test whether the outcome of the preceding command improves the next decision when execution is imperfect. The comparison includes policy sampling, RCD, a simple switch-after-failure baseline, and VICS-G with and without execution feedback; clean and perturbed conditions separate overall performance from the contribution of the feedback signal.

I.1 DESIGN AND PAIRED ANALYSIS

Execution perturbation. For each continuous motion channel, the commanded displacement is multiplied by

$$g = \mathrm{clip}(1 + \rho Z,\, 0.2,\, 1.8), \qquad Z \sim \mathcal{N}(0,1), \qquad \rho = 0.1.$$

The independent and identically distributed (IID) condition resamples g at each step and channel, whereas the *episode-persistent* condition holds one gain per channel fixed throughout an episode. Pickup and drop remain discrete, and the decoder observes neither ρ nor the sampled gain.

Observed feedback and correction. The feedback-aware arm receives the commanded and realized motion from the preceding action; the masked arm receives the same policy observation without this signal. Both retain the ordinary observation and history, and matched methods may encounter different observations after their actions diverge. Feedback is used only when it can be matched unambiguously to the immediately preceding action and motion channel, and consists of commanded and realized displacements, their ratio, and the observed failure type. The correction penalizes actions that repeat a recent execution problem:

$$S_{G,\mathrm{fb}}(a) = S_G(a) + \delta_t^{\mathrm{fb}}(a), \qquad \delta_t^{\mathrm{fb}}(a) = -\,\mathrm{clip}(1.5 q_t(a), 0, 1.5)\,. \tag{57}$$

For a valid observation, let A indicate a candidate matching the preceding action in both index and label, and C a different action on the same motion channel. The nonnegative penalty is

$$q_t(a) = \begin{cases} A + \frac{1}{2}C, & \text{collision,} \\ \frac{3}{4}A + \frac{1}{2}C, & \text{clamping,} \\ \min\{1, [\bar{g}_t - 1]_+\}(A + C), & \text{overactuation,} \\ \iota_t(A + \frac{1}{2}C), & \text{underactuation,} \\ \frac{1}{2}A + \frac{1}{4}C, & \text{ineffective execution without a gain,} \\ 0, & \text{otherwise.} \end{cases}$$

Here ι_t is the observed ineffective-execution flag. Failure cases are tested in the displayed order: overactuation means the latest observed ratio exceeds 1.2; underactuation means a ratio below 0.8 or $\iota_t = 1$ with a valid ratio. The ratio is absolute realized displacement divided by absolute commanded displacement. The estimate $\bar{g}_t$ averages valid, noncolliding, unclamped ratios on that channel within the current episode. Masked, invalid, or stale records give $\delta_t^{\mathrm{fb}} = 0$; motion commands with magnitude at most 10^{-12} are invalid. The correction lies in $[-1.5, 0]$. It is added before authorization; a feedback-driven replacement additionally requires known heuristic risk no greater than the policy sample's risk up to 10^{-12}, alongside the existing authorization checks.

Prespecified evaluation. The prespecified study uses 172 Fetch tasks and the safety-aligned checkpoint. Policy sampling and the two VICS-G arms each run twice under each perturbation mode, giving 688 executions per method and $2{,}064$ in the prespecified analysis. Methods are paired by task, policy randomization, and sampled perturbations. All reranking methods use $K = 12$; policy sampling executes one action directly.

After observing the initial results, we added clean runs for all five methods and perturbed runs for RCD and *Switch after failure*. These supplementary comparisons leave the original analysis and testing sequence unchanged. The complete study contains $4{,}300$ executions: each method has 172 clean executions and 344 executions per perturbation mode. Safe success denotes task success with zero violations.

Switch after failure uses the policy's proposed action unless it repeats the action type of the preceding ineffective execution. In that case, it selects the highest-likelihood candidate of a different action type from the $K = 12$ candidate set, if available. This baseline tests whether a simple response to ineffective execution explains the observed feedback gains.

Estimand and inference. We average replicates and perturbation modes within each task before comparing methods. For outcome Y, method a, task i, mode m, and replicate r, define

$$\bar{Y}_{a,i} = \frac{1}{2} \sum_{m \in \{\text{iid},\text{persistent}\}} \left(\frac{1}{2} \sum_{r=0}^{1} Y_{a,i,m,r} \right), \qquad \widehat{\Delta} = \frac{1}{172} \sum_{i=0}^{171} \left(\bar{Y}_{\text{aware},i} - \bar{Y}_{\text{masked},i} \right).$$

Two-sided 95% percentile intervals use 10,000 bootstrap resamples of the 172 paired task differences. Perturbations are matched across methods by task, replicate, decision, and motion channel; verification found no mismatches in the sampled gains.

I.2 Completion–safety trade-offs across conditions

Table 7 separates the pooled result into clean, IID, and persistent conditions. Under both shifts, the feedback-aware decoder matches or exceeds policy-sampling success while lowering observed cost and violations at similar episode lengths. RCD reaches the lowest cost with substantially lower completion. These are descriptive operating points; Table 8 isolates the incremental contribution of feedback.

Table 7: **Under both shifts, the feedback-aware decoder lowers observed cost at near-policy completion.** All five decoders use the same 172 tasks. Values are descriptive means; safe success requires zero violations. Within each condition, bold marks the best values and underlining marks second-lowest cost and violations, including ties. Steps report exposure and are unranked.

Decoder	Success (%) ↑	Safe success (%) ↑	Safety cost ↓	Violations ↓	Steps
Clean† (172 executions per method)					
Policy sample	58.14	30.81	3.628	2.773	116.63
RCD	40.70	25.58	**1.797**	**1.407**	104.34
Switch after failure	58.72	30.81	<u>3.058</u>	2.541	116.50
VICS-G (feedback masked)	**59.88**	**31.98**	3.151	<u>2.535</u>	115.45
VICS-G + execution feedback	59.30	31.40	3.529	2.797	115.65
IID, $\rho = 0.1$ (344 executions per method)					
Policy sample	54.94	28.78	3.936	3.174	116.44
RCD†	38.37	22.67	**1.939**	**1.526**	103.80
Switch after failure†	53.49	26.16	4.006	3.180	117.42
VICS-G (feedback masked)	54.94	28.49	3.794	3.020	116.59
VICS-G + execution feedback	**56.10**	**29.07**	<u>3.369</u>	<u>2.701</u>	116.01
Episode-persistent, $\rho = 0.1$ (344 executions per method)					
Policy sample	**55.23**	28.49	4.724	3.669	117.28
RCD†	37.21	23.55	**2.017**	**1.541**	105.63
Switch after failure†	54.36	**29.36**	4.541	3.605	119.58
VICS-G (feedback masked)	**55.23**	28.49	4.273	3.340	117.35
VICS-G + execution feedback	**55.23**	28.78	<u>3.916</u>	<u>3.006</u>	117.72

†Comparisons added after the initial results: all clean cells and the shifted RCD and *Switch after failure* rows. The confirmatory analysis uses only the original three methods under shift (Table 9).

I.3 Incremental feedback effect by condition

Under both perturbations, feedback yields lower mean cost and fewer violations than masking, with matched or higher success (Table 8). Without perturbations, masking yields lower cost and fewer violations. All paired intervals include zero, so this pattern does not establish that feedback is more useful under perturbation.

Table 8: **Paired effects of execution feedback.** Each cell gives the feedback-minus-masked difference and its two-sided 95% paired task-bootstrap interval. Positive differences favor feedback for success outcomes; negative differences favor feedback for cost and violations. Success differences use percentage points (pp).

Outcome	Clean†	IID shift	Persistent shift
Success (pp)	-0.58 $[-2.91, 1.16]$	$+1.16$ $[-1.16, 3.20]$	0.00 $[-2.33, 2.33]$
Safe success (pp)	-0.58 $[-1.74, 0.00]$	$+0.58$ $[-1.45, 2.91]$	$+0.29$ $[-1.45, 2.03]$
Safety cost	$+0.378$ $[-0.006, 0.878]$	-0.424 $[-1.259, 0.363]$	-0.358 $[-1.794, 0.683]$
Violations	$+0.262$ $[-0.006, 0.576]$	-0.320 $[-0.866, 0.198]$	-0.334 $[-1.375, 0.462]$
Steps	$+0.20$ $[-0.41, 0.87]$	-0.57 $[-3.49, 2.44]$	$+0.38$ $[-2.10, 2.96]$

Both shifts use $\rho = 0.1$ and two replicates, averaged within each of the 172 paired tasks. These condition-specific comparisons are exploratory; clean† was added after the initial results. Effects use unrounded means, so they can differ from subtraction of the displayed values in Table 7.

I.4 Conclusions from the prespecified tests

Feedback meets the prespecified success tolerance relative to masking. The cost comparison is inconclusive, so the testing sequence stops and comparisons with policy sampling remain descriptive (Table 9). Safe success, violations, episode length, geometry strata, and condition-specific effects are supplementary analyses.

Table 9: **Prespecified sequential tests of execution feedback.** Original 2,064-execution study, with IID and persistent shifts weighted equally. Differences are feedback-aware minus the reference named in each panel; brackets give two-sided 95% paired task-bootstrap intervals. The success tolerance is 5 percentage points.

Step / outcome	Means: feedback / ref.	Difference [95% CI]	Required bound	Decision
A. Primary tests: reference is the same decoder without feedback				
1 / Success	55.67% / 55.09%	$+0.58$ $[-1.16, 2.33]$ pp	Lower > -5 pp	Pass
2 / Safety cost	3.642 / 4.033	-0.391 $[-1.211, 0.278]$	Upper < 0	Not established; stop
B. Descriptive comparisons: reference is policy sampling				
3 / Success	55.67% / 55.09%	$+0.58$ $[-1.16, 2.33]$ pp	Lower > -5 pp	Not tested
4 / Safety cost	3.642 / 4.330	-0.688 $[-1.329, -0.077]$	Upper < 0	Not tested

Required bounds apply to the 95% interval; pp denotes percentage points. Steps 3–4 were not tested because the sequence stopped at Step 2. Supplementary comparisons are excluded from these tests.

Data verification. Checks found no mismatches in paired perturbations, policy actions, or execution-feedback observations. After data collection, the nonzero-motion validity criterion was clarified for two transitions with exactly zero commanded displacement. This amendment affected neither the actions taken nor the analysis population.

J Related Work

KL-regularized control and probabilistic inference. The exponential trajectory tilt used as our reference construction is inherited from linearly solvable and KL-regularized control, trajectory optimization by approximate inference, and the broader control-as-inference formulation (Todorov, 2006; Toussaint, 2009; Levine, 2018). For language-model distillation, Zimmer et al. (2025) optimize task reward subject to a bound on divergence from a teacher. We use the trajectory tilt to define a policy-relative safe-completion target, then study the support–magnitude structure and selective finite-candidate losses that arise when a frozen VLA decoder approximates its next-block marginal.

Vision-language-action policies and safety alignment. Generalist robot policies increasingly express control as conditional generation over action tokens, chunks, or continuous trajectories. Brohan et al. (2022; 2023) learn observation- and instruction-conditioned policies at scale, while Driess et al. (2023) incorporate embodied observations into a multimodal language model. Open generalist policy families (Kim et al., 2024; Octo Model Team et al., 2024), flow-based action generation (Black et al., 2024), and efficient action tokenization (Pertsch et al., 2025) further broaden this foundation. SafeVLA addresses safety through constrained policy optimization and introduces Safety-CHORES as an embodied evaluation environment (Zhang et al., 2025). Our setting is complementary: task-optimized and safety-aligned checkpoints remain fixed, and the intervention acts only on candidate selection at inference time.

Constrained and future-aware decoding. Inference-time guidance changes a generative model's selection rule without updating its parameters. Future discriminators and lookahead search evaluate downstream properties of candidate text continuations (Yang & Klein, 2021; Lu et al., 2022). Ji et al. (2025) formulate safe language generation as constrained sequential inference with learned task and safety critics, while Nguyen et al. (2026) use future-value estimates to guide particle resampling toward a sequence-level power target. In robotics, SafeDec applies temporal-logic constraints to predicted traces through hard masking and Robustness Constrained Decoding (Kapoor et al., 2025). We study the history-conditioned mass of policy-supported suffixes that achieve safe task completion after the candidate-induced transition. The deployed score is a selective heuristic approximation to that target.

Planning, viability, and runtime assurance. Model-based control and Monte Carlo planning compare actions through predicted future trajectories. Huang et al. (2025) use a teacher's Monte Carlo search trees to construct a curriculum of prefixes and estimate prefix-aware advantages for policy training. Safe reinforcement learning represents long-horizon constraints through budgets, augmented states, or constrained objectives (Sootla et al., 2022; Lin et al., 2023). Viability theory and Hamilton–Jacobi reachability characterize states from which admissible evolution remains possible (Aubin, 1991; Bansal et al., 2017), while runtime shields and predictive safety filters intervene on nominal policies to enforce specified constraints (Alshiekh et al., 2018; Hsu et al., 2023; Wabersich & Zeilinger, 2021). Our support notion is narrower by construction: it is defined under the frozen continuation law rather than over all admissible controls. Correspondingly, the deployed decoder neither reconstructs a full belief state nor claims invariant-set or closed-loop guarantees.

Evaluation of embodied safety. Safety-oriented VLA evaluation must distinguish task completion from trajectory safety. Safety-CHORES evaluates embodied tasks with explicit safety constraints (Zhang et al., 2025). Fan et al. (2026) report success-but-unsafe behavior and violation severity to expose unsafe trajectories hidden by aggregate success. A converse ambiguity arises when cumulative cost falls because a method terminates earlier or fails sooner. We therefore report success counts, cumulative cost, violation counts, and episode length jointly, and do not interpret lower unconditioned cost as dominance when completion and exposure also decrease.