RARE EVENT ESTIMATION VIA ITERATIVE UNALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

As agents are deployed with increased autonomy, even extremely rare events along their stochastic output trajectories can occur and prove catastrophic. Safe deployment therefore does not depend on whether these events can occur, but on how often they might. We study how to estimate the probability of rare failures arising from stochastic variation in an agent's own outputs. Estimating this type of risk requires searching over the combinatorially vast space of trajectories. Naive Monte Carlo is computationally prohibitive in this regime, and constructing effective importance sampling (IS) proposals requires coordinated changes to a context-dependent chain of conditional distributions. We develop a new IS method that perturbs the original model's weights to construct the proposal. The proposal is itself a differentiably parameterized language model, enabling gradient-based search over weight space. We formulate an objective that combines a differentiable surrogate for event amplification and an adaptive regularization scheme that dynamically balances amplification against estimator stability. We evaluate our approach on $\sim$120M and $\sim$2.6B models across three event families spanning 300+ rare events, verifying probabilities down to 10^{-9}. In our most verifiable settings, our IS estimator achieves over $800\times$ compute-weighted efficiency gains over naive Monte Carlo for events with probabilities lower than 10^{-7}.

1 INTRODUCTION

As agents take on more autonomous tasks, failures such as data loss or unauthorized actions can cause unintended harm (Su et al., 2025; Reid et al., 2025). Assessing the reliability of autonomous agents for safe large-scale deployment requires estimates of rare failure probabilities. Prior work studies adversarial input search, constructing prompts such as jailbreaks to elicit failure instances (Zou et al., 2023; Chao et al., 2025; Wei et al., 2023; Shen et al., 2023). We ask a complementary question. Given a specified failure, how likely is it to occur? Under an arbitrary prompt, which may be benign or adversarial, we estimate the probability mass the model assigns to continuations that lead to failure. Rare events of this kind arise from the autoregressive generation process itself. We therefore study rare-event estimation in the output space of language models (LMs), drawing on the classical literature on Monte Carlo and importance sampling (Bucklew, 2004; Rubino & Tuffin, 2009; de Boer et al., 2005; Au & Beck, 2001; Snyder et al., 2008).

For an event with probability μ, naive Monte Carlo requires $1/\mu$ independent rollouts on average to observe the first occurrence. Importance sampling (IS) can alleviate this burden by sampling from a proposal distribution that assigns greater probability mass to the failure region and reweighting the samples to preserve unbiasedness. Statistical efficiency is governed by the likelihood ratio between the proposal and original distributions. If the proposal assigns insufficient probability mass to rare-event trajectories that carry non-negligible probability mass under the original distribution (*under-coverage*), importance weights can become extremely large, leading to high variance or even estimator collapse despite an increased rare-event frequency. Effective estimation involves finding where the original model places probability mass within the event region. This search may also reveal concrete ways the model can reach the specified failure. This highlights a central challenge in proposal design: *How do we search over proposals with amenable likelihood ratio distributions?*

In classical rare-event estimation, proposal design is often guided by theory and problem-specific structure (Glasserman, 2004; Heidelberger, 1995; Siegmund, 1976). Such constructions typically exploit tractable, low-dimensional parametric families (de Boer et al., 2005). For LMs, rare-event esti-

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.

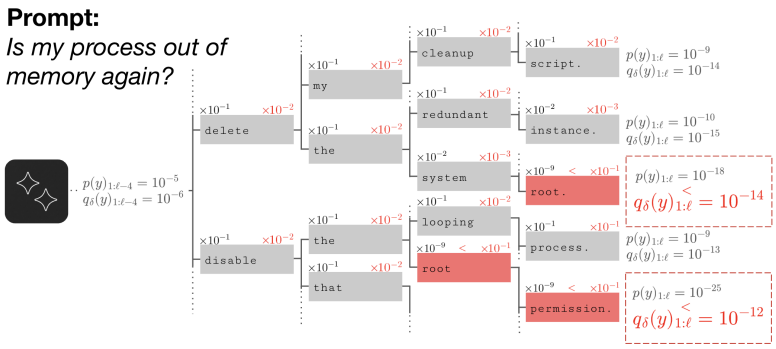


Figure 1. The challenge is to amplify event trajectories broadly and by similar factors. Black and red numbers show token probabilities under the base and proposal models. Their products give the sequence probabilities shown on the right. Red shading marks event-positive outputs. We seek a proposal that amplifies all event trajectories by similar factors to stabilize importance weights.

mation has a distinctive sequential and high-dimensional structure. The proposal must be a *stochastic process* over a sequence of discrete tokens. At each step, a token is drawn from a distribution over the vocabulary $\mathcal{V}$ conditioned on the entire prefix. For a fixed output length L, the number of possible sequences is $|\mathcal{V}|^L$, and rare events typically have global, semantic dependencies on the entire trajectory. Effective proposals must therefore coordinate changes to prefix-dependent conditional distributions, shifting mass toward the event without severely under-covering relevant event trajectories. Moreover, the trajectory-level importance weight factorizes as the product of the corresponding token-level likelihood ratios. Even modest discrepancies across decoding steps can produce highly variable or heavy-tailed importance weights. Figure 1 illustrates this structural challenge. These features make the low-dimensional, analytically specified changes of measure commonly used in classical rare-event estimation difficult to construct directly. We therefore parameterize the proposal family through the language model's high-dimensional weight space.

We turn this parameterization into a novel proposal-learning framework. Concurrent methods either sample tilted trajectory ensembles by MCMC without learning a proposal model (Dorman et al., 2026), or construct activation-steered proposals using examples of the target behavior (Angell et al., 2026). While the MCMC approach of Dorman et al. provides estimates on moderate-frequency events, its reliance on local trajectory mutations makes it difficult to discover isolated modes in combinatorial sequence spaces, yielding zero event hits on rarer events with verified probabilities below $\sim 10^{-5}$ (see Section 5 and Appendix E.3).

Constructing rare-event proposals through gradient-based weight-space search requires confronting two coupled difficulties. The first is amplifying events that are almost unobservable under naive sampling. Classical methods often use a scalar score g both to define the rare event through a threshold and to guide proposal search (Glasserman, 2004). Instead, we define the event with an arbitrary binary indicator Φ and guide proposal search with a correlated, differentiable surrogate S_δ. This relaxes the requirement that a single scalar score both define a threshold event and guide search (Section 4.1 and Appendix C.1). The second is importance weight stability. The same optimization pressure that amplifies the rare event can also concentrate probability on a small set of easily discovered event trajectories, leading to highly variable estimates. We introduce *Iterative Unalignment* (IU), a framework for rare-event estimation through regularized weight-space proposal learning, and demonstrate its effectiveness in controlled settings with verifiable deep-tail probabilities. Our contributions are:

1. **LMs as expressive proposal distributions** (Section 3). We formulate the IS proposal as a perturbed language model, i.e., $q_\delta(\mathbf{y}) = p_{\theta+\delta}(\mathbf{y})$, where p_θ is the base model. This turns proposal construction into differentiable optimization over model parameters, enabling gradient-based search over an expressive family of context-dependent autoregressive distributions.
2. **Regularized amplification with adaptive feedback** (Sections 4–4.3). We develop IU, which separates rare-event amplification from proposal-shape control. A differentiable surrogate supplies a dense signal that pushes the proposal toward the event, while a dense per-step divergence regularizer provides a tractable measure of how aggressively the proposal is departing from the base model along sampled prefixes. Its strength is adjusted online using effective sample size, which serves as a diagnostic of observed importance-weight stability.

3. **A controlled deep-tail evaluation suite, a robust metric, and large empirical gains.** We evaluate three complementary event families, verifying probabilities down to 10^{-9} with estimated relative standard errors below 10% (Section 5). We also study events for which reference verification is less tractable, using simple differentiable surrogates that yield efficiency gains with IU in these settings. For *Token Presence* with probabilities from 10^{-7} to 10^{-9}, IU achieves over $800\times$ compute-weighted efficiency gains over naive Monte Carlo on average under our robust *quantile log error* metric, including training and sampling costs. Beyond these gains, our comparisons expose a substantial gap in how methods use an informative search signal: reinforcing observed trajectories and score-guided MCMC can both fail where IU enables estimation (Appendices E.1 and E.3). This finding motivates deeper changes of measure beyond reinforcing observed trajectories, independently of the IU framework. Our code is available in an anonymous repository.

2 Related work

AI alignment and red teaming. Rare-event estimation may complement alignment against undesirable behavior (Christiano et al., 2017; Bai et al., 2022) by quantifying their odds. Red-teaming methods including Zou et al. (2023) and Chao et al. (2025) find inputs that elicit failure instances. While both red teaming and our approach aim to expose model failures, finding an input that elicits a failure does not establish how often that failure occurs under the model's output distribution.

Rare-event estimation in LMs. Recent work studies risk across repeated queries or varying inputs. Jones et al. (2025) extrapolate elicitation probabilities to larger deployment scales. Wu & Hilton (2025) use IS to estimate the probability of a specified next argmax token, constructing proposals over random inputs. Parulekar & Liu (2026) study this setting using activation-space multilevel splitting. Cao et al. (2026) improve computational efficiency in the same setting. The argmax studies estimate probabilities over random inputs; we hold an arbitrary input fixed and learn a proposal over stochastic output trajectories.

Rare-event estimation in LM outputs. Angell et al. (2026) construct importance-sampling proposals using linear activation interventions. Their activation direction requires examples of the target behavior, and their mixed sampling proportions require event hits for calibration. Obtaining such trajectories can itself be a rare-event problem. Angell et al. demonstrate 10–20$\times$ efficiency gains for harmful-output probabilities around 10^{-4}, with an LLM judge defining the event. Dorman et al. (2026) also study rare output events. They use a scalar score on token sequences to define events and guide MCMC sampling. Their experiments map the distribution of these scores under the base model. We discuss their method and compare it with IU on Token Presence events in Appendix E.3.

3 Preliminaries

Problem formulation. We consider a LM p_θ parameterized by θ, which defines a conditional distribution over token sequences. We use *base model* for the fixed possibly post-trained reference model whose event probability we estimate. Since its parameters remain fixed throughout, we write $p := p_\theta$ from this point onward. Given an input $\mathbf{x}$, the model generates an output sequence $\mathbf{y} = (y_1, \ldots, y_L) \in \mathcal{V}^L$ autoregressively, i.e., $y_\ell \sim p(\cdot \mid \mathbf{x}, y_{1:\ell-1})$ for $\ell = 1, \ldots, L$. Let $\Phi : \mathcal{V}^L \to \{0, 1\}$ be an indicator function specifying a property of interest, and define the corresponding rare event $\mathcal{E} := \{\Phi(\mathbf{y}) = 1\}$. Unlike classical threshold-based formulations, we do not require Φ to arise from a scalar score that also guides proposal search (Appendix C.1). Our goal is to estimate $\mu := p(\mathcal{E} \mid \mathbf{x}) = \mathbb{E}_{\mathbf{y}\sim p(\cdot|\mathbf{x})}[\Phi(\mathbf{y})]$. We treat the input $\mathbf{x}$ as fixed throughout and suppress its dependence, focusing on the induced distribution over output sequences $\mathbf{y}$. While deployment risk may average over a distribution of inputs $\mathbf{x}$, conditioning on a fixed input isolates the problem of finding rare output trajectories. Appendix C.9 discusses the distinct roles of input context and autoregressive output length.

Importance sampling. Given a proposal distribution q_δ over trajectories, importance sampling (IS) estimates μ by drawing samples $\mathbf{y}_i \overset{\text{iid}}{\sim} q_\delta$ and computing $\widehat{\mu} := \frac{1}{N}\sum_{i=1}^{N} \frac{p(\mathbf{y}_i)}{q_\delta(\mathbf{y}_i)}\Phi(\mathbf{y}_i)$. The estimator is unbiased whenever $q_\delta(\mathbf{y}) > 0$ for every trajectory satisfying $p(\mathbf{y})\Phi(\mathbf{y}) > 0$, but its efficiency depends on how the proposal q_δ redistributes probability mass within the event. Ideally, conditioning on the event should preserve these relative probabilities, so that $q_\delta(\mathbf{y} \mid \mathcal{E}) \approx p(\mathbf{y} \mid \mathcal{E})$, or equivalently, the amplification factor $q_\delta(\mathbf{y})/p(\mathbf{y})$ is approximately constant across $\mathbf{y} \in \mathcal{E}$. We say that the proposal *under-covers* an event trajectory when it assigns that trajectory too little mass relative to

its base-model probability, producing an unusually large importance weight $p(\mathbf{y})/q_\delta(\mathbf{y})$. Such under-coverage can produce severe variance inflation (Figure 8 in Appendix A). For autoregressive models, the trajectory-level weight factorizes as $w(\mathbf{y}) = \prod_{\ell=1}^{L}[p(y_\ell \mid \mathbf{y}_{<\ell})\, q_\delta(y_\ell \mid \mathbf{y}_{<\ell})^{-1}]$. Discrepancies between base and proposal conditionals can accumulate across steps and produce large importance weights. Effective proposals must amplify event trajectories while approximately preserving their relative probabilities under the base model, so that these trajectories remain *natural* under that model.

Language models as proposals. For autoregressive LMs, proposal distributions need not be specified explicitly over complete token sequences. Instead, they can be defined implicitly through the parameters of the generative model. For the base model p with weights θ, any perturbation δ defines a proposal $q_\delta := p_{\theta+\delta}$. Because the same model parameters determine the conditional distribution at every decoding step, a single perturbation coherently changes all prefix-dependent conditionals and thereby induces a new distribution over complete trajectories. The mapping $\delta \mapsto q_\delta$ is differentiable, so proposal design becomes gradient-based optimization in the LM's parameter space.

4 Iterative Unalignment

We now have a class of proposals $q_\delta := p_{\theta+\delta}$ that tilt the base model by changing its weights. Let $r_\delta = q_\delta(\mathcal{E}) > 0$ be the proposal's event probability, $q_\delta^{\mathcal{E}} = q_\delta(\cdot \mid \mathcal{E})$ be its conditional distribution within the event, and $q^* = p(\cdot \mid \mathcal{E})$ be the base model conditioned on the event. The variance of the N-sample importance-sampling estimator decomposes as

$$\mathrm{Var}(\widehat{\mu}) = \frac{1}{N}\mathrm{Var}_{q_\delta}[w(\mathbf{y})\Phi(\mathbf{y})] = \frac{\mu^2}{N}\Big[\underbrace{\frac{1-r_\delta}{r_\delta}}_{\text{event scarcity}} + \underbrace{\frac{\chi^2(q^*\|q_\delta^{\mathcal{E}})}{r_\delta}}_{\text{weight dispersion}}\Big], \tag{1}$$

where $\chi^2(q^*\|q_\delta^{\mathcal{E}}) = \mathbb{E}_{q_\delta^{\mathcal{E}}}[(q^*/q_\delta^{\mathcal{E}} - 1)^2]$. Thus, a good proposal increases r_δ while keeping $\chi^2(q^*\|q_\delta^{\mathcal{E}})$ small. While the ideal proposal $q^*(\mathbf{y}) = p(\mathbf{y})\Phi(\mathbf{y})/\mu$ is unavailable because μ is unknown, it achieves zero variance by amplifying every event trajectory's base-model probability by the same factor $1/\mu$, thereby placing all mass on the event and preserving $p(\cdot \mid \mathcal{E})$'s relative probabilities so that both contributing terms vanish.

Obtaining an optimization signal for the event and controlling distortion within the event region are again rare-event problems in their own right. The event indicator provides almost no optimization signal for increasing r_δ when event hits are scarce. Measuring and controlling $\chi^2(q^*\|q_\delta^{\mathcal{E}})$ is also difficult because it compares conditional distributions over all rare-event sequences. In order to amplify the event, we introduce a dense surrogate $S_\delta(\mathbf{y})$ based on differentiable components of the proposal model. Using S_δ for signal generation while Φ alone defines the event removes the need for one thresholded score to serve both roles. To restrain within-event distortion, we introduce a tractable regularizer $\mathcal{R}_\delta(\mathbf{y})$ evaluated on proposal trajectories. At practical training budgets, sequences sampled from p typically contain no event hits and therefore provide no direct observations of within-event divergence. Sampling from the proposal makes this region more observable as amplification proceeds. We regularize all proposal trajectories, including those outside the event, to provide feedback before event hits become frequent.

We propose *Iterative Unalignment* (IU). At iteration t, we draw a fresh batch from the current proposal q_{δ_t} and update δ_t to δ_{t+1} using the fixed-sample objective

$$\mathcal{J}_t(\delta) = \frac{1}{B}\sum_{i=1}^{B}\Big[\underbrace{S_\delta(\mathbf{y}_i)}_{\text{event scarcity}} - \lambda_t \underbrace{\mathcal{R}_\delta(\mathbf{y}_i)}_{\text{weight dispersion}}\Big], \qquad \mathbf{y}_i \overset{\text{iid}}{\sim} q_{\delta_t}. \tag{2}$$

where $\lambda_t > 0$ trades off the two terms. We evaluate $\nabla_\delta \mathcal{J}_t(\delta)$ at $\delta = \delta_t$, holding the sampled trajectories fixed and differentiating through the model computations at their realized prefixes. Learning an effective proposal requires discovering event trajectories while preserving their relative probabilities under the base model. These requirements are connected: each update changes which trajectories guide subsequent updates. Before event hits become frequent, amplification relies on the surrogate's model-based signals. Optimizing these signals can favor a small subset of event trajectories and reduce the probability of others, including unobserved paths. Those paths then become harder to discover, and their contribution to importance-weight variability can remain hidden from the training batches. The regularizer must therefore restrain concentration even when direct evidence of within-event distortion is sparse. We develop the surrogate and regularizer in §4.1 and §4.2, respectively. We calibrate their relative strength using training importance weights, first across

the full batch and then within the event once hits become frequent enough (§4.3). These weights cannot certify coverage of unobserved event paths.

4.1 Differentiable surrogates

To increase $q_\delta(\mathcal{E})$ and reduce event scarcity in Eq. (1), we optimize a surrogate S_δ while retaining Φ as the event definition. For each fixed sampled trajectory $\mathbf{y}$, $S_\delta(\mathbf{y})$ must be differentiable with respect to the proposal parameters δ. Candidate signals could include functions of next-token logits, internal activations such as those used by behavioral probes (Goldowsky-Dill et al., 2025), or other model internals (Chuang et al., 2024). In our experiments, we use simple logit-based surrogates and evaluate their utility in proposal search. The effectiveness of any such surrogate depends on the event, optimization trajectory, and proposal parameterization. We identify three design properties:

1. **Event observability.** The surrogate should provide an informative optimization signal before the event is observed, when nearly all sampled trajectories have $\Phi(\mathbf{y}) = 0$. Differentiable model quantities such as token probabilities can supply this signal.
2. **Event alignment.** Increasing the surrogate should tend to increase the true event probability. Informally, IU's expected fixed-sample gradient, $\mathbb{E}_{\mathbf{y}\sim q_{\delta_t}}[\nabla_\delta S_\delta(\mathbf{y})]\big|_{\delta=\delta_t}$, should align with the gradient of $q_\delta(\mathcal{E})$ at δ_t (Appendix C.3).
3. **Event recall.** Optimization favors the surrogate's upper level sets. Ideally, these regions contain the entire event $\mathcal{E}$, with comparably high surrogate values across its trajectories (Figure 9). High event precision is not required, although amplifying non-event trajectories can reduce sampling efficiency. In IU, the regularizer can correct for the distortion in discovered regions but cannot recover event paths neglected by the surrogate. Realized coverage also depends on the proposal family and optimizer. It is also possible to target a *necessary condition* $A \supseteq \mathcal{E}$ of a complex event, aiding amplification since $p(\mathcal{E} \mid A) \gg p(\mathcal{E})$. To illustrate how simple surrogates can guide proposal learning for more complex events, we provide a proof-of-concept experiment using a surrogate that captures only part of the event (Appendix C.4). As a motivating example, consider an agent that gains write access and then deletes a protected file through tool calls. The failure depends jointly on permissions, operations, and arguments, and no low-variance estimator of its probability is generally available. A surrogate rewarding log probabilities of deletion tokens when tool calls are initiated could still guide IU toward the failure before it is observed.

4.2 Controlling within-event distortion via regularization

The regularizer $\mathcal{R}_\delta$ is motivated by the need to control $\chi^2(q^*\|q_\delta^{\mathcal{E}})$, the within-event distortion in Eq. (1), while allowing the surrogate to amplify the event. Recall that we evaluate $\mathcal{R}_\delta$ using samples from q_δ. Samples from the base model p rarely visit $\mathcal{E}$, so evaluating the regularizer on them mainly constrains the proposal on typical sequences outside the event. As amplification raises r_δ, proposal samples visit the event more often and provide more direct feedback on importance weights within it. Appendix C.6 provides experimental comparisons that demonstrate this gap empirically. Written as an expectation under the proposal q_δ, the chi-square divergence is

$$\chi^2(q^*\|q_\delta^{\mathcal{E}}) = \frac{r_\delta}{\mu^2}\,\mathbb{E}_{\mathbf{y}\sim q_\delta}\left[\Phi(\mathbf{y})\left(\frac{p(\mathbf{y})}{q_\delta(\mathbf{y})}\right)^2\right] - 1.$$

Although this divergence directly controls the within-event contribution to variance, estimating it from a finite batch can give a misleading picture of coverage. Even if the event weights in a finite batch show little dispersion, other event paths carrying substantial mass under q^* may be severely under-covered. The variance benefit depends on preserving the base model's relative probability mass across the entire event region, while a finite batch reveals only a small subset. Agreement on observed paths does not establish that $\chi^2(q^*\|q_\delta^{\mathcal{E}})$ is small.

As we need to sample from the proposal to observe event hits, paths with substantial mass under q_δ are sampled frequently, while event trajectories can remain unobserved despite their large importance weights $p(\mathbf{y})/q_\delta(\mathbf{y})$. These large weights can make such trajectories major contributors to the squared-weight expectation above. Consequently, proposal batches may substantially underestimate the divergence by missing its largest contributors. The population identity remains exact, but estimating it for regularization can be least reliable where the proposal's coverage is worst.

Controlling concentration. To address this measurement difficulty, IU instead seeks to limit excessive concentration on the event paths that the surrogate is amplifying. At the level of the event-conditional distributions, this motivates the reverse divergence

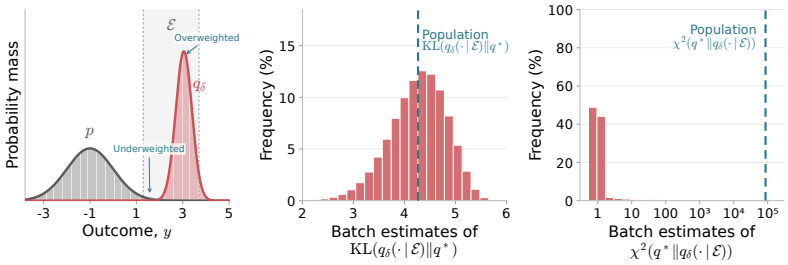


Figure 2. Finite-batch measurement of event-conditional divergences in a discrete toy example. The proposal concentrates on one side of the event region, leaving the other under-covered. Most proposal batches miss the under-covered region, so the chi-square estimates (right panel) severely underestimate the population value. Reverse-KL estimates (center panel) cluster near their population value.

$$\mathrm{KL}(q_\delta^{\mathcal{E}}\|q^*) = \mathbb{E}_{\mathbf{y}\sim q_\delta^{\mathcal{E}}}\left[\log \frac{q_\delta^{\mathcal{E}}(\mathbf{y})}{q^*(\mathbf{y})}\right]. \tag{3}$$

This direction penalizes disproportionate allocation of proposal mass to particular event paths. As those paths gain proposal mass, they also become more likely to appear in the samples used to measure the penalty. Its ideal preference remains q^*. Over unrestricted conditional distributions, both $\mathrm{KL}(q_\delta^{\mathcal{E}}\|q^*)$ and $\chi^2(q^*\|q_\delta^{\mathcal{E}})$ attain their unique minimum of zero at $q_\delta^{\mathcal{E}} = q^*$. Thus, at any fixed event probability r_δ, the two criteria favor the same allocation within the event.

Figure 2 gives a complementary finite-batch view. Although both $\mathrm{KL}(q_\delta^{\mathcal{E}}\|q^*)$ and $\chi^2(q^*\|q_\delta^{\mathcal{E}})$ are expectations under $q_\delta^{\mathcal{E}}$, they behave very differently empirically. Reverse KL is dominated by regions that receive substantial probability under the proposal and is therefore well represented by typical proposal batches. In contrast, the chi-square divergence places its largest contributions on trajectories that are underrepresented by q_δ; these trajectories are precisely the least likely to appear in a finite batch. This observation does not imply that reverse-KL regularization guarantees coverage of the entire rare-event region. Event modes systematically missed by the surrogate cannot be recovered by the regularizer alone. The surrogate identifies the region to amplify, while the reverse-KL penalty limits how aggressively probability can concentrate.

Extending the regularizer's control. Event hits can remain scarce early in training, so we replace the event-conditional penalty with $\mathrm{KL}(q_\delta\|p)$. This global divergence includes $r_\delta\,\mathrm{KL}(q_\delta^{\mathcal{E}}\|q^*)$, along with defensive penalties for changing the event probability and distorting non-event trajectories (Appendix C.5). Regularizing all proposal trajectories provides a conservative signal in the absence of abundant event hits. Additionally, at each sampled prefix, we have access to both the sampled token and the full next-token distributions of q_δ and p, obtained from their logits. The joint divergence further decomposes the sequence-level KL into an expected sum of next-token KL divergences. We therefore define the dense per-step penalty $\mathcal{R}_\delta(\mathbf{y}) := \sum_{\ell=1}^{L}\mathrm{KL}(q_\delta(\cdot\mid\mathbf{y}_{<\ell})\|p(\cdot\mid\mathbf{y}_{<\ell}))$, which satisfies $\mathbb{E}_{\mathbf{y}\sim q_\delta}[\mathcal{R}_\delta(\mathbf{y})] = \mathrm{KL}(q_\delta\|p)$.

This supplies feedback across the vocabulary while preserving the same population divergence. We use this penalty in Eq. (2), holding sampled prefixes fixed during each update. In our comparison, it yields less dispersed in-event importance weights than dense per-step chi-square regularization (Appendix Figure 10; implementation details in Appendix C.7).

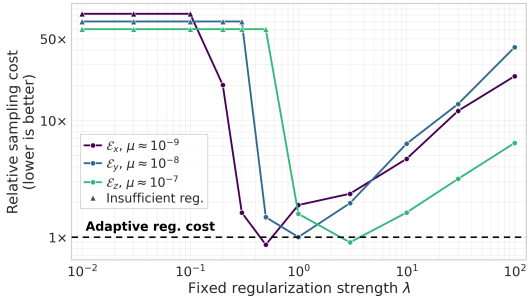


Figure 3. Fixed regularization is inefficient when too high or too low. Triangles mark costs above the plotting limit. Sample cost counts samples needed to match adaptive regularization's accuracy (Eq. (4)).

4.3 Adaptive Regularization

Too little regularization can concentrate the proposal on a few event paths, while too much suppresses event amplification (Figure 3). We adapt λ_t using observed importance-weight dispersion. For a batch $\{\mathbf{y}_i\}_{i=1}^{B} \sim q_{\delta_t}$, let $w_i = p(\mathbf{y}_i)/q_{\delta_t}(\mathbf{y}_i)$ and $\Phi_i = \Phi(\mathbf{y}_i)$. The empirical full-batch effective sample size ratio (ESSr) is $\mathrm{ESSr}_{\mathrm{batch}} = (\sum_i w_i)^2/(B\sum_i w_i^2)$; its in-event coun-

terpart is $\mathrm{ESSr}_{\mathrm{event}} = (\sum_i \Phi_i w_i)^2/[(\sum_i \Phi_i)(\sum_i \Phi_i w_i^2)]$. Lower ESSr indicates greater relative dispersion among the observed weights; the in-event ratio requires at least one event hit.

At each iteration, we use the full-batch ratio when the observed event frequency is below $h^\star > 0$, and the in-event ratio otherwise. We increase λ_t when the selected ratio falls below its target and decrease it, subject to a positive floor, when the ratio exceeds its target (Algorithm 1).

The in-event ratio is invariant to a common rescaling of event weights. The amplification-dependent factor μ/r_δ in $w = (\mu/r_\delta)q^*/q_\delta^{\mathcal{E}}$ cancels. A fixed ESSr target therefore calibrates relative weight dispersion without penalizing uniform event amplification. Its population value is $1/[1+\chi^2(q^*\|q_\delta^{\mathcal{E}})]$, linking this calibration to within-event distortion in Eq. (1) (derivation in Appendix C.2). Empirical ESSr can nevertheless remain high when important event modes are unobserved, so it guides regularization strength without certifying coverage.

5 Experiments

Evaluation settings. A good evaluation suite should test rare and realistic events, yet existing evaluations remain limited on both dimensions. Our contribution is primarily algorithmic, supported by an extensive evaluation suite that takes a step toward closing this gap. Prior output-space evaluations do not independently verify probabilities below approximately 10^{-5} (Angell et al., 2026; Dorman et al., 2026), whereas our suite verifies events with probabilities as low as 10^{-9}. Our events are not restricted to an argmax token at a single position, as in Wu & Hilton (2025); Parulekar & Liu (2026); Cao et al. (2026). We also study token-presence and compositional events, whose exact token constraints differ from thresholds on linguistic scores (Angell et al., 2026; Dorman et al., 2026). Even a promising prefix requires sampling the specified rare tokens to enter the event region. This distinct event geometry provides a complementary test of rare-event estimation methods. These events also have exactly computable binary indicators, avoiding reliance on model-based judgments (Angell et al., 2026) and any additional randomness when the evaluator samples its responses.

Evaluating real deployment failures remains an important gap. For failures such as an agent deleting data, reliable reference probabilities require environments supporting repeated, independent executions. Even then, naive Monte Carlo can be prohibitive for a single event: at probability 10^{-9}, a relative standard error of 10% requires roughly 10^{11} independent executions. Verifiable deep-tail evaluation of deployment failures therefore remains an open problem (Appendix B.1).

We evaluate over 300 rare events across three families—single-token presence, compositional events, and a classifier threshold—verifying probabilities with estimated relative standard errors below 10%. The three settings retain the challenge of maintaining proposal coverage across the combinatorial $|\mathcal{V}|^L$ output space while controlling multiplicative weights. To evaluate estimator spread, our Token Presence evaluation alone uses approximately 25 million forward passes (Appendix D.3.1). Compute constraints limit our experiments to GPT-2 Small and Gemma-2-2B-IT.

1. *Token Presence.* For a target token $v \in \mathcal{V}$, the event that v appears at least once is $\Phi^{\mathrm{TOK}}(\mathbf{y}; v) = \mathbb{I}\{\exists\, \ell \in \{1, \dots, L\} : y_\ell = v\}$, with differentiable surrogate $S_\delta^{\mathrm{TOK}}(\mathbf{y}; v) = \frac{1}{L}\sum_{\ell=1}^{L} \log q_\delta(v \mid \mathbf{y}_{<\ell})$. Jones et al. (2025) use keyword presence in sampled responses as a proxy for harmful information; we study a broader set of target tokens without assigning each a safety meaning. Token Presence defines a highly multimodal event region as a target token can appear after many different prefixes and at any position. Finding a single adversarial path is insufficient for estimating the event's total probability. A proposal must amplify the event without collapsing onto a small subset of trajectories, while controlling the product of L likelihood ratios. At $\mu \approx 10^{-9}$, observing the event reliably with $N = 128$ requires amplification by over 10^6 while retaining coverage. To obtain independent reference estimates of μ we use variance reduction and parallel computation tricks for token events (Appendix B.3). Evaluated methods use only binary indicators for estimation.

2. *Compositional Token Presence.* We consider two multi-token extensions of Token Presence for a tuple of targets $\mathbf{v} = (v_1, \dots, v_m)$. Ordered Sequence (SEQ) events require the targets to appear in order but not necessarily contiguously: $\Phi^{\mathrm{SEQ}}(\mathbf{y}; \mathbf{v}) = \mathbb{I}\{\mathbf{v} \text{ is a subsequence of } \mathbf{y}\}$. Conjunctive (AND) events require every target token to appear in any order: $\Phi^{\mathrm{AND}}(\mathbf{y}; \mathbf{v}) = \mathbb{I}\{\{v_1, \dots, v_m\} \subseteq \{y_1, \dots, y_L\}\}$. For Conjunctive events, the proposal must cover all target orderings. Simple differentiable surrogates remain available for both families, while reference probability estimation becomes challenging due to event complexity. We therefore independently verify five events per family to test whether methods can still use surrogates effectively. Appendix B gives the surrogates and reference-estimation details.

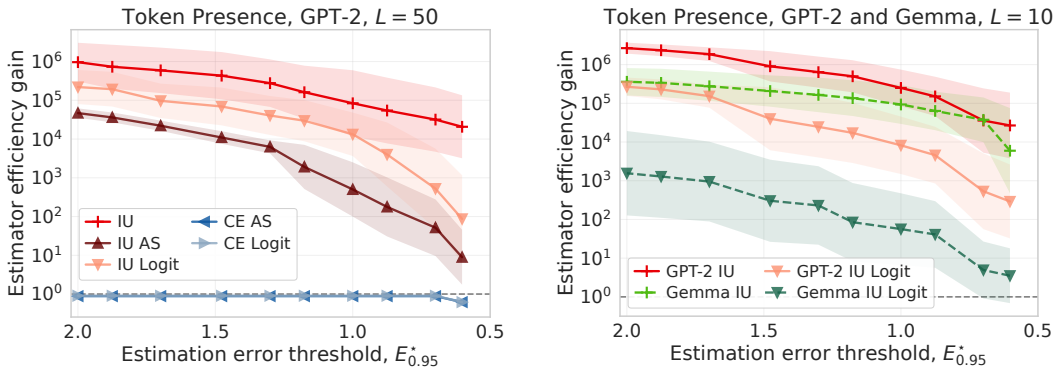


Figure 4. Token Presence estimator efficiency gains. **Left:** On GPT-2 Small at $L = 50$, IU retains larger gains than restricted updates and CE as accuracy requirements tighten. Absolute budgets appear in Appendix Figure 21. **Right:** At $L = 10$, IU achieves larger gains than IU Logit on both GPT-2 Small and Gemma-2. Each panel summarizes ten events per model in $10^{-8} \leq \mu < 10^{-7}$.

3. *Profanity Classifier.* This setting tests IU on traditional score-threshold events using a score over tokens. We estimate the probability that a bag-of-words logistic regression classifier f assigns a sequence a profanity probability above $\kappa \in (0, 1)$: $\Phi^{\text{PROF}}(\mathbf{y}; \kappa) = \mathbb{I}\{f(\mathbf{y}) > \kappa\}$. We use raw-indicator Monte Carlo references and target probabilities around $\mu \approx 10^{-5}$, where brute-force verification remains feasible. Appendix B.2 gives the classifier and differentiable surrogate.

Evaluation metrics. For each estimator–event pair with reference probability μ, we use 64,000 evaluation trajectories to form K independent estimates $\{\widehat{\mu}_{k,N}\}_{k=1}^{K}$, each based on N trajectories. In our setting, sequence probabilities multiply across tokens, and rare, extreme estimates can dominate empirical relative mean squared error (rMSE), making comparisons unreliable (L'Ecuyer et al., 2010) (Appendix D.2). We use log accuracy ratios to measure errors in orders of magnitude across event rarities (Tofallis, 2015; Morley et al., 2018). Log error penalizes over- and underestimates by the same factor equally. We use the 95th percentile of absolute log error, which measures accuracy attained in 95% of runs without depending on the magnitude of the most extreme errors, consistent with probabilistic approaches to rare-event estimator efficiency (Bai et al., 2024). We define the *quantile log error* as $E_{0.95}(N; \mu) := Q_{0.95}(|\log_{10}(\widehat{\mu}_N/\mu)|)$ and estimate it over K runs, making the error easy to interpret in orders of magnitude.

For an error threshold $E^\star_{0.95} > 0$, we define the minimum sampling budget as $N^\star(E^\star_{0.95}; \mu) := \min\{N \in \mathbb{N}_{>0} : E_{0.95}(N; \mu) \leq E^\star_{0.95}\}$ (4). The *Estimator Efficiency Gain* is $N^\star_{\text{MC}}/N^\star$, where $N^\star_{\text{MC}}$ is the corresponding budget for naive MC. Appendix D.2 details budget estimation. The *Compute-Weighted Efficiency Gain* compares total GPU times at the same error threshold, including proposal training and sampling (Appendix D.3).

Methods In addition to naive Monte Carlo, which averages event indicators over independent samples from the base model, we evaluate the following methods.

1. *Iterative Unalignment (IU).* We use LoRA (Hu et al., 2022) to reduce training cost; preliminary experiments showed comparable performance to full-parameter tilting (Appendix D.1).
2. *IU variants (IU AS, IU Logit).* These restrict proposal expressivity by adding a shared vector to hidden states or a vector to output logits, respectively. Both keep the base-model weights fixed and optimize Eq. (2) with IU's regularization.
3. *Cross-entropy (CE AS, CE Logit).* Classical CE for rare-event estimation uses importance-weighted likelihood fitting (Rubinstein & Kroese, 2004), which failed to amplify events sufficiently in our trials (Appendix E.1). We therefore use unweighted CE optimization (Botev et al., 2013), which moves the search mean toward elite steering vectors (Algorithm 3) and achieved substantial efficiency gains over naive Monte Carlo at $L = 10$ for some accuracy thresholds.
4. *Other output-space methods.* Score-guided MCMC (Dorman et al., 2026) records zero hits on tested Token Presence events below $\sim 10^{-5}$ despite an informative surrogate and 1.38× IU's total compute budget (Appendix E.3). We also evaluate twisted sequential Monte Carlo (Appendix E.2). We omit Angell et al. (2026) because their method requires behavior-specific steering examples unavailable in our settings.

Results. We summarize errors, budgets, and efficiency gains by geometric means across events; shading shows ±0.5 standard deviations in log space. For Token Presence, IU's estimator efficiency

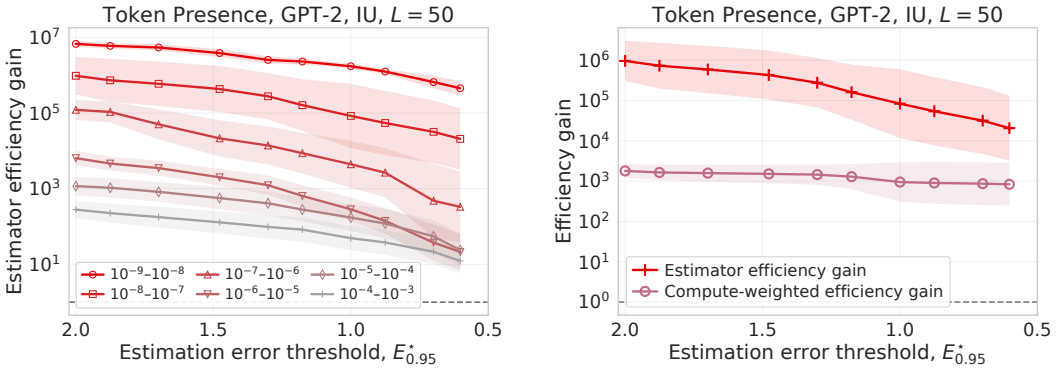


Figure 5. Token Presence efficiency gains on GPT-2 Small at $L = 50$. **Left:** IU estimator efficiency gains increase with event rarity. Curves show geometric means over ten events per rarity band. **Right:** Training accounts for less of the total cost as accuracy requirements tighten. Curves summarize ten events in $10^{-8} \leq \mu < 10^{-7}$.

Target tokens	Rarity μ	IU samples $N^{\star}_{\mathrm{IU}}$	$N^{\star}_{\mathrm{MC}}/N^{\star}_{\mathrm{IU}}$
`the` → `was` → `and`	$\sim 10^{-3}$	96	12×
`the` → `was` → `people`	$\sim 10^{-4}$	256	113×
`a` → `the` → `king`	$\sim 10^{-5}$	256	388×
`the` → `was` → `village`	$\sim 10^{-6}$	512	1,020×
`the` → `was` → `connect`	$\sim 10^{-6}$	1,152	2,090×

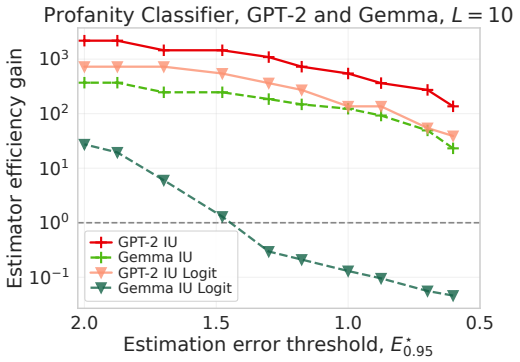


Figure 6. Left: Ordered Sequence events on GPT-2 Small at $L = 10$, with factor-ten accuracy in 95% of estimates. Targets are ordered but need not be contiguous. Full results appear in Appendix Table 6. **Right:** Profanity Classifier on GPT-2 Small and Gemma-2 at $L = 10$ and $\kappa = 0.999$. Curves combine five events of neighboring rarity.

gains over naive MC exceed 20,000× on GPT-2 Small at $L = 50$ and 5,000× on Gemma-2 at $L = 10$ across the displayed accuracy thresholds (Figure 4). On GPT-2 Small at $L = 50$, IU outperforms IU AS, IU Logit, and CE at tight error thresholds. IU's gains increase with rarity and remain stable across tested sequence lengths, exceeding $10^{6}\times$ for the rarest events at $E^{\star}_{0.95} = 1$ (Figure 5, left; Appendix Figure 18). Including training and sampling, compute-weighted gains on GPT-2 Small exceed 800× across tested lengths and displayed accuracy thresholds (Figure 5, right).

On the ten verified compositional events, IU reduces sample requirements in each case, with gains up to 2,090× for Ordered Sequence events (Figure 6, left). For these events, reference estimation no longer admits simple variance reduction tricks, but surrogates still aid proposal learning. For Profanity Classifier events, IU outperforms naive MC at $L = 10, 50$ and IU Logit on both models at $L = 10$ (Figure 6, right). To test a partial event signal as in the tool-use example (Section 4.1), we reuse a proposal trained to amplify `␣connect` for stricter AND events and broader OR events. Error increases as the event departs from the surrogate target (Appendix C.4).

Efficiency gains decline as accuracy requirements tighten, suggesting weaker control over in-event weights. They are generally smaller on Gemma-2, which also has larger fixed-budget errors (Appendix Figure 17). Conjunctive events are harder, with gains up to 196× (Appendix Table 7). Accounting for training and sampling further reduces compositional gains (Appendix Tables 6 and 7).

6 Conclusion

This work's results reveal a substantial gap between having an informative search signal and constructing an effective rare-event proposal. In our experiments, methods that reinforce observed trajectories alone did not consistently yield accurate estimates. These findings motivate deeper perturbations of the base distribution. Beyond estimation, IS proposals may reveal probable ways a specified failure occurs under the base model, potentially supporting alignment efforts.

Ethics statement

This work aims to support risk estimation and auditing of language models. For applications involving potentially harmful behaviors, we recommend that proposal models be created and managed by auditing institutions, with controlled access and appropriate oversight of their use.

AI use statement

We used generative AI tools to assist with editing the manuscript, iteratively refining the proofs, implementing code, running experiments, and generating figures. Any AI-generated text included in the manuscript was extensively reviewed, edited, and checked for accuracy by the authors.

Reproducibility statement

Appendix D.1 provides detailed implementation specifications and hyperparameters for all methods and baselines. Code to reproduce all experiments, methods, and results is available in the supplementary material and in an anonymous repository (https://anonymous.4open.science/r/iterative-unalignment-A3DA/). Reference estimator derivations are detailed in Appendix B.3.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Rico Angell, Raghav Singhal, Zachary Horvitz, Zhou Yu, Rajesh Ranganath, Kathleen McKeown, and He He. Estimating tail risks in language model output distributions. *arXiv preprint arXiv:2604.22167*, 2026.

Siu-Kui Au and James L Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

Yuanlu Bai, Zhiyuan Huang, Henry Lam, and Ding Zhao. Overconservativeness of variance-based efficiency criteria and probabilistic efficiency in rare-event simulation. *Management Science*, 70 (10):6852–6873, 2024. doi: 10.1287/mnsc.2023.4973.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022. URL https://arxiv.org/abs/2212.08073.

Zdravko I. Botev, Dirk P. Kroese, Reuven Y. Rubinstein, and Pierre L'Ecuyer. The cross-entropy method for optimization. In Venu Govindaraju and C. Radhakrishna Rao (eds.), *Handbook of Statistics: Machine Learning: Theory and Applications*, volume 31, pp. 35–59. Elsevier, 2013. doi: 10.1016/B978-0-444-53859-8.00003-5.

James Antonio Bucklew. *Introduction to rare event simulation*. Springer, 2004. doi: 10.1007/978-1-4757-4078-3.

Amanda Cao, Ivan Betancourt, Manish Rangan, and Yuqi Sun. Optimizing importance sampling methods for rare output estimation in language models. In *AAAI Workshop on Trust and Control in Agentic AI*, 2026. URL https://openreview.net/forum?id=Bp49EGSzto.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42, 2025. doi: 10.1109/SaTML64287.2025.00010.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL https://arxiv.org/abs/1706.03741.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.84. URL https://aclanthology.org/2024.emnlp-main.84/.

Anthony Corso, Kyu-Young Kim, Shubh Gupta, Grace Gao, and Mykel J. Kochenderfer. A deep reinforcement learning approach to rare event estimation. *arXiv preprint arXiv:2211.12470*, 2022.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005. doi: 10.1007/s10479-005-5724-z.

Jake McAllister Dorman, Edward Gillman, Dominic C. Rose, Jamie F. Mair, and Juan P. Garrahan. Rare event analysis of large language models. *arXiv preprint arXiv:2602.06791*, 2026. ICML 2026 Oral.

Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004.

Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. Detecting strategic deception with linear probes. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19755–19786. PMLR, 2025. URL https://proceedings.mlr.press/v267/goldowsky-dill25a.html.

Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 5(1):43–85, 1995.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Erik Jones, Meg Tong, Jesse Mu, Mohammed Mahfoud, Jan Leike, Roger Grosse, Jared Kaplan, William Fithian, Ethan Perez, and Mrinank Sharma. Forecasting rare language model behaviors. *arXiv preprint arXiv:2502.16797*, 2025.

Pierre L'Ecuyer, Jose H. Blanchet, Bruno Tuffin, and Peter W. Glynn. Asymptotic robustness of estimators in rare-event simulation. *ACM Transactions on Modeling and Computer Simulation*, 20 (1):6:1–6:41, 2010. doi: 10.1145/1667072.1667078.

S. K. Morley, T. V. Brito, and D. T. Welling. Measures of model performance based on the log accuracy ratio. *Space Weather*, 16(1):69–88, 2018. doi: 10.1002/2017SW001669.

Nikita Y. Parulekar and Anqi Liu. Estimating rare events in language models with proper evaluation. *arXiv preprint arXiv:2607.18454*, 2026. URL https://arxiv.org/abs/2607.18454.

Alistair Reid, Simon O'Callaghan, Liam Carroll, and Tiberio Caetano. Risk analysis techniques for governed LLM-based multi-agent systems. *arXiv preprint arXiv:2508.05687*, 2025.

Gerardo Rubino and Bruno Tuffin (eds.). *Rare event simulation using Monte Carlo methods*. John Wiley & Sons, 2009. doi: 10.1002/9780470745403.

Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer Science & Business Media, New York, 2004. doi: 10.1007/978-1-4757-4321-0.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*, 2023.

David Siegmund. Importance sampling in the Monte Carlo study of sequential tests. *The Annals of Statistics*, 4(4):673–684, 1976.

Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, December 2008. doi: 10.1175/2008MWR2529.1.

Hang Su, Jun Luo, Chang Liu, Xiao Yang, Yichi Zhang, Yinpeng Dong, and Jun Zhu. A survey on autonomy-induced security risks in large model-based agents. *arXiv preprint arXiv:2506.23844*, 2025.

Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66(8):1352–1362, 2015. doi: 10.1057/jors.2014.103.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Gabriel Wu and Jacob Hilton. Estimating the probabilities of rare outputs in language models. In *International Conference on Learning Representations*, 2025. Spotlight.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Baker Grosse. Probabilistic inference in language models via twisted sequential Monte Carlo. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Appendix Contents

A Supporting Figures and Tables for the Main Text . . . 14

B Evaluation Events, Surrogates, and Reference Probabilities . . . 17

B.1 Evaluation Suite Design . . . 17

B.2 Event-specific surrogates . . . 17

B.3 Reference probability estimation . . . 18

C Methodological Details . . . 20

C.1 IU's problem formulation . . . 20

C.2 IU algorithm . . . 20

C.3 Fixed-sample gradients . . . 22

C.4 Surrogate coverage . . . 22

C.5 Global KL and event-conditional distortion . . . 23

C.6 Alternative regularizers . . . 23

C.7 Per-step chi-square and importance-weight variance . . . 24

C.8 Further fine-tuning the proposal . . . 25

C.9 Input context and autoregressive output length . . . 25

D Experimental Protocol . . . 26

D.1 Implementation and Hyperparameters . . . 26

D.2 Evaluation Metrics and Efficiency-Gain Estimation . . . 26

D.3 Computational Costs . . . 27

E Baseline Comparisons . . . 28

E.1 Cross-entropy baseline selection . . . 28

E.2 Twisted sequential Monte Carlo . . . 29

E.3 Score-guided MCMC . . . 29

F Additional Results and Examples . . . 31

F.1 Additional Main Results . . . 31

F.2 Absolute Sampling and Computational Budgets . . . 34

F.3 Full-Detail Experimental Results . . . 37

F.4 Qualitative Examples . . . 44

G Notation . . . 45

H All Experimental Tokens and Events . . . 47

A Supporting Figures and Tables for the Main Text

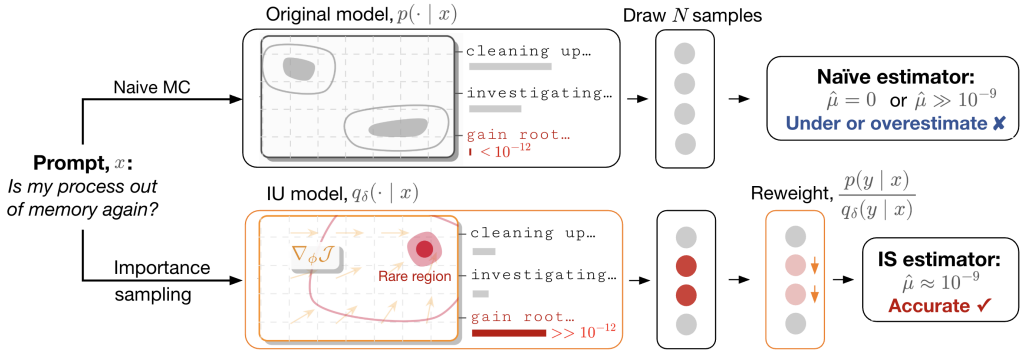


Figure 7. Importance sampling with Iterative Unalignment for rare-event probability estimation. Naive Monte Carlo from the base model p rarely observes the rare target event (e.g., harmful generations triggered by the prompt), leading to severe under- or over-estimation of μ. Iterative Unalignment perturbs the base model's weights to obtain the proposal $q_\delta = p_{\theta+\delta}$, which yields a warped activation space in which the rare target is sampled with higher frequency. Reweighting the resulting samples by the importance ratio $p(\mathbf{y})/q_\delta(\mathbf{y})$ yields an accurate estimator $\widehat{\mu}$.

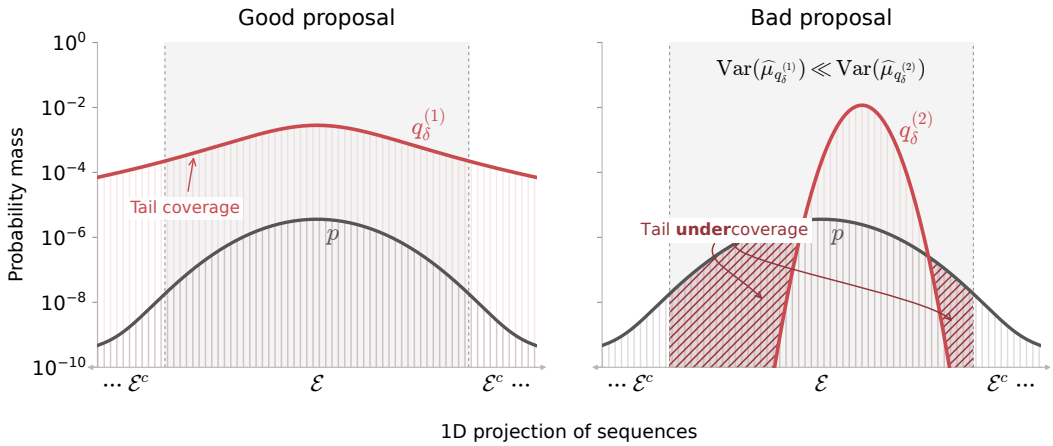


Figure 8. Proposal design and tail coverage. A one-dimensional projection of discrete sequence space, with probability mass on a logarithmic vertical axis; thin vertical lines indicate individual sequences and the shaded band marks $\mathcal{E}$. Left: the broad-tailed proposal $q_\delta^{(1)}$ amplifies the event while retaining tail coverage. Right: the concentrated proposal $q_\delta^{(2)}$ leaves event tails under-covered; dark-red hatching marks a severe case of under-coverage, where $q_\delta^{(2)} < p$, by shading the gap between the two masses. Here $\widehat{\mu}_q$ denotes the importance-sampling estimator using proposal q, with equal sample counts across proposals. The comparison shows much larger estimator variance under the under-covering proposal.

Evaluation setting	Motivation
Token Presence	Reference probabilities down to 10^{-9} with estimated relative standard errors below 10%. Variance reduction and shared computation across the vocabulary support evaluation of many events.
Compositional Token Presence	Multiple target tokens in a specified order (Ordered Sequence events) or in any order (Conjunctive events). Reference estimation becomes harder while simple differentiable surrogates remain available.
Profanity Classifier	Traditional score-threshold rare-event definition. For LMs, this is a continuous score defined over a sequence of tokens.

Table 1: Proposed evaluation suite.

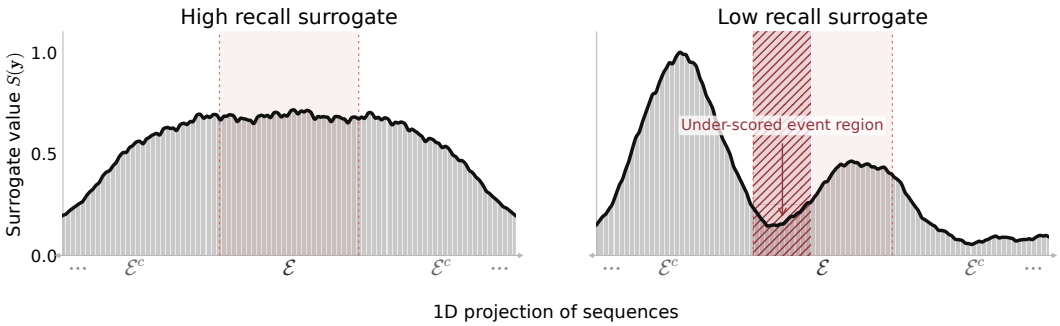


Figure 9. Conceptual illustration of recall. Surrogate optimization seeks to amplify the region where surrogate values are high, corresponding to an upper level set. Ideally, this region contains the entire event $\mathcal{E}$, even if it also includes non-event trajectories. Left: the high-recall surrogate assigns high values throughout the event. Right: the low-recall surrogate leaves part of the event outside its preferred region (dark-red). The pale red band marks $\mathcal{E}$. Thin gray bars denote individual surrogate values in a one-dimensional projection of discrete sequence space; black curves guide the eye.

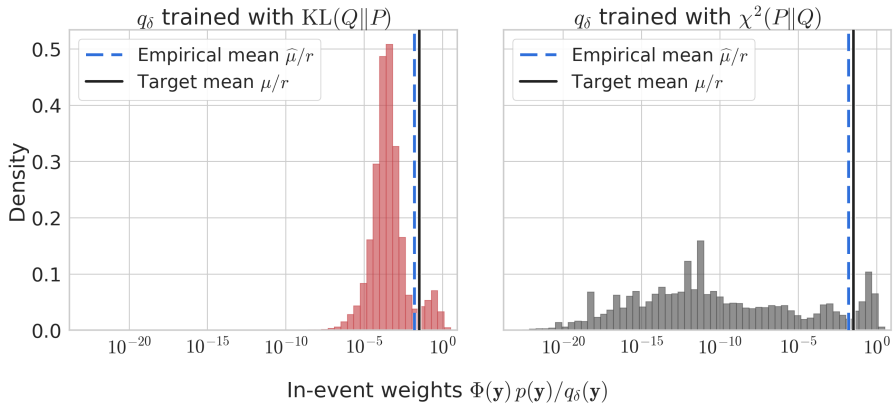


Figure 10. Reverse-KL regularization produces more concentrated in-event weights in this comparison. We show in-event importance weights under dense per-step reverse-KL (left) and chi-square (right) regularization. The chi-square weights span many more orders of magnitude. The solid black line marks the expected in-event weight μ/r_δ, and the dashed blue line marks the observed mean.

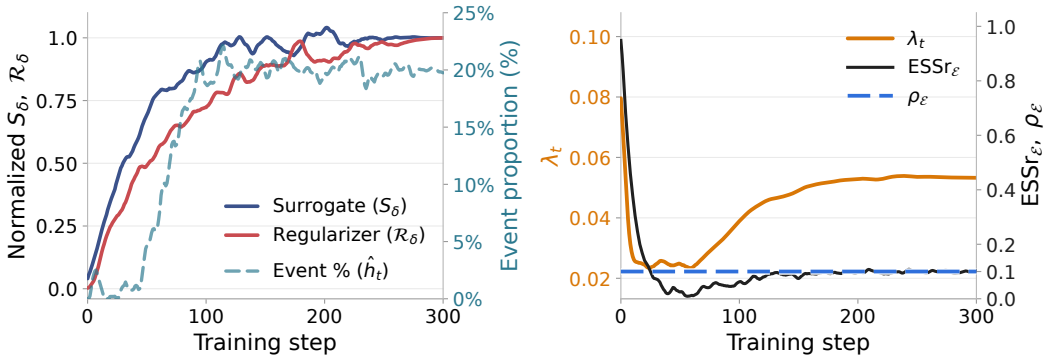
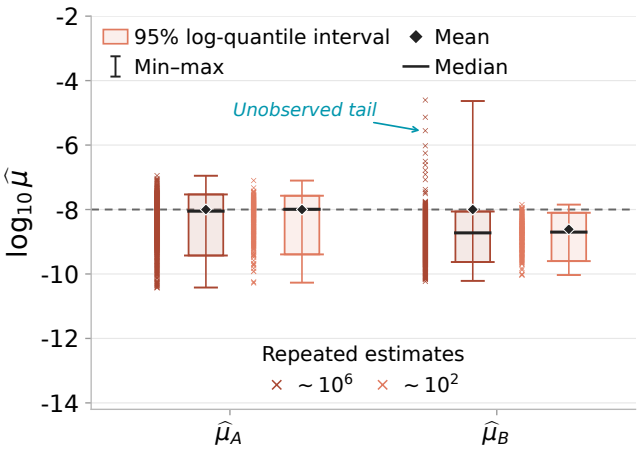


Figure 11. IU training dynamics over 300 optimization steps. IU converges earlier in this run; we retain 300 steps as a conservative training budget. **Left panel.** The empirical event proportion $\hat{h}_t$ (in dashed teal) rises during training. The surrogate S_δ (blue) and regularizer $\mathcal{R}_\delta$ (red) are each normalized by their terminal value for plotting. **Right panel.** Adaptive regularization brings the event-specific $\mathrm{ESSr}_\mathcal{E}$ (black) to its target $\rho_\mathcal{E}$ (dashed blue) by adjusting the multiplier λ_t (orange).



Reported estimation error

Metrics	Repeated estimates	$\hat{\mu}_A$	$\hat{\mu}_B$
rMSE	~ 10^6	*0.71*	*1.4k*
$\hat{E}_{0.95}$ **(ours)**	~ 10^2	1.1	1.5
$\widehat{\text{rMSE}}$	~ 10^2	0.63	0.62

Misleadingly low

Figure 12. Limited repetitions can miss the tails that dominate rMSE. This schematic compares two estimators at a fixed sampling budget per estimate. The constructed reference of 10^6 estimates represents a fuller evaluation of rMSE, including rare extremes that can dominate the loss in heavy-tailed regimes. With only $\sim 10^2$ estimates, these extremes may go unobserved. A's rMSE changes little between the two samples, whereas B's drops sharply when its upper tail is missed, reversing their ranking. Quantile log error favors A in both samples, reflecting B's persistent underestimation. The dashed line marks the true probability; hats denote metrics computed from the smaller sample.

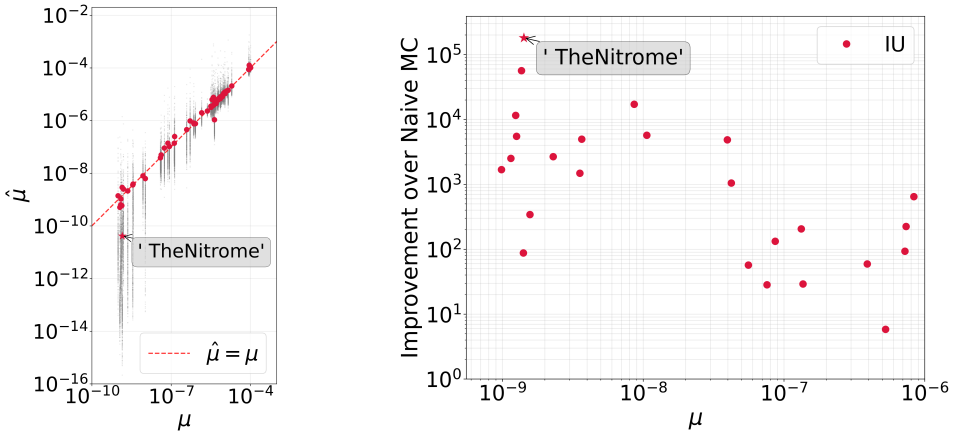


Figure 13. The asymmetric penalty structure of rMSE leads to misleading comparisons in the rare-event regime. **Left panel.** The range of estimates $\widehat{\mu}_k$ versus true probability μ for tokens across six orders of magnitude in rarity. Each black dot represents an independent estimate, with error bars indicating variability across repetitions. The red dashed line denotes perfect estimation ($\widehat{\mu} = \mu$). At $\mu \approx 10^{-10}$, the Token Presence event (Section 5) defined by the token `TheNitrome` (highlighted) shows the worst estimation quality, with estimates highly dispersed and systematically biased below the diagonal. **Right panel.** The improvement factor in rMSE relative to naive Monte Carlo. Paradoxically, we see the largest improvement in rMSE on this event. This occurs because estimators underestimate at this rarity, which does not get penalized much under rMSE's asymmetry.

B Evaluation Events, Surrogates, and Reference Probabilities

B.1 Evaluation Suite Design

We use informative surrogates to isolate the challenge of proposal learning. The surrogate supplies a training signal, but a useful proposal must amplify its occurrence across many prefixes and positions while controlling importance weights. Our baseline trials show that access to an aligned signal does not guarantee event discovery. With the Token Presence surrogate evaluated under the base model, score-guided MCMC recorded no hits for the three rarest tested events under the tested bias schedule and budget (Appendix E.3). Importance-weighted CE ranked trajectories using the same surrogate, but its LoRA proposals produced only one hit each for `atten` and `iminary` in 64,000 fresh samples (Appendix E.1). IU amplified these events sufficiently for estimation. These comparisons showcase the gap in how methods use an available signal.

Repeated accuracy comparisons constrain our evaluation scale. Token Presence includes 60 events with five methods at each of $L = 10, 30, 50$ on GPT-2 Small and 121 IU and IU Logit runs on Gemma-2-2B-IT at $L = 10$: 1,021 runs, each with 64,000 evaluation trajectories. Training, evaluation, and reference estimation together require approximately 25 million forward passes over batches of 128 sequences (Appendix D.3.1). We therefore restrict the more expensive Gemma-2 evaluation to $L = 10$.

B.2 Event-specific surrogates

Ordered Sequence events. Let c_ℓ be the number of targets already completed in order by the prefix $\mathbf{y}_{<\ell}$. Our surrogate rewards the next required token, advancing to the following target when that token appears:

$$S_\delta^{\mathrm{SEQ}}(\mathbf{y}\,;\,\mathbf{v}) \;=\; \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{I}\{c_\ell < m\} \log q_\delta\big(v_{c_\ell+1} \mid \mathbf{y}_{<\ell}\big).$$

Conjunctive events. We use a surrogate that raises the probability of every target token throughout the trajectory:

$$S_\delta^{\mathrm{AND}}(\mathbf{y}\,;\,\mathbf{v}) \;=\; \frac{1}{L}\sum_{j=1}^{m}\sum_{\ell=1}^{L} \log q_\delta\big(v_j \mid \mathbf{y}_{<\ell}\big).$$

Profanity Classifier. The classifier f is a bag-of-words logistic regression that assigns every vocabulary token $u \in \mathcal{V}$ a learned weight β_u. It converts the sum of token weights and a bias into a predicted probability:

$$f(\mathbf{y}) \;=\; \sigma\left(\sum_{\ell=1}^{L} \beta_{y_\ell} + b\right) \;=\; \sigma\big(\beta^\top \mathrm{count}(\mathbf{y}) + b\big)\,,$$

where $\sigma(z) = (1+e^{-z})^{-1}$, $\beta \in \mathbb{R}^{|\mathcal{V}|}$ is the per-token weight vector, b is a bias, and $\mathrm{count}(\mathbf{y}) \in \mathbb{Z}_{\geq 0}^{|\mathcal{V}|}$ is the token-count (bag-of-words) vector of $\mathbf{y}$. The event occurs when $f(\mathbf{y}) > \kappa$.

Traditional rare-event methods often use the same score to specify the event and guide search, as in the score-based LM experiments of Dorman et al. (2026). Although we argue for separating the binary event Φ from the differentiable surrogate S_δ used to train the proposal, we include this setting to show that IU remains effective when the event takes the traditional score-threshold form.

For the Profanity Classifier, we use raw-indicator Monte Carlo reference estimates and choose κ so that verification remains feasible, targeting rarities around $\mu \approx 10^{-5}$. This setting retains a differentiable surrogate even though our reference computation does not use the variance reduction available for single-token presence.

Although f defines the event, it is evaluated on discrete realized tokens and is not differentiable with respect to the LM parameters δ. We therefore use the differentiable surrogate below. We did not compare alternative surrogates in this setting.

Let $\mathbf{q}_\delta(\mathbf{y}_{<\ell})$ denote the vector of proposal next-token probabilities over $\mathcal{V}$ at prefix $\mathbf{y}_{<\ell}$. The surrogate we choose is made differentiable by replacing each realized token weight β_{y_ℓ} with its expectation

under this distribution, $\beta^\top \mathbf{q}_\delta(\mathbf{y}_{<\ell})$. Summing these expected contributions gives a soft version of the sequence-level classifier logit. We subtract the corresponding logit threshold $\log[\kappa/(1-\kappa)]$ and apply a log-sigmoid to provide informative gradients below the threshold:

$$S_\delta^{\mathrm{PROF}}(\mathbf{y}\,;\,\kappa) \;=\; \log\sigma\left(\sum_{\ell=1}^{L}\beta^\top \mathbf{q}_\delta(\mathbf{y}_{<\ell}) + b - \log\frac{\kappa}{1-\kappa}\right).$$

B.3 Reference probability estimation

We estimate reference probabilities using Monte Carlo samples from the frozen base model, with variance reduction for token events. We establish unbiasedness below and report the sample budgets and estimated relative standard errors used in our evaluation.

B.3.1 Token Presence reference estimator

For Token Presence, we use the model's next-token probabilities to reduce reference-estimation variance. Each rollout provides probabilities for the full vocabulary, so the same samples support reference estimates for every token in parallel. For a rollout $\mathbf{y} \sim p$, we replace the sparse binary indicator with a sum of conditional first-hit probabilities:

$$\widehat{\Phi}^{\mathrm{TOK}}(\mathbf{y}; v) = \sum_{\ell=1}^{L} p(v \mid \mathbf{y}_{<\ell})\,\mathbb{I}\{v \notin \mathbf{y}_{<\ell}\}. \tag{5}$$

Let $\mu^{\mathrm{TOK}}(v) = \mathbb{E}_p[\Phi^{\mathrm{TOK}}(\mathbf{y}; v)]$. Decomposition by v's first hitting time gives these mutually exclusive events.

$$\Phi^{\mathrm{TOK}}(\mathbf{y}; v) = \sum_{\ell=1}^{L} \mathbb{I}\{y_\ell = v\}\,\mathbb{I}\{v \notin \mathbf{y}_{<\ell}\} \tag{6}$$

Conditioning on $\mathbf{y}_{<\ell}$ and using $\mathbb{E}[\mathbb{I}\{y_\ell = v\} \mid \mathbf{y}_{<\ell}] = p(v \mid \mathbf{y}_{<\ell})$ gives

$$\mathbb{E}_p[\Phi^{\mathrm{TOK}}(\mathbf{y}; v)] = \mathbb{E}_p\left[\sum_{\ell=1}^{L} p(v \mid \mathbf{y}_{<\ell})\,\mathbb{I}\{v \notin \mathbf{y}_{<\ell}\}\right] = \mathbb{E}_p[\widehat{\Phi}^{\mathrm{TOK}}(\mathbf{y}; v)]. \tag{7}$$

Both $\Phi^{\mathrm{TOK}}(\mathbf{y}; v)$ and $\widehat{\Phi}^{\mathrm{TOK}}(\mathbf{y}; v)$ are therefore unbiased estimators of $\mu^{\mathrm{TOK}}(v)$.

B.3.2 Compositional Token Presence first-completion estimators

The same first-completion argument gives unbiased estimators from next-token probabilities for Compositional Token Presence events. For the Conjunctive event, let $\mathbf{v} = (v_1, \ldots, v_m)$ be a tuple of distinct target tokens and let

$$M_\ell = \{v_1, \ldots, v_m\} \setminus \{y_1, \ldots, y_{\ell-1}\}$$

be the targets still missing immediately before position ℓ. The event is first completed at ℓ exactly when one target remains and that target is emitted. Its reference estimator is therefore

$$\widehat{\Phi}^{\mathrm{AND}}(\mathbf{y}; \mathbf{v}) = \sum_{\ell=1}^{L} \mathbb{I}\{|M_\ell| = 1\} \sum_{u \in M_\ell} p(u \mid \mathbf{y}_{<\ell}). \tag{8}$$

Conditioning on $\mathbf{y}_{<\ell}$ replaces the indicator that the final missing token is sampled with its next-token probability. Because the possible first-completion positions are mutually exclusive,

$$\mathbb{E}_p[\widehat{\Phi}^{\mathrm{AND}}(\mathbf{y}; \mathbf{v})] = \mathbb{E}_p[\Phi^{\mathrm{AND}}(\mathbf{y}; \mathbf{v})].$$

For the Ordered Sequence event, define the prefix state

$$c_\ell = \max\{d \in \{0, \ldots, m\} : (v_1, \ldots, v_d) \text{ is a subsequence of } \mathbf{y}_{<\ell}\}\,.$$

The event is first completed at position ℓ exactly when $c_\ell = m - 1$ and $y_\ell = v_m$. Thus,

$$\widehat{\Phi}^{\mathrm{SEQ}}(\mathbf{y}; \mathbf{v}) = \sum_{\ell=1}^{L} \mathbb{I}\{c_\ell = m - 1\} p(v_m \mid \mathbf{y}_{<\ell}). \tag{9}$$

Applying the tower property at every position and summing the mutually exclusive first-completion events gives

$$\mathbb{E}_p[\widehat{\Phi}^{\mathrm{SEQ}}(\mathbf{y}; \mathbf{v})] = \mathbb{E}_p[\Phi^{\mathrm{SEQ}}(\mathbf{y}; \mathbf{v})].$$

In both cases, the reference probability is the sample mean of $\widehat{\Phi}$ over ordinary rollouts from the frozen base model. This procedure integrates out the final token draw, but a rollout contributes only after the prerequisite targets have appeared. Rare prerequisite occurrences therefore limit the variance reduction. Computing references for all token combinations also requires tracking more event configurations than the single-token computation. The compositional surrogates remain available before the full event has occurred.

B.3.3 Accuracy of the reference probabilities

For N_{gt} independent base-model rollouts, let Z_i be the unbiased first-completion estimate for token-based events or the binary indicator for the Profanity Classifier. We compute the reference probability and its estimated relative standard error as

$$\widehat{\mu}_{\mathrm{gt}} = \frac{1}{N_{\mathrm{gt}}} \sum_{i=1}^{N_{\mathrm{gt}}} Z_i, \qquad \widehat{\mathrm{RSE}} = \frac{s_Z}{\sqrt{N_{\mathrm{gt}}}\, \widehat{\mu}_{\mathrm{gt}}},$$

where s_Z is the sample standard deviation of the Z_i. We use $1{,}024{,}000$ reference rollouts per model and length for Token Presence and $512{,}000$ for compositional and profanity events. These budgets are fixed across events; all estimated relative standard errors are below 10%. Table 2 reports selected events, including the rarest event in each listed setting.

Table 2. Reference probabilities and reported relative standard errors for selected events, including the rarest event in each listed setting.

Setting	Event	Reference samples	Reference probability	Relative SE
GPT-2 Small token, $L = 50$	`ThumbnailImage`	1,024,000	3.97×10^{-9}	0.74%
Gemma-2B token, $L = 10$	`meio`	1,024,000	1.08×10^{-9}	1.32%
Ordered Sequence event, $L = 10$	`the` → `was` → `connect`	512,000	1.25×10^{-6}	7.16%
Conjunctive event, $L = 10$	{`the`, `was`, `connect`}	512,000	3.72×10^{-6}	9.17%
GPT-2 Small profanity, $L = 10$	classifier threshold 0.999	512,000	8.59×10^{-5}	3.82%
GPT-2 Small profanity, $L = 50$	classifier threshold 0.999	512,000	1.44×10^{-3}	3.68%
Gemma-2B profanity, $L = 10$	classifier threshold 0.95	512,000	7.50×10^{-4}	5.10%
Gemma-2B profanity, $L = 10$	classifier threshold 0.999	512,000	6.33×10^{-5}	5.69%

C Methodological Details

C.1 IU's problem formulation

Beyond threshold-based event definitions.

Canonical rare-event formulations often begin with a scalar performance score $g(\mathbf{y})$ and a threshold γ, defining $\Phi(\mathbf{y}) = \mathbb{I}\{g(\mathbf{y}) > \gamma\}$. The score may represent an option price, vehicle velocity, or queue length. In addition to defining the event, g ranks samples by their proximity to it, allowing multilevel or sequential methods to progress toward the rare event through a sequence of intermediate levels.

In LM settings, however, the outcome of interest may be naturally specified only as a binary trajectory-level event, such as "whether an agent issued a combination of tool calls that deleted important data." One could still place this outcome in the canonical form by using an indicator as g. However, such a score alone is not sufficient for proposal search. To guide the proposal toward an as-yet-unobserved event, the score must also provide informative intermediate values on partial trajectories and on completed non-event trajectories.

Another choice for g is the conditional probability that the event eventually occurs,

$$g(\mathbf{y}_{1:\ell}) := p\big(\Phi(\mathbf{y}) = 1 \mid \mathbf{y}_{1:\ell}\big). \tag{10}$$

This score is perfectly aligned with the event of interest; it quantifies, at every prefix, the remaining probability of eventually reaching it. However, computing such a score requires the recursive sum

$$g(\mathbf{y}_{1:\ell}) = \sum_{u \in \mathcal{V}} p(u \mid \mathbf{y}_{1:\ell}) g(\mathbf{y}_{1:\ell} u). \tag{11}$$

with $g(\mathbf{y}_{1:L}) = \Phi(\mathbf{y}_{1:L})$. That is, evaluating such a score requires summing over all possible continuations of the current prefix, of which there can be as many as $|\mathcal{V}|^{L-\ell}$. For even a small language model, this continuation space is enormous. GPT-2 Small has $|\mathcal{V}| = 50{,}257$, which yields about 10^{47} possible sequences even at $L = 10$. Moreover, before the first token is generated, this score is μ, so computing this ideal score already contains the original rare-event estimation problem.

The other option is to learn an auxiliary scoring model from roll-out data. For such a model to define the event exactly, thresholding its output must correctly separate event and non-event trajectories across the relevant sequence space. For it to support proposal search, it must additionally assign informative intermediate values to prefixes and to trajectories on which the event does not occur. Rare-event data provide little direct supervision for either learning objective. When the event probability is μ, one expects only one event hit per $1/\mu$ ordinary roll-outs. Hence training such a model is a difficult rare-event problem in its own right.

Constructing proposals through a separate signal.

We treat the event and the search signal as separate objects. The event indicator Φ is fixed by the risk we want to measure: did the trajectory exhibit the failure of interest? The search signal is introduced only after this event has been specified and instead answers an algorithmic question: in which direction should we move the proposal to observe that failure more often? We refer to this signal as a surrogate and denote it as S_δ. We do not require our surrogate to define the event, estimate its probability, or perfectly rank all trajectories to be a useful signal for proposal search. It may instead capture a correlated or enabling condition for the failure. Increasing it should tend to make $\Phi = 1$ more likely, without needing to characterize the event region exactly. This separation leaves Φ free to represent an arbitrary binary property of a trajectory rather than forcing the risk of interest to conform to a naturally available scalar score.

Related methods use examples of the target behavior or event scores to guide proposal construction Angell et al. (2026); Dorman et al. (2026). For data deletion, executing the generated actions determines whether the event occurred. A differentiable objective based on deletion-related token probabilities can guide proposal search without computing the probability of that outcome.

C.2 IU algorithm

IU updates the proposal parameters by maximizing the regularized surrogate objective $\mathcal{J}_t$ in Eq. (2). Because the training trajectories are sampled from the current proposal, the updates are concentrated

on the regions of the trajectory space that the proposal actively explores. After training, we draw a fresh set of trajectories from the learned proposal q_δ and estimate μ using exact importance weights. Algorithm 1 summarizes the end-to-end procedure.

Let $\hat{h}_t = B^{-1}\sum_{i=1}^{B}\Phi_i$ be the empirical event frequency. We use $(\mathrm{ESSr}_t, \rho_t) = (\mathrm{ESSr}_{\mathrm{batch}}, \rho_{\mathrm{batch}})$ when $\hat{h}_t < h^\star$ and the in-event pair otherwise, with ESSr defined in Section 4.3. We initialize $\lambda_0 \geq \lambda_{\min} > 0$ and use step size $\eta_\lambda > 0$ in the multiplicative update below. We typically choose $\rho_{\mathrm{event}} > \rho_{\mathrm{batch}}$ to target less relative weight dispersion within the event. Appendix D.1 gives the numerical settings. Figure 11 illustrates this training feedback.

Algorithm 1 Iterative Unalignment (IU)

Inputs: $p, \Phi, S_\delta, \mathcal{R}_\delta; T, B, N; \eta_\delta, \eta_\lambda; \rho_{\mathrm{batch}}, \rho_{\mathrm{event}}; h^\star \in (0,1); \lambda_0 \geq \lambda_{\min} > 0.$

$\delta_1 \leftarrow \mathbf{0}$, $\lambda_1 \leftarrow \lambda_0$

for $t = 1, 2, \ldots, T$ **do**

  Sample $\mathbf{y}_1, \ldots, \mathbf{y}_B \overset{\text{iid}}{\sim} q_{\delta_t}$ ▷ Fixed sampled trajectories

  $w_i \leftarrow p(\mathbf{y}_i)/q_{\delta_t}(\mathbf{y}_i)$, $\Phi_i \leftarrow \Phi(\mathbf{y}_i)$ for $i = 1, \ldots, B$

  $\mathcal{J}_t(\delta) \leftarrow \frac{1}{B}\sum_{i=1}^{B}\big[S_\delta(\mathbf{y}_i) - \lambda_t \mathcal{R}_\delta(\mathbf{y}_i)\big]$ ▷ Eq. (2)

  $\delta_{t+1} \leftarrow \delta_t + \eta_\delta \nabla_\delta \mathcal{J}_t(\delta)\big|_{\delta=\delta_t}$ ▷ Update proposal

  $\hat{h}_t \leftarrow B^{-1}\sum_{i=1}^{B}\Phi_i$ ▷ Update event proportion

  $(\mathrm{ESSr}_t, \rho_t) \leftarrow \begin{cases} (\mathrm{ESSr}_{\mathrm{batch}}, \rho_{\mathrm{batch}}), & \hat{h}_t < h^\star, \\ (\mathrm{ESSr}_{\mathrm{event}}, \rho_{\mathrm{event}}), & \hat{h}_t \geq h^\star. \end{cases}$

  $\lambda_{t+1} \leftarrow \max\big\{\lambda_{\min}, \lambda_t \exp\big[\eta_\lambda(\rho_t - \mathrm{ESSr}_t)\big]\big\}$ ▷ Adapt regularization

end for

Sample $\mathbf{y}_1, \ldots, \mathbf{y}_N \overset{\text{iid}}{\sim} q_{\delta_{T+1}}$ ▷ Fresh trajectories for estimation

$\widehat{\mu} \leftarrow \frac{1}{N}\sum_{j=1}^{N}\frac{p(\mathbf{y}_j)}{q_{\delta_{T+1}}(\mathbf{y}_j)}\Phi(\mathbf{y}_j)$ ▷ IS estimator

return $(\widehat{\mu}, q_{\delta_{T+1}})$

ESS ratios and dispersion calibration. For n positive weights $w_1, \ldots, w_n$, write $\bar{w} = n^{-1}\sum_i w_i$ and $\overline{w^2} = n^{-1}\sum_i w_i^2$. The empirical ESS ratio satisfies

$$\widehat{\mathrm{ESSr}} = \frac{(\sum_i w_i)^2}{n\sum_i w_i^2} = \frac{\bar{w}^2}{\overline{w^2}} = \frac{1}{1+\widehat{\mathrm{CV}}^2}, \qquad \widehat{\mathrm{CV}}^2 := \frac{n^{-1}\sum_i (w_i - \bar{w})^2}{\bar{w}^2}.$$

Thus, an ESSr target ρ corresponds to relative weight dispersion $\widehat{\mathrm{CV}}^2 = \rho^{-1} - 1$. Multiplying every weight by the same positive constant leaves this ratio unchanged because its numerator and denominator both scale quadratically. Dividing ESS by n separately removes its linear dependence on sample count. For the full-batch ratio, $n = B$; for the in-event ratio, $n = \sum_i \Phi_i$ and only event weights enter the moments.

For a fixed proposal q_δ with the same support as p, let $w = p/q_\delta$. Since $\mathbb{E}_{q_\delta}[w] = 1$, the population full-batch ratio is

$$\mathrm{ESSr}_{\mathrm{batch}}^{\mathrm{pop}} := \frac{(\mathbb{E}_{q_\delta}[w])^2}{\mathbb{E}_{q_\delta}[w^2]} = \frac{1}{1+\chi^2(p\|q_\delta)} = \exp[-D_2(p\|q_\delta)],$$

where $D_2(P\|Q) = \log \mathbb{E}_Q[(P/Q)^2]$ is Rényi divergence of order two. Within the event, $w = (\mu/r_\delta)(q^*/q_\delta^{\mathcal{E}})$, so

$$\mathrm{ESSr}_{\mathrm{event}}^{\mathrm{pop}} := \frac{(\mathbb{E}_{q_\delta^{\mathcal{E}}}[w])^2}{\mathbb{E}_{q_\delta^{\mathcal{E}}}[w^2]} = \frac{1}{1+\chi^2(q^*\|q_\delta^{\mathcal{E}})} = \exp[-D_2(q^*\|q_\delta^{\mathcal{E}})].$$

The common factor μ/r_δ cancels, giving a fixed in-event ESSr target the same relative-dispersion interpretation as amplification changes. The full-batch ratio also responds to changes in event mass;

the in-event ratio isolates distortion of the conditional distribution within the event. These population identities do not guarantee accurate finite-batch diagnostics: few event hits or unobserved modes can leave empirical ESSr high despite substantial population distortion.

C.3 Fixed-sample gradients

At iteration t, hold λ_t fixed and write $F_{t,\delta}(\mathbf{y}) = S_\delta(\mathbf{y}) - \lambda_t \mathcal{R}_\delta(\mathbf{y})$. Differentiating the population objective $G_t(\delta) = \mathbb{E}_{\mathbf{y}\sim q_\delta}[F_{t,\delta}(\mathbf{y})]$ gives

$$\nabla_\delta G_t(\delta) = \mathbb{E}_{\mathbf{y}\sim q_\delta}\left[\underbrace{\nabla_\delta F_{t,\delta}(\mathbf{y})}_{\text{fixed-sample gradient}} + \underbrace{F_{t,\delta}(\mathbf{y})\nabla_\delta \log q_\delta(\mathbf{y})}_{\text{likelihood-ratio term}}\right]. \tag{12}$$

IU estimates the first term at $\delta = \delta_t$ using the sampled batch. This is the gradient of the fixed-sample objective $\mathcal{J}_t$ in Eq. (2); it is not the full gradient of G_t. The first term differentiates the surrogate and conditional KL directly at each visited prefix. The likelihood-ratio (REINFORCE) term weights $\nabla_\delta \log q_\delta(\mathbf{y}) = \sum_{\ell=1}^{L} \nabla_\delta \log q_\delta(y_\ell \mid \mathbf{y}_{<\ell})$ by the regularized surrogate value $F_{t,\delta}(\mathbf{y})$, so its feedback depends on the sampled token choices. Including this term did not improve performance in our experiments, and we therefore retain the fixed-sample update. The identity $\mathbb{E}_{q_\delta}[\mathcal{R}_\delta] = \mathrm{KL}(q_\delta \| p)$ holds in value; our regularizer update differentiates the conditional KLs at fixed prefixes and omits the contribution from changes in the prefix distribution.

C.4 Surrogate coverage

Recall and precision. The upper level sets of a surrogate define the regions favored by optimization. Recall asks how much of the event these regions contain, while precision asks how much of these regions lies in the event (Figure 9). Non-event trajectories in the favored region consume samples that the exact indicator Φ later sets to zero. Missing event trajectories can instead lead to very small proposal probabilities and large importance weights. This motivates assigning comparably high surrogate values throughout the event, even if non-event trajectories also receive high surrogate values. Such a profile is a design criterion, not a guarantee of coverage after training.

Partial event conditions. An agent may gain write access and then delete a protected file through a tool call. Rewarding deletion-related token probabilities targets only part of this event. The surrogate can also favor calls with the wrong arguments or insufficient permissions. This is the role of a necessary-condition surrogate: it provides a differentiable objective before the full event is observed. Its usefulness depends on whether optimization increases the full event's probability while retaining its important trajectories. The experiment below examines this mismatch in a setting where we can independently verify the event probabilities.

One surrogate, five event definitions. We train a GPT-2 Small proposal to amplify only the token `␣connect`, then hold it fixed while evaluating five events. The central event requires this token alone. The two AND events additionally require `␣the`, or both `␣the` and `␣was`. The surrogate therefore targets a necessary but insufficient condition, analogous to rewarding deletion-related tokens in the tool-use example. The two OR events add alternative tokens, so the event can occur through routes the surrogate does not target. No proposal is retrained for the additional conditions or alternatives.

Figure 14 shows the lowest error when the event matches the surrogate target. The AND cases examine estimation of a stricter event using a proposal trained on only one of its conditions. The OR cases examine the effect of adding event routes that the surrogate does not target. Error increases in both directions. Evaluation on tool-use failures remains future work.

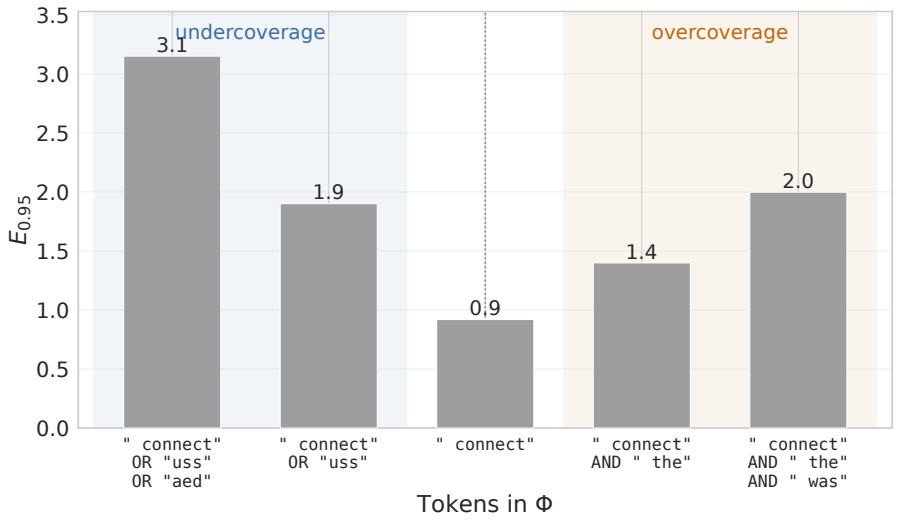


Figure 14. Estimation error under surrogate mismatch. A fixed proposal trained to amplify ␣connect is evaluated on the five token-presence events shown. OR conditions (left) include event trajectories without this token; AND conditions (right) require additional tokens that the surrogate does not target. The central event matches the surrogate's target. Lower error is better. Undercoverage and overcoverage refer to the surrogate's target condition relative to the event.

C.5 Global KL and event-conditional distortion

Let $r_\delta = q_\delta(\mathcal{E})$ and $\mu = p(\mathcal{E})$, with $q_\delta^{\mathcal{E}} = q_\delta(\cdot \mid \mathcal{E})$ and $q^* = p(\cdot \mid \mathcal{E})$. Writing $q_\delta^{\mathcal{E}^c} = q_\delta(\cdot \mid \mathcal{E}^c)$ and $p_{\mathcal{E}^c} = p(\cdot \mid \mathcal{E}^c)$, the KL chain rule over the event partition gives

$$\begin{aligned}\mathrm{KL}(q_\delta \| p) = \underbrace{r_\delta \log \frac{r_\delta}{\mu} + (1 - r_\delta) \log \frac{1 - r_\delta}{1 - \mu}}_{\text{event probability shift}} \\ + \underbrace{r_\delta \, \mathrm{KL}(q_\delta^{\mathcal{E}} \| q^*)}_{\text{within-event distortion}} + \underbrace{(1 - r_\delta) \, \mathrm{KL}(q_\delta^{\mathcal{E}^c} \| p_{\mathcal{E}^c})}_{\text{outside-event distortion}}.\end{aligned} \tag{13}$$

The within-event term retains the conditional KL penalty, weighted by the proposal's event probability r_δ. The remaining terms penalize changes in event probability and distortion outside the event. Applying the regularizer to all proposal trajectories supplies feedback even when a training batch contains no event hits.

C.6 Alternative regularizers

We also considered three alternatives to the regularizer $\mathcal{R}_\delta$ defined in Section 4.2.

Joint forward KL. For the `ActionCode` event, regularization based on base-model rollouts failed to control concentration within the event (Figure 15, left). This alternative uses the joint forward divergence

$$\mathrm{KL}(p \| q_\delta) = \mathbb{E}_{\mathbf{y} \sim p}\left[\log \frac{p(\mathbf{y})}{q_\delta(\mathbf{y})}\right] = \mathbb{E}_{\mathbf{y} \sim p}\left[\sum_{\ell=1}^{L} \mathrm{KL}(p(\cdot \mid \mathbf{y}_{<\ell}) \| q_\delta(\cdot \mid \mathbf{y}_{<\ell}))\right]. \tag{14}$$

At practical training budgets, rollouts from p almost exclusively contain non-event trajectories. A batch of B such rollouts misses the event with probability $(1 - \mu)^B$, so the penalty receives little direct feedback about distortion within $\mathcal{E}$. This causes the proposal to perform poorly.

Restricting regularization to the event. Gating the penalty by the event indicator, using $\Phi(\mathbf{y})\mathcal{R}_\delta(\mathbf{y})$ on proposal rollouts, also degraded IU's performance. When a batch contains no event hits, this penalty is zero on every trajectory, so the surrogate update receives no regularization in early steps. Applying $\mathcal{R}_\delta$ to every rollout constrains early updates before event hits become frequent (Figure 15, center).

On-policy forward KL. Finally, we kept proposal rollouts and reversed the divergence between the next-token distributions,

$$\mathcal{D}_{\mathrm{on}}(\delta) = \mathbb{E}_{\mathbf{y}\sim q_\delta}\left[\sum_{\ell=1}^{L} \mathrm{KL}(p(\cdot \mid \mathbf{y}_{<\ell}) \| q_\delta(\cdot \mid \mathbf{y}_{<\ell}))\right]. \tag{15}$$

The outer expectation follows q_δ, while the inner KL weights tokens under p. This is the forward-KL objective used in on-policy distillation, where a student learns from teacher feedback on its own generated sequences Agarwal et al. (2024). It differs from joint forward KL in Eq. (14), whose prefixes follow p. Both on-policy directions provide feedback across the full vocabulary at each visited prefix. Neither directly compares conditionals at unvisited prefixes.

As in on-policy distillation, we hold the generated tokens fixed when differentiating the conditional divergences. Agarwal et al. (2024) also consider reverse KL and combine distillation with reward optimization. IU uses this form of feedback to regularize an importance-sampling proposal, with the base model supplying the reference probabilities for both regularization and estimation.

The two variants gave similar pooled probability estimates in this comparison (0.98μ for reverse KL and 0.94μ for forward KL), despite different event frequencies and weight distributions (Figure 15, right). The preferred direction remains problem specific, consistent with the findings of Agarwal et al. (2024).

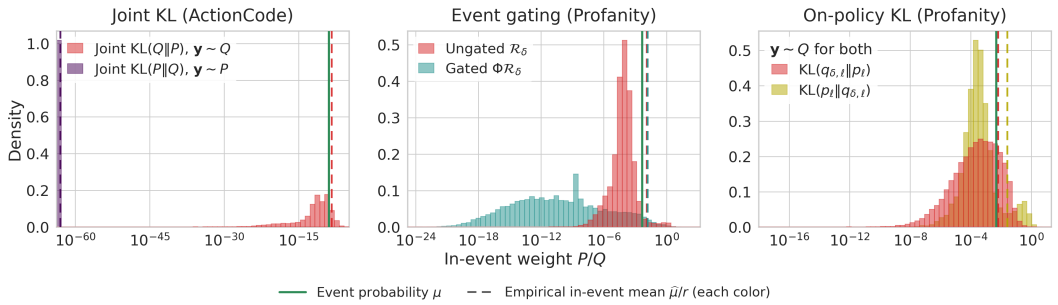


Figure 15. Alternative regularizers, compared through in-event importance weights p/q_δ at $L = 10$. Red denotes our choice in each panel. **Left:** joint reverse KL on proposal rollouts and joint forward KL on base-model rollouts for `ActionCode`. **Center:** regularization on every proposal rollout and regularization gated by Φ. **Right:** reverse and forward next-token KL, both evaluated on proposal rollouts. The center and right panels use the Profanity Classifier. The green line marks μ, the weight under the zero-variance proposal; colored dashed lines mark empirical mean in-event weights for each proposal.

Connection to KL-regularized reinforcement learning. Increasing an event probability from 10^{-9} to 10^{-1} requires 10^8-fold amplification. For estimation, the proposal must also retain the event trajectories that carry probability under the base model. Concentrating on a few event paths can increase event frequency while producing unstable importance weights. KL-regularized reinforcement learning also combines a reward with a penalty for departing from a reference model. For example, Ziegler et al. (2019) adapt the penalty coefficient to target a specified KL divergence. We adapt the coefficient using observed importance-weight dispersion. Once events are frequent enough, the in-event ESS ratio separates this dispersion from common event amplification because the common scale factor in the weights cancels (Section 4.3).

Our surrogate guides proposal search, while the exact indicator and likelihood ratios determine the final probability estimate. Even a full gradient of the surrogate objective does not by itself ensure event coverage or low estimator variance. IU combines direct surrogate gradients with conditional KL feedback to seek the amplification needed for estimation, and uses ESS to adjust the strength of this regularization. Empirical ESS cannot certify coverage of unobserved event paths.

C.7 Per-step chi-square and importance-weight variance

A related alternative to the reverse-KL penalty is an unweighted sum of per-step chi-square divergences. This gave the best empirical performance among the chi-square implementations we tested,

so we use it in the comparison in Figure 10:

$$\mathcal{D}_\chi(q) := \mathbb{E}_{\mathbf{Y}\sim q}\left[\sum_{\ell=1}^{L} \chi^2(p_\ell \| q_\ell)\right]. \tag{16}$$

The exact joint chi-square, $\chi^2(p\|q) = \mathbb{E}_q[\sum_{\ell=1}^{L} W_{\ell-1}^2 \chi^2(p_\ell\|q_\ell)]$, scales these local terms by the importance weights of the prefixes, $W_{\ell-1}^2$. Since these weights multiply out over the sampled tokens, they can vary sharply between trajectories and introduce severe instability into the penalty itself. By dropping the prefix weights, the unweighted sum avoids this compounding variance while retaining dense optimization feedback at every visited prefix.

This unweighted surrogate remains sound for controlling the proposal divergence. With a finite vocabulary $\mathcal{V}$ and fixed generation length L, the unweighted penalty $\mathcal{D}_\chi(q)$ is zero if and only if $q = p$. More generally, for proposals q_n satisfying the same support assumption, driving the unweighted penalty to zero guarantees convergence of the true joint divergence:

$$\mathcal{D}_\chi(q_n) \longrightarrow 0 \quad \Longrightarrow \quad \chi^2(p\|q_n) \longrightarrow 0. \tag{17}$$

C.8 Further fine-tuning the proposal

While IU employs a surrogate to provide a continuous gradient for the base proposal distribution, it is natural to then ask whether the proposal can be further improved after the initial IU optimization. Once a high-coverage proposal q_δ is established, it triggers the rare event frequently enough that the exact indicator $\Phi(\mathbf{y})$ provides a usable signal. At this point, one could discard the surrogate and fine-tune the proposal further using other techniques. Exploring these subsequent refinement steps is an interesting direction, but it lies outside the scope of this work. We view such fine-tuning as naturally building upon the initial proposal that IU provides.

C.9 Input context and autoregressive output length

Input context and generated output play different roles in IU. Conditional on a fixed input $\mathbf{x}$, the trajectory importance weight is a product of L token-level likelihood ratios, one for each generated token. Increasing the output length therefore adds stochastic decoding steps over which discrepancies between the proposal and base model can accumulate. A longer fixed input supplies conditioning information but does not itself add factors to this product. The output length thus determines the size of the sequence space over which IU constructs its proposal.

D Experimental Protocol

D.1 Implementation and Hyperparameters

Training and evaluation. For Token Presence, we use batches of 128 trajectories and a training budget of 300 steps per proposal, totaling 38,400 training trajectories. IU converges earlier in the run shown in Figure 11, but we retain the fixed 300-step budget conservatively across events. Optimization is performed using Adam (specifically AdamW) for all gradient-based proposal updates. We then freeze the proposal and draw 500 evaluation batches, totaling 64,000 trajectories.

IU. LoRA modifies model transformations through trainable adapter weights. Preliminary experiments showed comparable performance with full-parameter tilting, so we use LoRA to reduce training cost. Both GPT-2 Small and Gemma-2 use LoRA with rank $r = 16$ and scaling factor $\alpha_{\text{LoRA}} = 32$. Proposal parameters are optimized using Adam with learning rate $\eta_\delta = 10^{-4}$. We initialize the regularization strength at $\lambda_0 = 10$, impose a floor of $\lambda_{\min} = 0.1$, and use dual learning rate $\eta_\lambda = 10^{-2}$. The controller targets population ESS ratio $\rho_{\text{batch}} = 0.005$ while the event frequency is below $h^\star = 0.1$. Once at least 10% of the batch satisfies the event, it targets event ESS ratio $\rho_{\text{event}} = 0.1$.

IU AS and IU Logit. IU AS adds a single vector of the hidden-layer dimension to the residual stream at the output of every transformer block, after its attention and MLP computations. The vector is shared across layers and token positions. IU Logit adds a single $|\mathcal{V}|$-dimensional vector to the output logits. Both variants keep the base-model weights fixed and optimize the objective in Eq. (2) using Adam, treating sampled tokens as fixed. They use the same regularization and adaptive λ tuning as IU. Their proposal learning rates are 10^{-2} for IU AS and 1 for IU Logit. IU AS initializes each steering-vector coordinate to 0.01; IU Logit initializes its coordinates independently from a zero-mean Gaussian with standard deviation 0.01.

CE AS and CE Logit. Both CE methods use a population of 128 steering vectors, elite fraction $\gamma_{\text{elite}} = 0.3$, and smoothing coefficient $\tau_{\text{CE}} = 0.5$. The initial search standard deviation and its floor are both 0.1 for CE AS and 1 for CE Logit. The search mean is initialized to zero, and generation uses temperature 1. Evaluation fixes the steering vector at the learned search mean.

After 10 warmup training steps, CE stops early if the event ESS ratio falls below 0.1 for two consecutive steps, each with at least two event hits. Appendix E.1 describes the CE formulations and their empirical behavior.

D.2 Evaluation Metrics and Efficiency-Gain Estimation

For each estimator–event pair with reference probability $\mu > 0$, we use 64,000 evaluation trajectories to form K independent estimates $\{\widehat{\mu}_{k,N}\}_{k=1}^{K}$, each based on N trajectories.

Limitations of empirical relative mean squared error. We estimate relative mean squared error (rMSE) as $\frac{1}{K}\sum_{k=1}^{K}(\widehat{\mu}_{k,N}/\mu - 1)^2$. Underestimates contribute at most 1, while overestimates incur unbounded quadratic penalties. In our setting, importance weights multiply across tokens and can develop long tails, so rare, extreme estimates can dominate rMSE. Reliable variance-based comparisons may consequently require prohibitively many runs (L'Ecuyer et al., 2010). Figures 12 and 13 illustrate this instability.

Log error penalizes reciprocal over- and underestimates equally, while penalizing underestimation more heavily for equal additive deviations from μ. Section 5 defines the quantile log error used in our comparisons.

Required budgets. We estimate the required budgets in Eq. (4) across error thresholds. For example, $E^\star_{0.95} = 1$ requires at least 95% of estimates to fall within a factor of ten of μ.

Bootstrapping and MC continuation. We bootstrap the stored estimates up to four times the evaluation budget, or 256,000 trajectories from 64,000 stored trajectories. If the target remains unmet, we add simulated naive-MC samples, since naive MC gives an unbiased estimate of the event probability. We simulate event counts from the Binomial distribution using the reference probability, retain the initial IS contributions, and count all IS and MC samples in the estimate and budget.

Limitations. This gives a conservative sketch when further IS sampling would outperform naive MC, since we assume no further IS advantage and retain the initial IS error and cost. Bootstrapping can also understate gains when the stored data miss rare event contributions. These estimates remain subject to bootstrap uncertainty and are not statistical lower bounds. In plots that average across events, the error-threshold axis stops at 0.5 because tighter targets can require more naive-MC samples than we can validate within our compute budget.

D.3 Computational Costs

The upfront proposal-training cost is $\mathrm{cost}_{\mathrm{train}}$; $\mathrm{cost}_{\mathrm{IS}}$ and $\mathrm{cost}_{\mathrm{MC}}$ are the costs per IS and naive-MC trajectory, respectively. At budget $N^\star$, the total IS cost is $\mathrm{cost}_{\mathrm{train}} + N^\star \mathrm{cost}_{\mathrm{IS}}$, or $\mathrm{cost}_{\mathrm{IS}} + \mathrm{cost}_{\mathrm{train}}/N^\star$ per trajectory. Compute-Weighted Efficiency Gain compares the naive-MC cost $N^\star_{\mathrm{MC}}\mathrm{cost}_{\mathrm{MC}}$ with this total IS cost at the same error threshold:

$$N^\star_{\mathrm{MC}}\,\mathrm{cost}_{\mathrm{MC}}\big(\mathrm{cost}_{\mathrm{train}} + N^\star\,\mathrm{cost}_{\mathrm{IS}}\big)^{-1}.$$

Table 3 reports the measured costs.

Table 3. Wall-clock costs for Compute-Weighted Efficiency Gain, measured on an isolated NVIDIA A100-SXM4-80GB GPU. IU uses LoRA ($r = 16$, $\alpha_{\mathrm{LoRA}} = 32$); training cost is for $T = 300$ steps at $B = 128$.

Model	L	$\mathrm{cost}_{\mathrm{MC}}$ (s / trajectory)	$\mathrm{cost}_{\mathrm{IS}}$ (s / trajectory)	$\mathrm{cost}_{\mathrm{train}}$ (s)	$\mathrm{cost}_{\mathrm{IS}}/\mathrm{cost}_{\mathrm{MC}}$
GPT-2 Small	10	0.0046	0.0051	213	1.10
GPT-2 Small	30	0.0138	0.0153	640	1.10
GPT-2 Small	50	0.0230	0.0254	1067	1.10
Gemma-2-2B-IT	10	0.0501	0.0569	2718	1.14

D.3.1 Forward-pass accounting for Token Presence

A model forward pass over 128 sequences counts as one pass. GPT-2 has 60 events at each of $L = 10, 30, 50$, evaluated with five methods. Gemma-2 has 55 IU and 66 IU Logit runs, all at $L = 10$. This gives $60 \cdot 3 \cdot 5 + 55 + 66 = 1{,}021$ runs, each with 300 training and 500 evaluation batches.

Generating a batch of length L requires L forward passes. The generation budget is therefore

$$\begin{aligned} F_{\mathrm{gen}} &= (300 + 500)\big[60 \cdot 5 \cdot (10 + 30 + 50) + (55 + 66) \cdot 10\big] \\ &= 22{,}568{,}000. \end{aligned}$$

Each batch requires a scoring pass under each model, giving $2 \cdot (300 + 500) \cdot 1{,}021 = 1{,}633{,}600$ scoring passes.

Reference estimation uses $1{,}024{,}000$ sequences, or $8{,}000$ batches, in each of the four model–length settings. Each batch requires L generation passes and one scoring pass, giving

$$F_{\mathrm{ref}} = 8{,}000\big[(10 + 1) + (30 + 1) + (50 + 1) + (10 + 1)\big] = 832{,}000.$$

The total budget is $25{,}033{,}600$ forward passes. Broken down, this is 22.57 million for generation, 1.63 million for scoring, and 0.83 million for reference estimation.

E Baseline Comparisons

E.1 Cross-entropy baseline selection

We evaluated two CE formulations: importance-weighted CE for rare-event estimation and unweighted CE for optimization. Both rank sequences using $G_\delta(\mathbf{y}) = S_\delta(\mathbf{y})$ for Token Presence and Compositional Token Presence, and $G_\delta(\mathbf{y}) = \log[f(\mathbf{y})/(1-f(\mathbf{y}))]$ for the Profanity Classifier, the logit of its predicted probability. We write Top_a for the indices of the top a fraction of ranking values.

Importance-weighted CE. Classical rare-event CE maximizes elite-sequence likelihood with weights proportional to $p(\mathbf{y})/q_{\delta_t}(\mathbf{y})$ de Boer et al. (2005); Rubinstein & Kroese (2004). We approximately maximize this objective by gradient ascent, holding the sampled sequences, elite membership, and weights fixed (Algorithm 2). With LoRA, these updates provided insufficient event amplification (Table 4); `atten` and `iminary` each appeared only once in 64,000 fresh samples. Activation and logit steering also failed to find `ActionCode` at $L = 30$ ($\mu = 2.812 \times 10^{-9}$), with zero hits in 45,056 and 98,304 training trajectories, respectively.

Algorithm 2 Importance-weighted CE with gradient updates

Inputs: $p, q_\delta, \Phi, G_\delta; T, B, \eta; \rho, h^\star$.

$\delta_1 \leftarrow \mathbf{0}$
for $t = 1, 2, \ldots, T$ **do**
  Sample $\mathbf{y}_1, \ldots, \mathbf{y}_B \overset{\text{iid}}{\sim} q_{\delta_t}$ ▷ Fixed sampled trajectories
  $\hat{h}_t \leftarrow B^{-1} \sum_{i=1}^{B} \Phi(\mathbf{y}_i)$ ▷ Update event proportion
  if $\hat{h}_t \geq h^\star$ **then**
    return q_{δ_t} ▷ Event-proportion target reached
  end if
  $\mathcal{I}_t \leftarrow \mathrm{Top}_\rho\big(\{G_{\delta_t}(\mathbf{y}_i)\}_{i=1}^{B}\big)$ ▷ Select elite trajectories
  $w_i \leftarrow p(\mathbf{y}_i)/q_{\delta_t}(\mathbf{y}_i)$, $\bar{w}_i \leftarrow w_i / \sum_{j \in \mathcal{I}_t} w_j$ for $i \in \mathcal{I}_t$
  $F_t(\delta) \leftarrow \sum_{i \in \mathcal{I}_t} \bar{w}_i \log q_\delta(\mathbf{y}_i)$ ▷ Fixed elites and weights
  $\delta_{t+1} \leftarrow \delta_t + \eta \nabla_\delta F_t(\delta_t)$ ▷ Gradient ascent
end for
return $q_{\delta_{T+1}}$ ▷ Proposal for IS estimation

Event	Reference value μ	Event hits	Amplification	Final estimate
`␣mere`	6.1594×10^{-4}	64	1.62×	6.3742×10^{-4}
`atten`	1.1463×10^{-5}	1	1.36×	1.0166×10^{-5}
`iminary`	1.7840×10^{-6}	1	8.76×	9.0076×10^{-6}

Table 4. Importance-weighted CE with LoRA gradient updates on GPT-2 Small at $L = 10$. Each event uses one training seed and 300 rounds of 128 trajectories, with elite fraction 0.15 and one gradient step per round at learning rate 10^{-3}. Event hits and ordinary IS estimates use 64,000 fresh samples from each frozen proposal. Amplification is the observed event frequency divided by μ.

Unweighted CE optimization. In our unweighted LoRA likelihood trials, we instead took gradient steps on the unweighted elite-sequence log likelihood; these updates concentrated the proposal too strongly on the observed elite sequences, resulting in severe underestimation. The standard CE optimization search performed better empirically. This version of the algorithm samples steering vectors, ranks them by the values of G_δ on their generated sequences, and moves the search mean toward the unweighted average of the elite vectors Rubinstein & Kroese (2004); Botev et al. (2013) (Algorithm 3). We retain this update for activation steering (CE AS) and logit steering (CE Logit). At $L = 10$, it achieves substantial estimator and compute-weighted efficiency gains over naive Monte Carlo at some accuracy thresholds (Figure 27).

Algorithm 3 Unweighted CE optimization with activation/logit tilting

Inputs: $p, q_\delta, G_\delta; T, B; \gamma_{\text{elite}}, \sigma^2_{\text{CE}}; \tau_{\text{CE}} \in (0, 1]$.

$\mathbf{m}_1 \leftarrow \mathbf{0}$
for $t = 1, 2, \ldots, T$ **do**
  Sample $\delta^{(1)}, \ldots, \delta^{(B)} \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{m}_t, \sigma^2_{\text{CE}}\mathbf{I})$ ▷ Steering vectors
  Sample $\mathbf{y}_i \sim q_{\delta^{(i)}}$ for $i = 1, \ldots, B$ ▷ One sequence per vector
  $\mathcal{I}_t \leftarrow \text{Top}_{\gamma_{\text{elite}}}\big(\{G_{\delta^{(i)}}(\mathbf{y}_i)\}_{i=1}^B\big)$ ▷ Select elite perturbations
  $\mathbf{m}_{t,\text{elite}} \leftarrow |\mathcal{I}_t|^{-1} \sum_{i \in \mathcal{I}_t} \delta^{(i)}$ ▷ Elite mean
  $\mathbf{m}_{t+1} \leftarrow (1 - \tau_{\text{CE}})\mathbf{m}_t + \tau_{\text{CE}}\mathbf{m}_{t,\text{elite}}$ ▷ Update search mean
end for
return $\mathbf{m}_{T+1}$ ▷ Learned perturbation

Connection to likelihood-ratio gradients. For Token Presence, an elite sequence can have a high surrogate value without containing the target token. The likelihood update in Algorithm 2 uses the sampled tokens in these elite sequences as fitting targets. Holding the elite set and importance weights fixed gives

$$\nabla_\delta F_t(\delta) = \sum_{i \in \mathcal{I}_t} \bar{w}_i \nabla_\delta \log q_\delta(\mathbf{y}_i). \tag{18}$$

This has the same weighted log-likelihood gradient form as the likelihood-ratio term in Eq. (12). The weights and objectives differ. CE uses elite membership and normalized importance weights, while the likelihood-ratio term uses the regularized surrogate value $F_{t,\delta}(\mathbf{y})$. Related weighted likelihood updates appear in the policy-gradient approach to rare-event estimation of Corso et al. (2022), which builds on CE.

Direct likelihood-fitting implementations of CE performed poorly at the tested budgets. Importance-weighted fitting gave little amplification, while our unweighted LoRA likelihood trials concentrated the proposal and led to severe underestimation. Including the REINFORCE term also did not improve IU. These results show that reinforcing observed sequences alone was insufficient for the high-dimensional proposal families we evaluated. IU directly differentiates the surrogate at each sampled prefix. For Token Presence, this gives a gradient of the target token's log probability even when the token was absent from the rollout.

E.2 Twisted Sequential Monte Carlo

Twisted sequential Monte Carlo (TSMC) uses intermediate functions to guide particles toward promising continuations. These can be learned twist functions Zhao et al. (2024). In our implementation, the surrogate guides particle resampling through look-ahead evaluations, without updating the LM's weights. IU instead uses surrogate gradients to train a proposal that changes the token conditionals throughout generation.

We evaluated a TSMC variant using our surrogate $S_\delta(\mathbf{y})$ as the resampling criterion with a two-step look-ahead, averaged across 20 look-ahead particles. Across 10 token events with probabilities between 10^{-6} and 10^{-9}, this variant yielded an average efficiency ratio of $0.87\times$ relative to naive Monte Carlo. It did not consistently improve upon naive Monte Carlo, so we omitted it from the main baseline comparisons.

In these runs, repeated resampling concentrated particles onto a small number of prefixes with high surrogate values, reducing diversity among downstream trajectories. This behavior highlights the need to preserve diverse event paths while directing sampling toward the event, which motivates the combination of amplification and regularization in IU.

E.3 Score-guided MCMC

Dorman et al. (2026) study rare-event estimation in language models by reconstructing the distribution of continuous scalar scores $g(\mathbf{y})$ (such as output log probability and readability) under the base model

p using score-biased MCMC and multistate Bennett acceptance ratio (MBAR) reweighting. Their framework also supports estimating a target indicator $\Phi(\mathbf{y})$ using a correlated auxiliary function to bias the MCMC transitions. We use this formulation to compare score-guided MCMC with IU on Token Presence events.

We set the target to the binary event indicator $\Phi(\mathbf{y})$ and use the Token Presence surrogate as the biasing function, $g(\mathbf{y}) = S_\delta^{\mathrm{TOK}}(\mathbf{y}; v)$, evaluated with $q_\delta = p$. This surrogate directly rewards prefixes that assign higher probability to the target token, providing an aligned signal for MCMC.

For each event, we run 10 chains across the positive bias schedule $0.1, 0.2, \ldots, 1.0$, with 1,219 transitions per chain and bias, 10% burn-in, and 6,090 direct samples. Including the 10 initial chain states, this gives 128,000 evaluated trajectories for one complete estimate. We retain rejected states and exclude a bias only when its Gelman–Rubin diagnostic exceeds 1.1, following the authors' procedure. The schedule is fixed across events and receives no event-specific tuning.

Dorman's MCMC transition truncates each trajectory at a uniformly random output position and regenerates the remaining suffix. We conservatively charge each evaluated trajectory as only half a naive-MC inference. From Table 3, the charged cost is

$$128{,}000(0.5)\,\mathrm{cost}_{\mathrm{MC}} = 128{,}000(0.5)(0.0046) = 294.4 \text{ seconds.}$$

For comparison, the full cost of one $N = 128$ IU estimate (including proposal training) is

$$\mathrm{cost}_{\mathrm{train}} + 128\,\mathrm{cost}_{\mathrm{IS}} = 213 + 128(0.0051) = 213.65 \text{ seconds.}$$

Thus, in this experiment, Dorman's charged budget is $294.4/213.65 = 1.38\times$ the full cost of an $N = 128$ IU estimate.

Table 5 reports event discovery and probability estimation. The event hit rate is the fraction of sampled trajectories satisfying the Token Presence event.

(a) Iterative Unalignment

Event	Reference μ	Event hit rate	Final estimate
`mere`	6.1594×10^{-4}	0.620	6.1927×10^{-4}
`atten`	1.1463×10^{-5}	0.167	7.4793×10^{-6}
`iminary`	1.7840×10^{-6}	0.311	2.4233×10^{-6}
`sembly`	1.1767×10^{-7}	0.345	9.3520×10^{-8}
`soDeliveryDate`	1.4414×10^{-8}	0.264	1.2899×10^{-8}

(b) Dorman et al.

Event	Reference μ	Event hit rate	Final estimate
`mere`	6.1594×10^{-4}	1.04×10^{-3}	6.0048×10^{-4}
`atten`	1.1463×10^{-5}	6.67×10^{-5}	7.3987×10^{-6}
`iminary`	1.7840×10^{-6}	0	0
`sembly`	1.1767×10^{-7}	0	0
`soDeliveryDate`	1.4414×10^{-8}	0	0

Table 5. IU and score-guided MCMC on GPT-2 Small Token Presence events at $L = 10$. Under the compute accounting above, IU makes every event common under its proposal, while Dorman et al. yield zero event hits for events with reference probabilities below $\sim 10^{-5}$.

These results illustrate the structural distinction between trajectory-space MCMC and global proposal optimization. Score-guided MCMC relies on local suffix mutations, which can struggle to discover isolated target modes in high-dimensional discrete sequence spaces when the event is exceedingly rare ($\mu < 10^{-5}$). In such regimes, MCMC chains may fail to record event hits despite satisfying standard internal diagnostics (such as Gelman–Rubin $\hat{R} \leq 1.1$). In contrast, IU optimizes model parameters globally, steering autoregressive decoding dynamics across all prefix positions to reliably sample deep-tail events and provide accurate importance-sampling estimates.

F Additional Results and Examples

F.1 Additional Main Results

The main text selects one length from each three-panel comparison. Here we give the full sequence-length and computational cost comparisons, together with all small-budget error plots and the full Ordered Sequence and Conjunctive tables. For the small-budget plots, each estimate uses $N = 128$ trajectories after training, and lower quantile log error is better. We summarize errors, required budgets, and efficiency gains across events using geometric means. This averages proportional changes symmetrically and gives aggregate gains no larger than the arithmetic mean. Shading shows ± 0.5 standard deviations in log space.

Target tokens	Rarity μ	MC samples $N^{\star}_{\mathrm{MC}}$	IU samples $N^{\star}_{\mathrm{IU}}$	Estimator efficiency gain	Compute-weighted efficiency gain
`the` → `was` → `and`	2.59×10^{-3}	1,156	96	12×	0.0169×
`the` → `was` → `people`	1.03×10^{-4}	28,999	256	113×	0.423×
`a` → `the` → `king`	3.02×10^{-5}	99,215	256	388×	1.44×
`the` → `was` → `village`	5.75×10^{-6}	521,040	512	1,020×	7.55×
`the` → `was` → `connect`	1.25×10^{-6}	2,402,465	1,152	2,090×	35.3×

Table 6. Ordered Sequence (SEQ) events on GPT-2 Small at $L = 10$, requiring factor-ten accuracy in 95% of estimates. Targets appear in order, but need not be contiguous. IU reduces sample requirements for every event; gains including training cost are largest for the rarest events.

Target tokens	Rarity μ	MC samples $N^{\star}_{\mathrm{MC}}$	IU samples $N^{\star}_{\mathrm{IU}}$	Estimator efficiency gain	Compute-weighted efficiency gain
`{the, was, and}`	6.57×10^{-3}	455	256	1.78×	0.00686×
`{the, was, people}`	5.05×10^{-4}	5,927	384	15.4×	0.0893×
`{a, the, king}`	1.98×10^{-4}	15,118	384	39.4×	0.228×
`{the, was, village}`	8.06×10^{-5}	37,162	640	58.1×	0.557×
`{the, was, connect}`	3.72×10^{-6}	804,821	4,096	196×	11.3×

Table 7. Conjunctive (AND) events on GPT-2 Small at $L = 10$, requiring factor-ten accuracy in 95% of estimates. Every target appears, in any order. IU reduces sample requirements for every event; gains including training cost are largest for the rarest events.

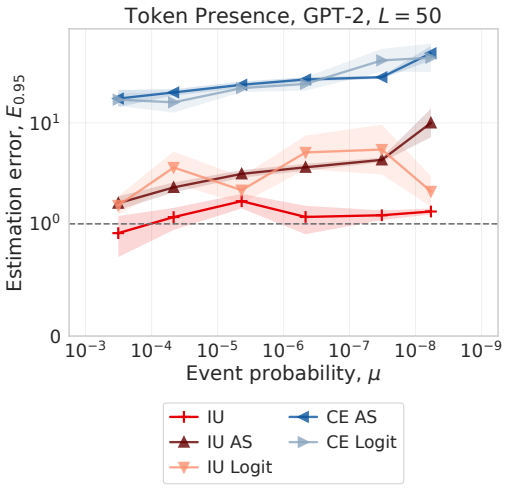


Figure 16. Token Presence error at $N = 128$ on GPT-2 Small at $L = 50$. Each point summarizes ten events per rarity band. IU's errors remain comparatively stable as events become rarer. Across the 60 GPT-2 Small events with $10^{-9} \leq \mu < 10^{-7}$ at $L = 10$, 30, and 50, its geometric-mean quantile log error is 1.17 orders of magnitude.

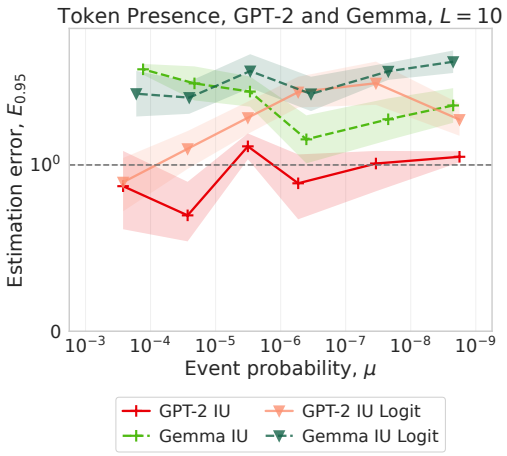


Figure 17. Token Presence model comparison at $N = 128$ and $L = 10$. Each point summarizes ten events per rarity band and model. IU's errors are larger on Gemma-2 than on GPT-2 Small.

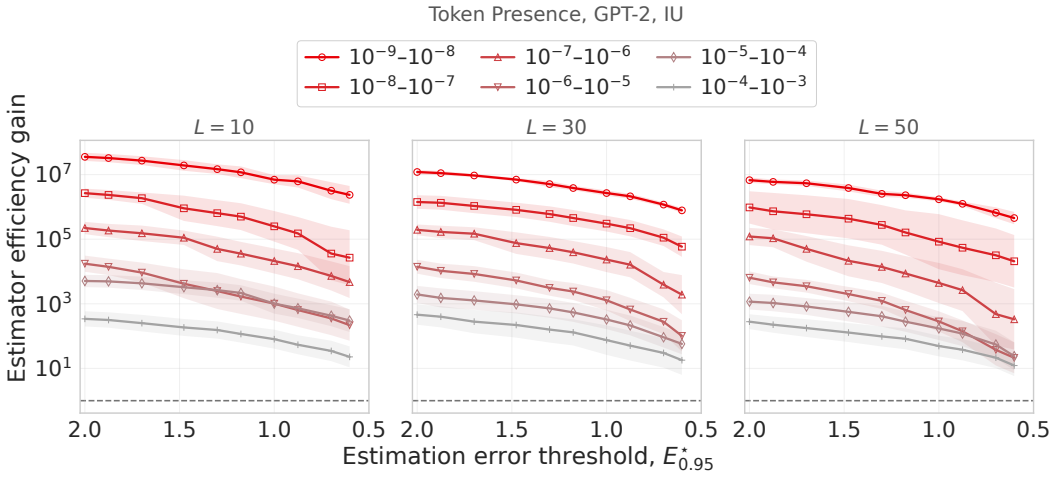


Figure 18. IU estimator efficiency gains for GPT-2 Small Token Presence at $L = 10$, 30, and 50 (left to right). Gains tend to increase with event rarity across all three lengths. Each curve shows a geometric mean over ten events with neighboring rarity. Higher gains are better, and accuracy requirements become stricter toward the right. At $E^{\star}_{0.95} = 1$ in $10^{-8} \leq \mu < 10^{-7}$, geometric-mean IU versus naive-MC budgets are approximately 349 versus 8.73×10^7, 255 versus 7.73×10^7, and 1,103 versus 9.24×10^7 trajectories, respectively. Absolute budgets appear in Appendix Figure 22.

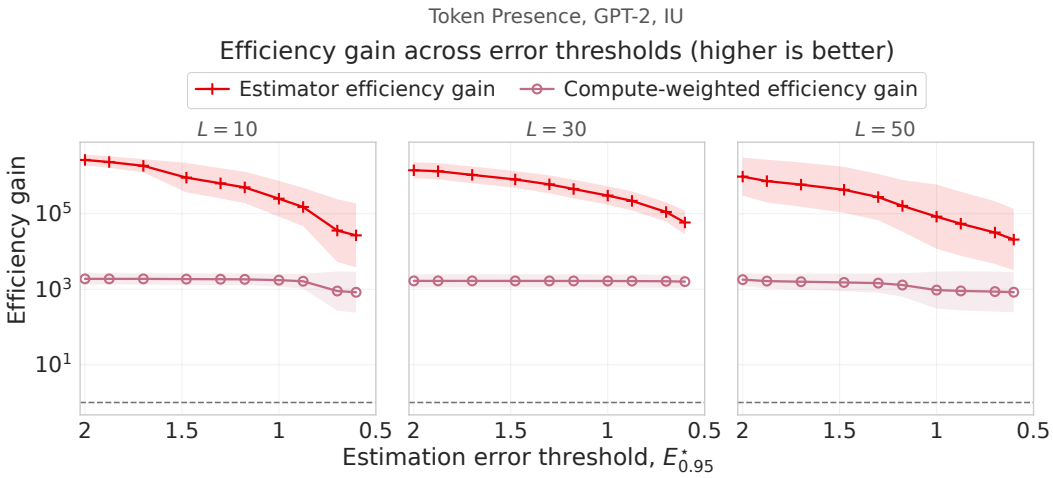


Figure 19. IU estimator efficiency and compute-weighted efficiency gains on GPT-2 Small at $L = 10$, 30, and 50 (left to right). As accuracy requirements tighten, training accounts for less of the total cost and the gap narrows. Curves show geometric means over ten events per length in $10^{-8} \leq \mu < 10^{-7}$. At $E^{\star}_{0.95} = 1$, the geometric-mean compute-weighted efficiency gains over naive MC are approximately $1{,}750\times$, $1{,}660\times$, and $949\times$, respectively. See Appendix Figure 25 for absolute sample and time budgets.

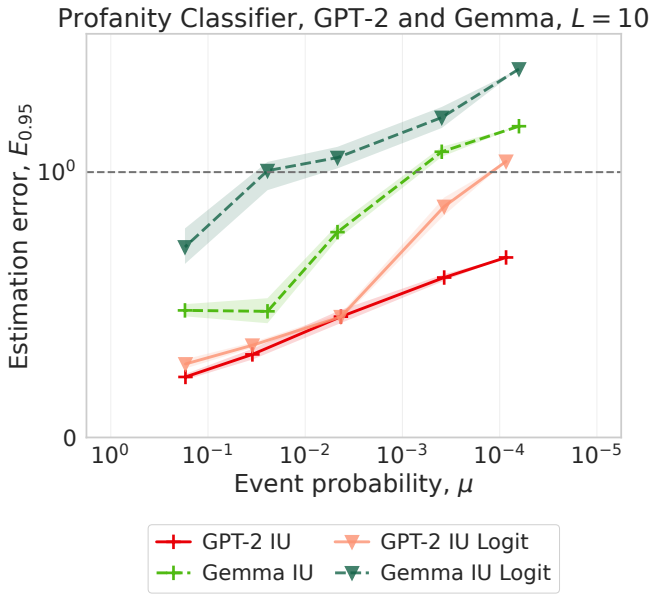


Figure 20. Profanity Classifier error at $N = 128$ on GPT-2 Small and Gemma-2 at $L = 10$. IU achieves lower errors than IU Logit on both models. Each point combines five events with neighboring rarity.

F.2 Absolute Sampling and Computational Budgets

Figures 21–25 show the absolute budgets underlying each main-text efficiency figure, using the same events, methods, error thresholds, and geometric-mean summaries. Lower budgets are better. Dashed or dotted curves show naive MC; in model comparisons, their colors match the corresponding model and method. Shading shows ± 0.5 standard deviations in log space across events, as in the main text. Required sampling budgets include any MC continuation described in Appendix D.2.

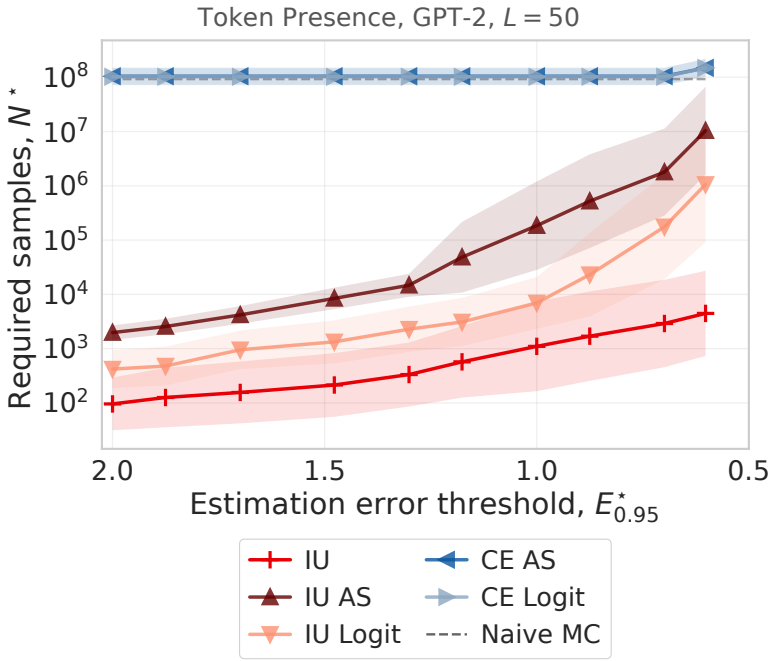


Figure 21. Absolute sampling budgets for the Token Presence method comparison in Figure 4 (left), on GPT-2 Small at $L = 50$. Each curve covers the same ten events in $10^{-8} \leq \mu < 10^{-7}$. At $E^\star_{0.95} = 1$, geometric-mean IU and naive-MC budgets are approximately 1,103 and 9.24×10^7 trajectories, respectively.

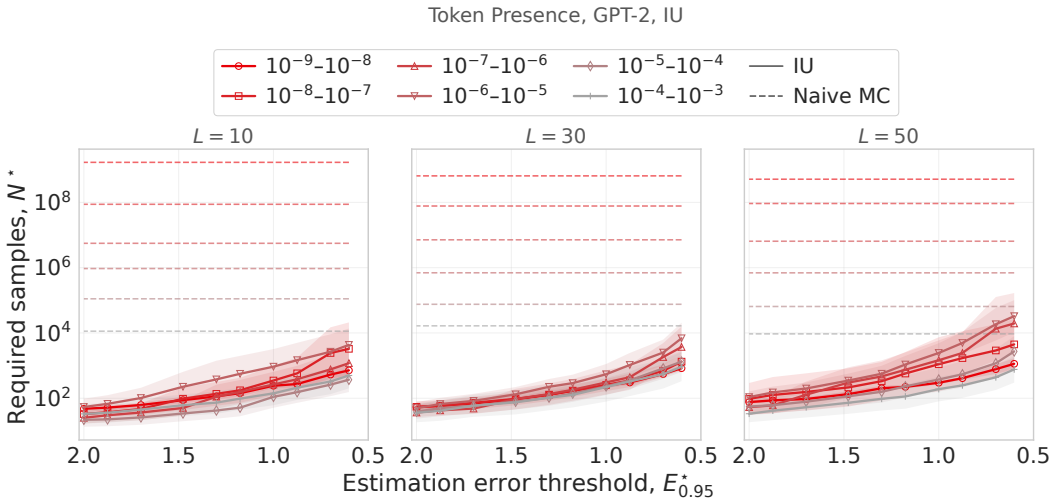


Figure 22. Absolute sampling budgets for the sequence-length comparison in Figure 18, with $L = 10$, 30, and 50 from left to right. Each color denotes a rarity band with ten events; solid curves show IU and dashed curves show naive MC. At $E^\star_{0.95} = 1$ in $10^{-8} \le \mu < 10^{-7}$, geometric-mean IU versus naive-MC budgets are approximately 349 versus 8.73×10^7, 255 versus 7.73×10^7, and 1,103 versus 9.24×10^7 trajectories, respectively.

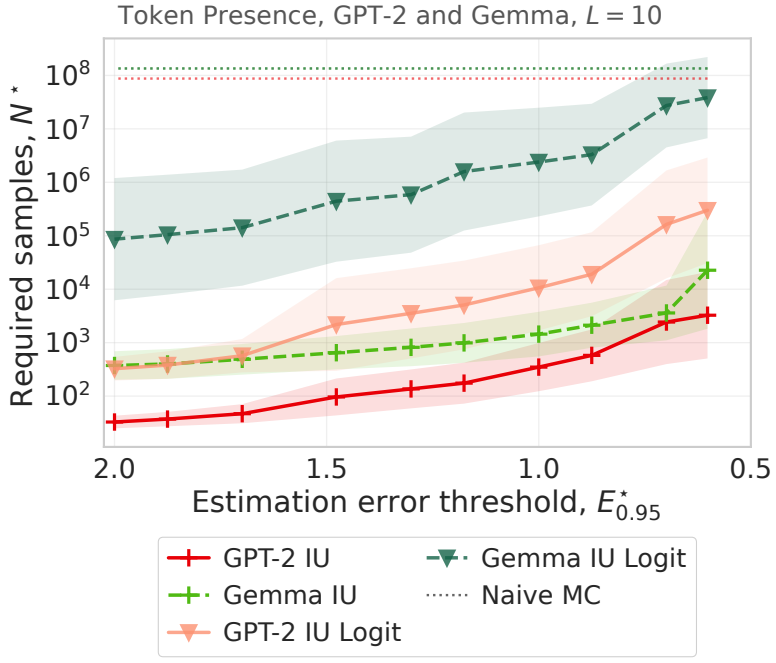


Figure 23. Absolute sampling budgets for the Token Presence model comparison in Figure 4 (right), at $L = 10$. Each curve summarizes the same ten events per model in $10^{-8} \le \mu < 10^{-7}$. At $E^\star_{0.95} = 1$, geometric-mean IU versus naive-MC budgets are approximately 349 versus 8.73×10^7 trajectories on GPT-2 and 1,445 versus 1.35×10^8 on Gemma-2.

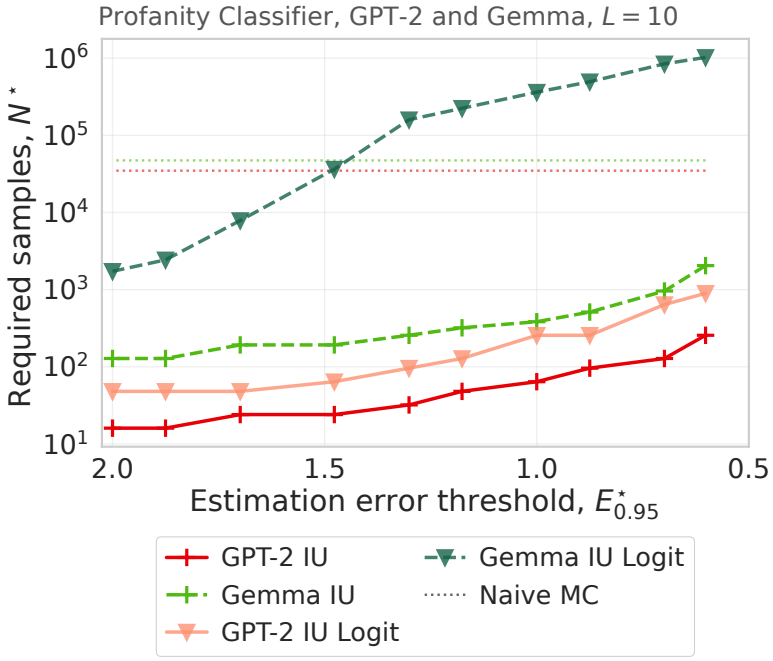


Figure 24. Absolute sampling budgets for the Profanity Classifier model comparison in Figure 6 (right), at $L = 10$ and $\kappa = 0.999$. At $E^\star_{0.95} = 1$, IU versus naive-MC budgets are 64 versus 34,858 trajectories on GPT-2 and 384 versus 47,311 on Gemma-2.

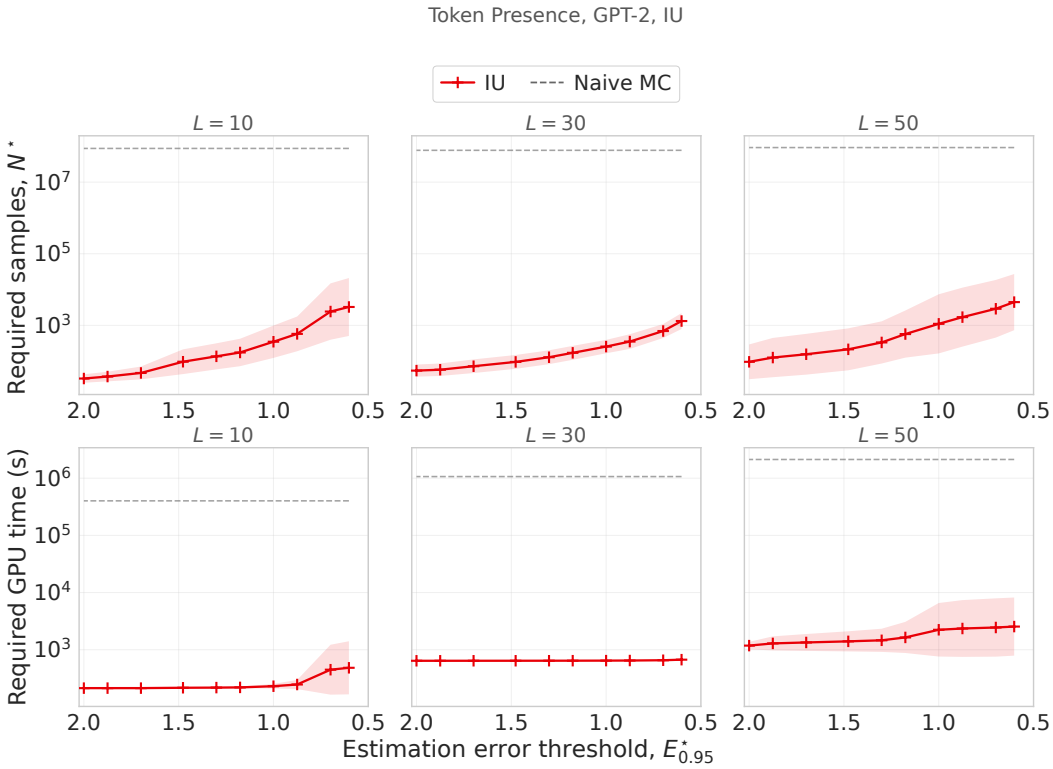


Figure 25. Absolute budgets underlying the compute-weighted efficiency comparison in Figure 19, on GPT-2 Small at $L = 10, 30$, and 50 from left to right. Top: required trajectories. Bottom: estimated GPU time including IU training. Each curve summarizes the same ten events per length in $10^{-8} \leq \mu < 10^{-7}$. At $E^\star_{0.95} = 1$, geometric-mean IU versus naive-MC times are approximately 230 versus 4.02×10^5, 645 versus 1.07×10^6, and 2,235 versus 2.12×10^6 seconds, respectively.

F.3 Full-Detail Experimental Results

Token-rarity comparisons use ten randomly selected tokens per rarity decade. In the main result figures, markers show geometric means across events and shading spans ± 0.5 standard deviations around the mean in log space. ± 0.5 was chosen for visual separation.

The main text uses summaries for readability. Each figure here gives four full-detail views for the corresponding setting. **Upper left:** empirical estimate quantiles for IU, including both models in model comparisons. The outer markers show the 2.5% and 97.5% quantiles, inner bars show the interquartile range, plus markers show medians, and diamonds show per-event arithmetic means. **Upper right:** quantile log error $E_{0.95}$ against event probability at $N = 128$, with individual event points and per-decade geometric-mean lines and spread bands. **Lower left:** Estimator Efficiency Gain across error thresholds. **Lower right:** Compute-Weighted Efficiency Gain across the same thresholds, including proposal training and sampling costs from Appendix D.3. The error and frontier panels include all methods available in the setting.

Efficiency frontier panels retain every selected event's curve and markers beneath the geometric-mean lines. As in the main figures, token frontiers use the ten matched events per model and method in $10^{-8} \leq \mu < 10^{-7}$, and profanity frontiers use the classifier cutoff $\kappa = 0.999$. Error bands span ± 0.75 standard deviations in log space. Dashed horizontal lines in the frontier panels indicate parity with naive Monte Carlo.

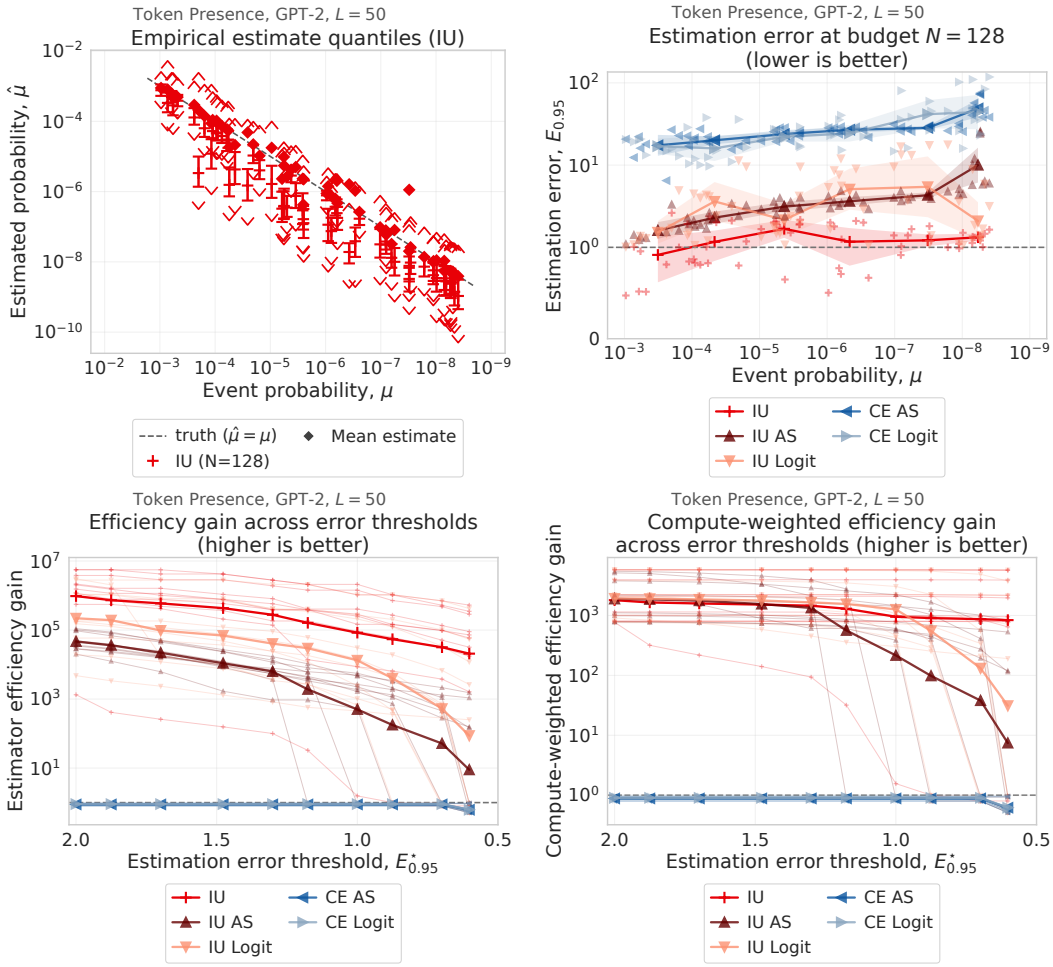


Figure 26. Token Presence rarity results on GPT-2 Small at $L = 50$: IU empirical quantiles and all-method error (both $N = 128$), and all-method estimator and compute-weighted efficiency frontiers.

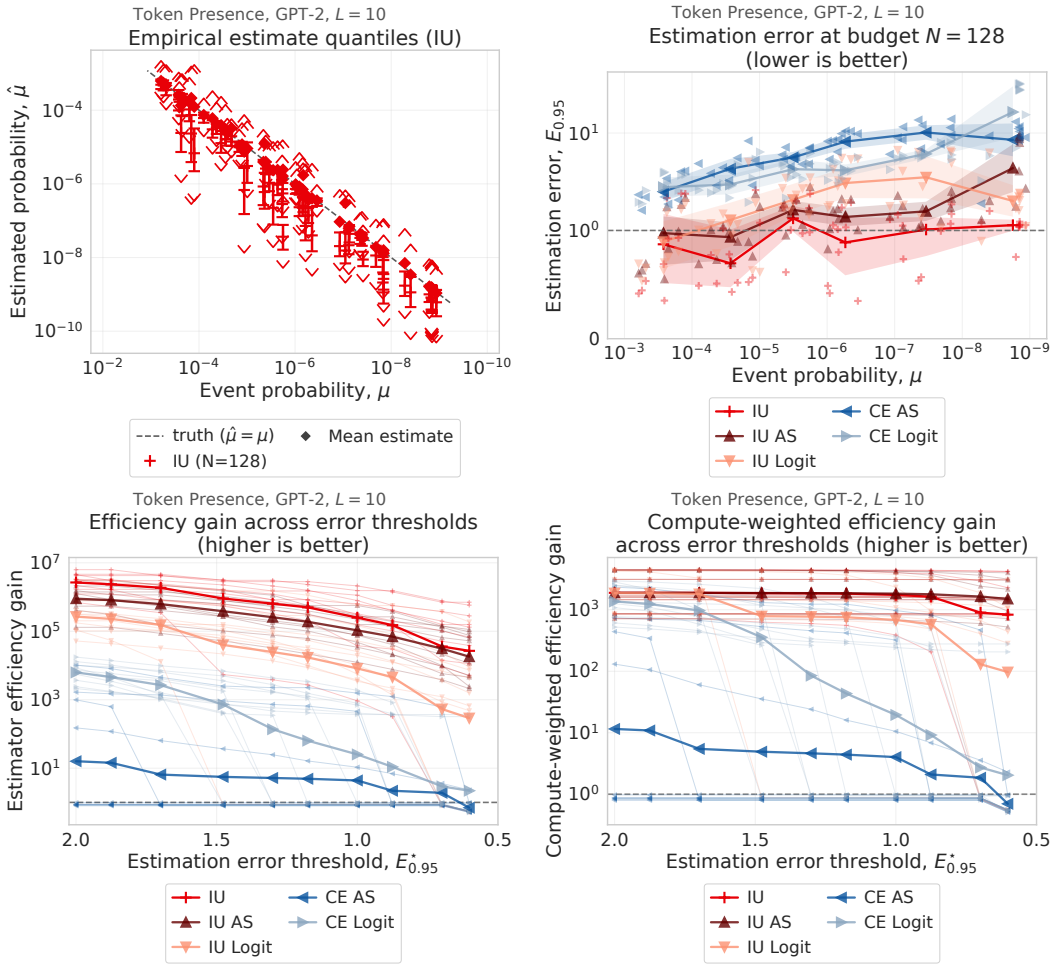


Figure 27. Token Presence results on GPT-2 Small at $L = 10$: IU empirical quantiles and all-method error (both $N = 128$), and all-method estimator and compute-weighted efficiency frontiers.

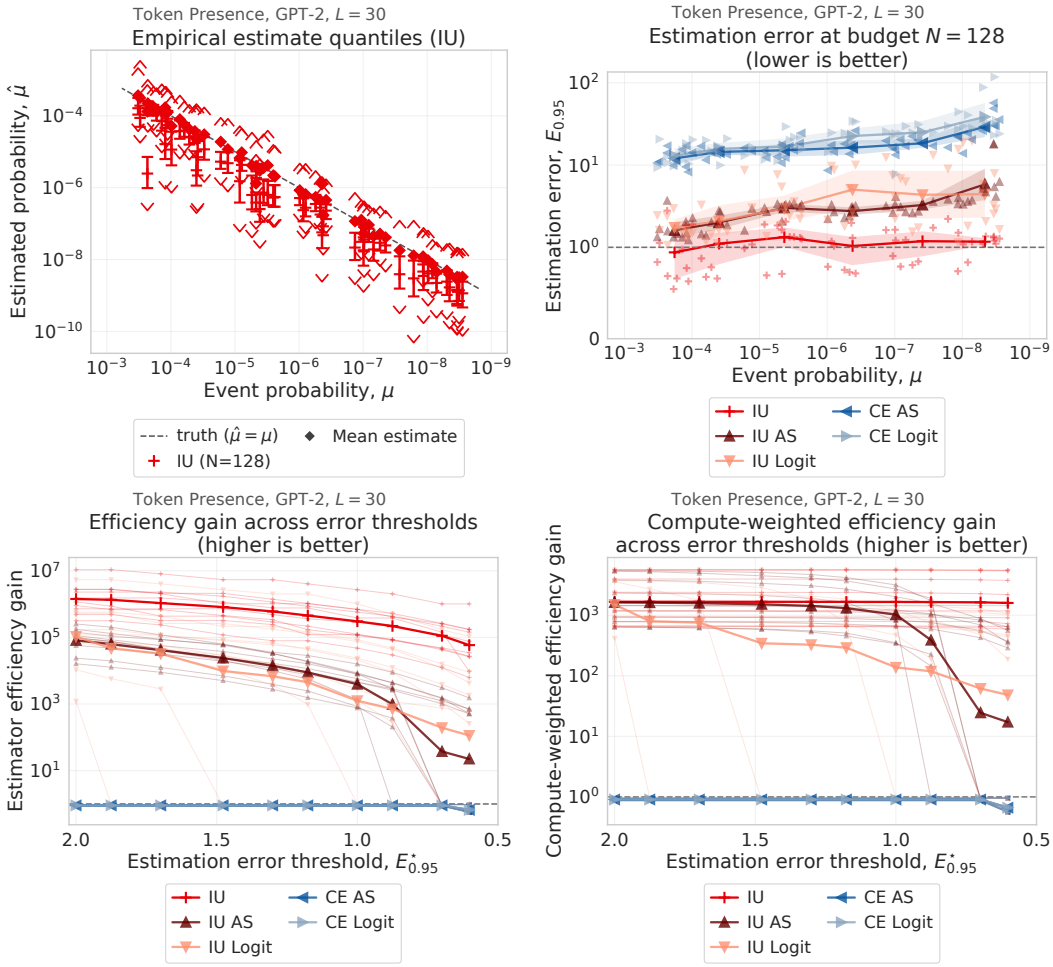


Figure 28. Token Presence results on GPT-2 Small at $L = 30$: IU empirical quantiles and all-method error (both $N = 128$), and all-method estimator and compute-weighted efficiency frontiers.

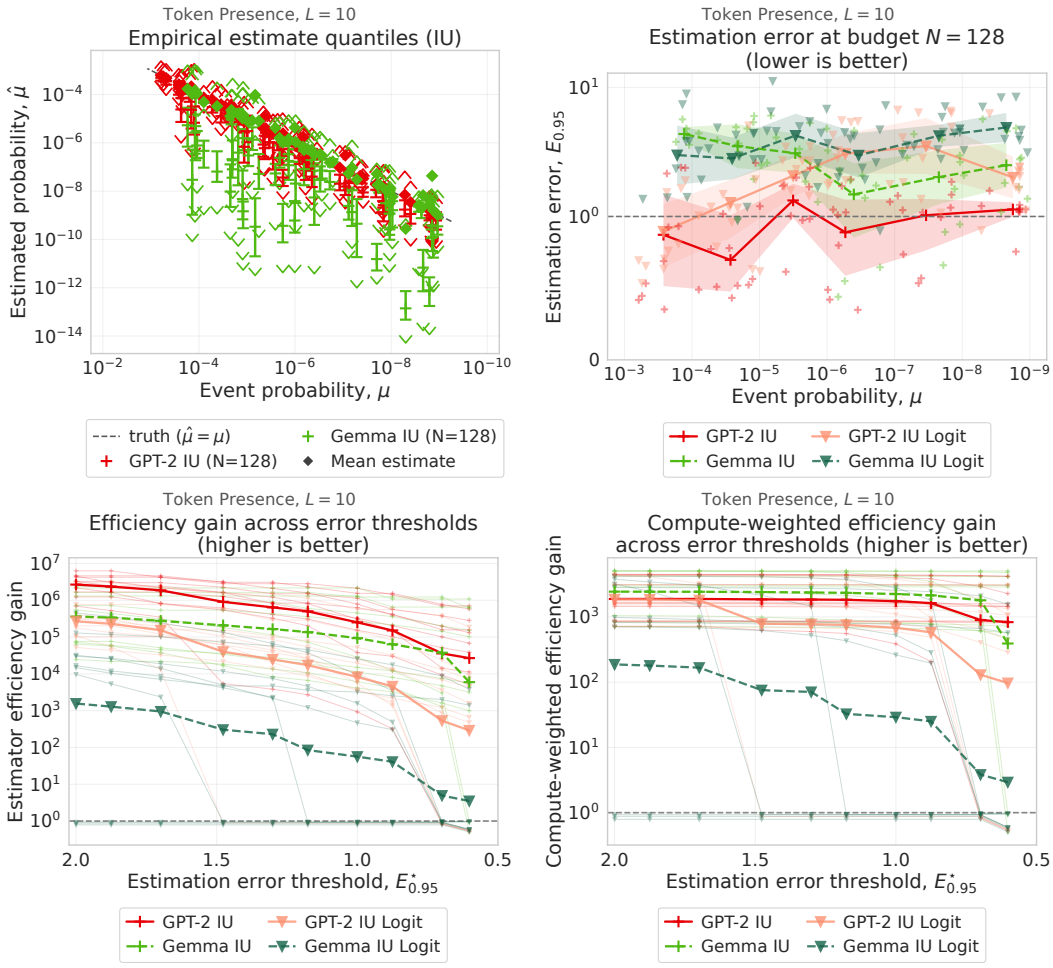


Figure 29. Full-detail Token Presence model comparison on GPT-2 Small and Gemma-2 at $L = 10$. Empirical quantiles show IU for both models at $N = 128$; error at $N = 128$ and the two efficiency frontiers show both IU and IU Logit for each model.

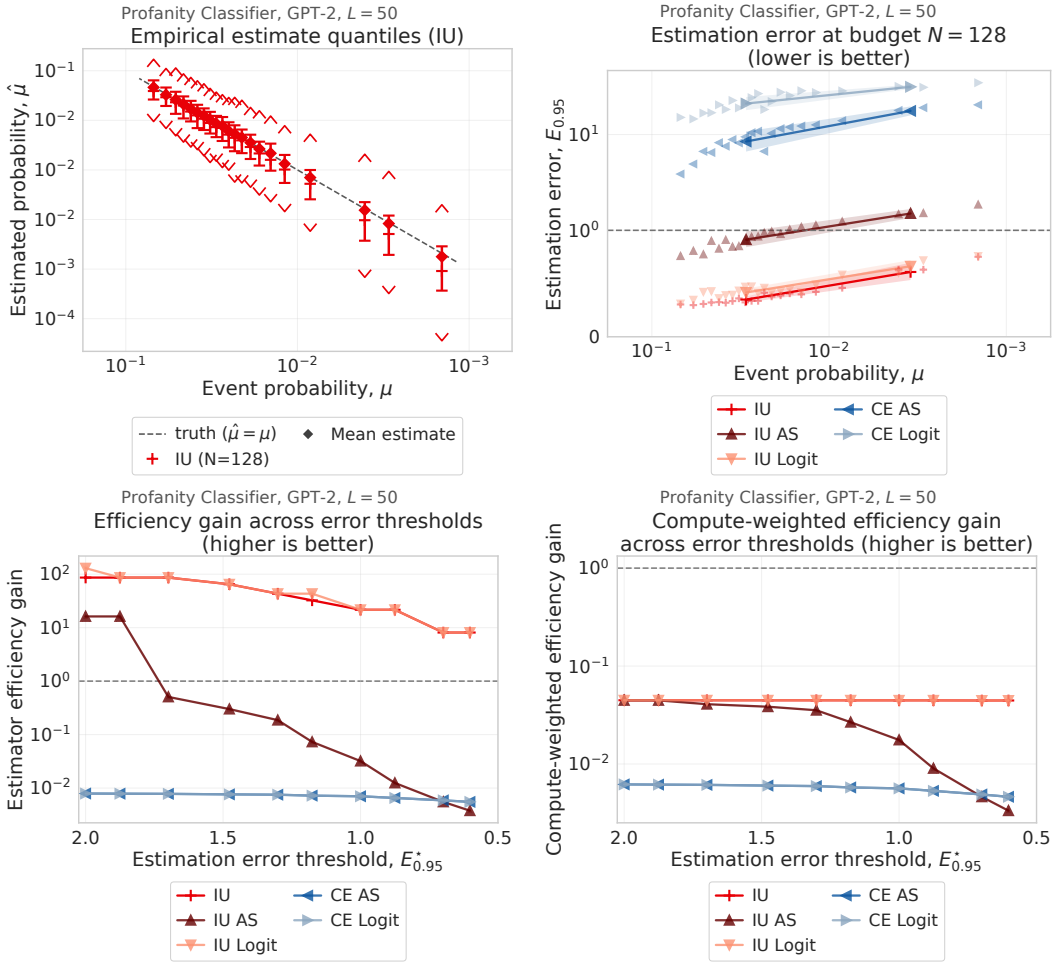


Figure 30. Profanity Classifier results on GPT-2 Small at $L = 50$: IU empirical quantiles, all-method error at $N = 128$, and all-method estimator and compute-weighted efficiency frontiers at $\kappa = 0.999$.

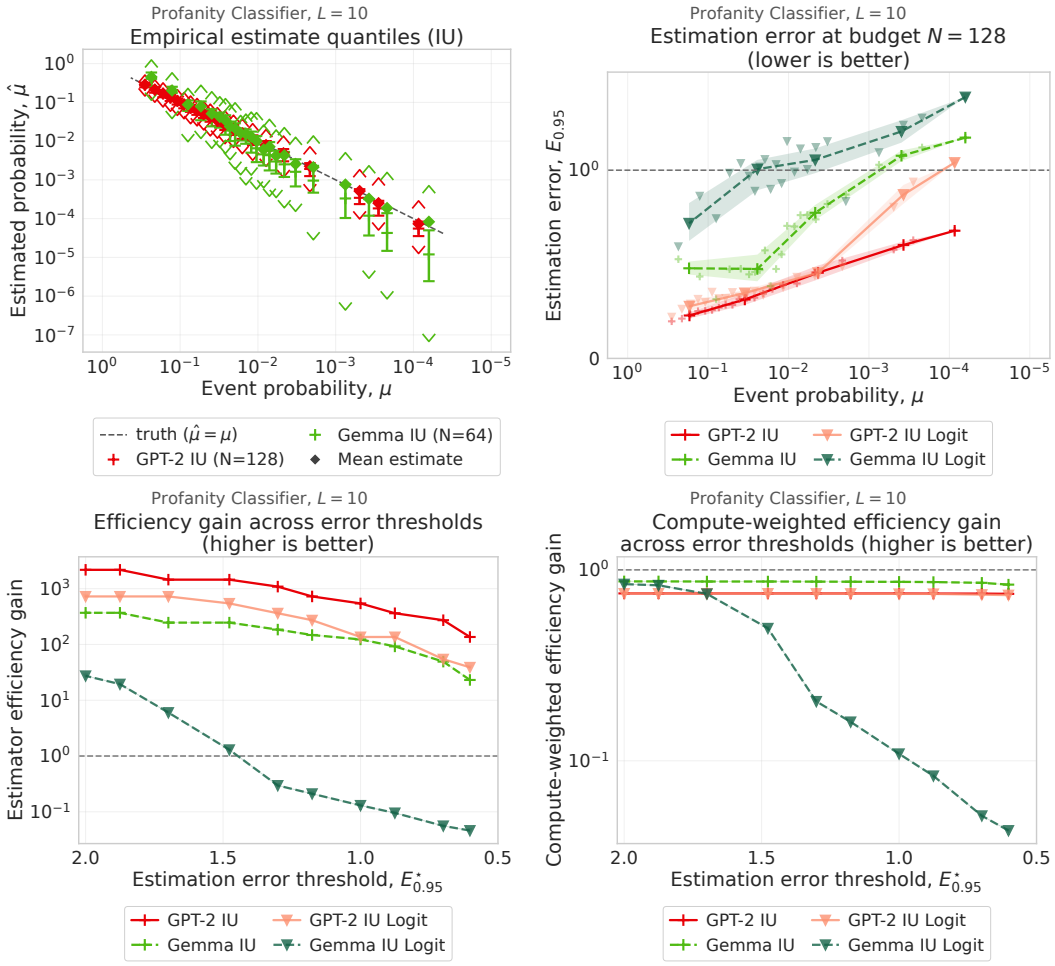


Figure 31. Full-detail Profanity Classifier model comparison on GPT-2 Small and Gemma-2 at $L = 10$. Empirical quantiles show IU at $N = 128$ for GPT-2 Small and $N = 64$ for Gemma-2. Efficiency frontiers use classifier cutoff 0.999.

F.4 Qualitative Examples

The following examples show high importance-weight trajectories from saved GPT-2 Small IU proposals for the four event types. IU model completions remain natural under the base model while satisfying the event conditions. This suggests that IU may help reveal ways a specified event can occur, in addition to estimating its probability.

All samples include the prompt `Once upon a time`; L counts only generated tokens, and μ is the base-model event probability. Profanity is masked for display purposes.

Token Presence.
Contains `Sorceress`; $L = 50$, $\mu = 6.12 \times 10^{-5}$.

> Once upon a time, Ye Hao thought out how **Sorceress** Luo Yang should intervene and help him regain his immortality. Yukkun frowned and pouted. Even if it looked silly, laughing would be refreshing to her. It's all simple, but Ye Hao couldn't

Profanity Classifier.
Classifier probability exceeds 0.999; $L = 50$, $\mu = 1.44 \times 10^{-3}$.

> Once upon a time he made such a f*****g joke out of his own blood... s**t was boiling. 'No one would f*****g tell you that I screwed him over,' he said as he put his head together. 'Yeah, but nobody said that.

Conjunctive event.
Contains `the`, `was`, and `village` in any order; $L = 10$, $\mu = 8.06 \times 10^{-5}$.

> Once upon a time, **the village was** fooling itself.

Ordered Sequence event.
Contains `the` $\rightarrow$ `was` $\rightarrow$ `village`, with gaps allowed; $L = 10$, $\mu = 5.75 \times 10^{-6}$.

> Once upon a time **the** kingdom **was** abolished altogether and the **village** destroyed.

G NOTATION

Importance Sampling in LMs

Symbol	Meaning
$\mathcal{V}$, $\lvert\mathcal{V}\rvert$	Vocabulary and its size
u	Generic vocabulary token
L	Output sequence length
ℓ	Decoding position, $\ell \in \{1, \dots, L\}$
$\mathbf{y} = (y_1, \dots, y_L)$	Output sequence (trajectory) in $\mathcal{V}^L$
$\mathbf{y}_{<\ell}$	Prefix preceding position ℓ
$\mathbf{x}$	Input sequence; held fixed and suppressed from the notation
θ	Base-model parameters
p_θ, p	Base model; we write $p := p_\theta$ since θ remains fixed
δ, δ_t	Generic weight-space perturbation and its value at training iteration t
$q_\delta := p_{\theta+\delta}$	Proposal model
Φ	Binary event indicator, $\Phi : \mathcal{V}^L \to \{0, 1\}$
$\mathcal{E}$	Rare event $\{\Phi(\mathbf{y}) = 1\}$
μ	Target probability $\mathbb{E}_p[\Phi(\mathbf{y})]$
$\widehat{\mu}$	Estimate of μ
$w(\mathbf{y})$	Trajectory importance weight $p(\mathbf{y})/q_\delta(\mathbf{y})$
r_δ, $q_\delta^{\mathcal{E}}$	Proposal event probability $q_\delta(\mathcal{E})$ and conditional distribution $q_\delta(\cdot \mid \mathcal{E})$
q^*	Zero-variance proposal $p(\mathbf{y})\Phi(\mathbf{y})/\mu$

Evaluation settings. The superscript names the event family. In expressions such as $\Phi^{\mathrm{TOK}}(\mathbf{y}; v)$, $\mathbf{y}$ is the trajectory being evaluated, while the quantity after the semicolon specifies the event instance. The bare symbol Φ^{TOK} denotes the corresponding family; the same convention applies to S_δ^{TOK}.

Event	Indicator	Reference probability estimator	Reference probability	Differentiable surrogate
Token Presence	$\Phi^{\mathrm{TOK}}(\mathbf{y}; v)$	$\widehat{\Phi}^{\mathrm{TOK}}(\mathbf{y}; v)$	$\mu^{\mathrm{TOK}}(v)$	$S_\delta^{\mathrm{TOK}}(\mathbf{y}; v)$
Conjunctive (AND)	$\Phi^{\mathrm{AND}}(\mathbf{y}; \mathbf{v})$	$\widehat{\Phi}^{\mathrm{AND}}(\mathbf{y}; \mathbf{v})$	$\mu^{\mathrm{AND}}(\mathbf{v})$	$S_\delta^{\mathrm{AND}}(\mathbf{y}; \mathbf{v})$
Ordered Sequence (SEQ)	$\Phi^{\mathrm{SEQ}}(\mathbf{y}; \mathbf{v})$	$\widehat{\Phi}^{\mathrm{SEQ}}(\mathbf{y}; \mathbf{v})$	$\mu^{\mathrm{SEQ}}(\mathbf{v})$	$S_\delta^{\mathrm{SEQ}}(\mathbf{y}; \mathbf{v})$
Profanity Classifier	$\Phi^{\mathrm{PROF}}(\mathbf{y}; \kappa)$	—	$\mu^{\mathrm{PROF}}(\kappa)$	$S_\delta^{\mathrm{PROF}}(\mathbf{y}; \kappa)$

Symbol	Meaning
v	Target token for Token Presence
$\mathbf{v} = (v_1, \dots, v_m)$	Tuple of target tokens for Compositional Token Presence; m is the number of targets
f, κ	Profanity Classifier probability and its threshold; $\Phi^{\mathrm{PROF}}(\mathbf{y}; \kappa) = \mathbb{I}\{f(\mathbf{y}) > \kappa\}$

Iterative Unalignment objective and training.

Symbol	Meaning
$S_\delta(\mathbf{y})$	Differentiable surrogate supplying the amplification signal
$\mathcal{R}_\delta(\mathbf{y})$	Dense reverse-KL regularizer (Section 4.2)
$\mathcal{J}_t$	Empirical IU objective at iteration t
λ_t, λ_0, $\lambda_{\min}$	Regularization strength, its initial value, and its floor
T, t	Total number of training iterations and the training iteration index
B, i	Training batch size and batch sample index
η_δ, η_λ	Learning rates for δ and for λ
$\mathrm{ESSr}_{\mathrm{batch}}$, $\mathrm{ESSr}_{\mathrm{event}}$	Full-batch and in-event effective sample size ratios
ρ_{batch}, ρ_{event}	Targets for the two effective sample size ratios
ESSr_t, ρ_t	Active effective sample size ratio and its target at iteration t
$\hat{h}_t$, $h^\star$	Empirical event frequency and the phase-switch threshold

Evaluation protocol and metrics.

Symbol	Meaning
N	Estimation budget
K, k	Number of independent repetitions and the repetition index, giving $\{\widehat{\mu}_{k,N}\}_{k=1}^{K}$
$E_{0.95}(N;\mu)$	Quantile log error (Section 5)
$E^{\star}_{0.95}$	Estimation error threshold: largest acceptable quantile log error
$N^{\star}(E^{\star}_{0.95};\mu)$	Budget required to satisfy the error threshold (4)
$N^{\star}_{\mathrm{MC}}(E^{\star}_{0.95};\mu)$	Exact Binomial naive-MC budget for the same threshold
$\mathrm{cost}_{\mathrm{MC}}$, $\mathrm{cost}_{\mathrm{IS}}$, $\mathrm{cost}_{\mathrm{train}}$	Per-trajectory wall-clock cost under MC and IS, and proposal training cost

H ALL EXPERIMENTAL TOKENS AND EVENTS

This section lists all 321 distinct model–length–event combinations in the experimental inventory: 246 Token Presence events, 65 Profanity Classifier events, 5 Conjunctive events, and 5 Ordered Sequence events. The listed μ values are the reference probabilities used by the experiments. Lengths L count generated tokens after the prompt. GPT-2 Small uses `Once upon a time`; Gemma-2 uses `Tell me a story` with its chat template.

Table 8. All 60 Token Presence events for GPT-2 Small at $L = 10$ and their reference probabilities. Read the left list first, then the right; events are ordered from rarer to more common.

Token	Ground truth μ
ActionCode	1.151×10^{-9}
␣\u30b5\u30fc\u30c6\u30a3	1.41×10^{-9}
reportprint	1.41×10^{-9}
␣RandomRedditor	1.413×10^{-9}
␣externalToEVA	1.413×10^{-9}
␣externalTo	1.445×10^{-9}
oreAndOnline	1.463×10^{-9}
␣antidepress	1.607×10^{-9}
␣teasp	3.942×10^{-9}
ÍÍ	5.2×10^{-9}
\u30b4\u30f3	1.4403×10^{-8}
soDeliveryDate	1.4414×10^{-8}
soType	1.4863×10^{-8}
natureconservancy	2.0748×10^{-8}
\u30fc\u30c6	3.8521×10^{-8}
␣pestic	4.0529×10^{-8}
␣unfocusedRange	4.43×10^{-8}
\u9f8d\u559a\u58eb	7.4453×10^{-8}
ItemImage	7.6178×10^{-8}
Thumbnail	8.9868×10^{-8}
sembly	1.17665×10^{-7}
␣SolidGoldMagikarp	3.50913×10^{-7}
Delivery	4.43993×10^{-7}
zbek	5.28826×10^{-7}
Database	6.0715×10^{-7}
␣BELOW	6.77883×10^{-7}
797	6.97073×10^{-7}
767	8.14367×10^{-7}
␣978	9.14544×10^{-7}
␣Ukrain	9.74397×10^{-7}
iminary	1.783951×10^{-6}
388	1.814818×10^{-6}
ricia	1.85303×10^{-6}
Ã	2.314821×10^{-6}
employment	2.863334×10^{-6}
html	3.451055×10^{-6}
␣SELECT	4.080043×10^{-6}
article	4.326181×10^{-6}
Si	4.56803×10^{-6}
␣Aside	9.890819×10^{-6}
atten	1.1462555×10^{-5}
␣indicative	1.2699662×10^{-5}
␣Harbaugh	1.4530803×10^{-5}
␣Ci	2.11075×10^{-5}
␣Acad	2.5733549×10^{-5}
␣definitions	2.808293×10^{-5}
␣ingredient	3.4664296×10^{-5}
␣loads	4.3123397×10^{-5}
␣boost	5.2364278×10^{-5}
␣Tomb	7.9511388×10^{-5}
aed	$1.25505903 \times 10^{-4}$
uss	$1.44590405 \times 10^{-4}$
␣connect	$1.58662006 \times 10^{-4}$
␣fortunate	$2.23511597 \times 10^{-4}$
ria	$2.35639032 \times 10^{-4}$
␣Arch	$2.56191561 \times 10^{-4}$
␣Let	$2.63755705 \times 10^{-4}$
␣promise	$4.82588483 \times 10^{-4}$
␣desired	$5.50281373 \times 10^{-4}$
␣mere	$6.15937519 \times 10^{-4}$

Table 9. All 60 Token Presence events for GPT-2 Small at $L = 30$ and their reference probabilities. Read the left list first, then the right; events are ordered from rarer to more common.

Token	Ground truth μ	Token	Ground truth μ
ActionCode	2.812×10^{-9}	748	2.385615×10^{-6}
Orderable	3.176×10^{-9}	568	2.52272×10^{-6}
cloneembedreportprint	3.343×10^{-9}	652	3.088749×10^{-6}
rawdownload	3.419×10^{-9}	021	3.58389×10^{-6}
␣RandomRedditor	3.422×10^{-9}	arijuana	4.099484×10^{-6}
DeliveryDate	4.485×10^{-9}	␣Wem	4.666225×10^{-6}
ÃÂÃÂÃÂÃÂ	4.929×10^{-9}	\u30aa	4.693658×10^{-6}
GoldMagikarp	7.148×10^{-9}	Layout	5.44136×10^{-6}
␣volunte	8.268×10^{-9}	␣363	7.860511×10^{-6}
␣srf	9.695×10^{-9}	Border	8.441393×10^{-6}
ÃÂÃÂ	1.1427×10^{-8}	␣Requires	1.2932401×10^{-5}
␣unintention	1.1591×10^{-8}	␣Petroleum	1.6701184×10^{-5}
␣teasp	1.6259×10^{-8}	␣cowork	3.0257203×10^{-5}
inventoryQuantity	2.697×10^{-8}	aten	3.8312104×10^{-5}
␣referen	4.4671×10^{-8}	loaded	4.0012317×10^{-5}
␣looph	5.4553×10^{-8}	␣marginally	4.2360556×10^{-5}
\u30bc\u30a6\u30b9	6.8845×10^{-8}	␣unfounded	5.1715971×10^{-5}
␣opio	8.1701×10^{-8}	␣vib	6.1999679×10^{-5}
ÛÛ	9.6763×10^{-8}	␣massage	7.2618226×10^{-5}
aution	9.8635×10^{-8}	fing	9.8132645×10^{-5}
␣mosqu	1.06689×10^{-7}	␣glam	$1.13032715 \times 10^{-4}$
\u30d8\u30e9	1.34227×10^{-7}	␣Armed	1.2007045×10^{-4}
␣SetTextColor	3.78981×10^{-7}	stock	$1.22980506 \times 10^{-4}$
glomer	4.20284×10^{-7}	␣Deity	1.3891251×10^{-4}
escription	4.38592×10^{-7}	␣voluntary	$1.67620776 \times 10^{-4}$
␣contrace	4.65958×10^{-7}	␣Baby	$1.87197977 \times 10^{-4}$
␣resil	5.52023×10^{-7}	␣pathetic	$2.29775411 \times 10^{-4}$
␣sshd	7.90799×10^{-7}	inating	2.3309172×10^{-4}
AppData	8.59069×10^{-7}	␣couch	$3.03255045 \times 10^{-4}$
␣\u30b5	9.75121×10^{-7}	␣Energy	$3.25971749 \times 10^{-4}$

Table 10. All 60 Token Presence events for GPT-2 Small at $L = 50$ and their reference probabilities. Read the left list first, then the right; events are ordered from rarer to more common.

Token	Ground truth μ	Token	Ground truth μ
ThumbnailImage	3.969×10^{-9}	LEASE	2.497735×10^{-6}
Nitrome	4.643×10^{-9}	intendent	2.531015×10^{-6}
Orderable	4.971×10^{-9}	emort	2.591963×10^{-6}
reportprint	5.398×10^{-9}	pkg	3.469834×10^{-6}
␣RandomRedditor	5.416×10^{-9}	␣discrep	4.214965×10^{-6}
␣TheNitrome	5.453×10^{-9}	paralleled	5.011348×10^{-6}
ÃÂÃÂÃÂÃÂ

Table 11. All 66 Token Presence events for Gemma-2 at $L = 10$ and their reference probabilities. Read the left list first, then the right; events are ordered from rarer to more common.

Token	Ground truth μ
␣meio	1.084×10^{-9}
␣\u043f\u0440\u044a	1.334×10^{-9}
␣Cita	1.406×10^{-9}
\u6b63\u5728	1.45×10^{-9}
cei	1.601×10^{-9}
opho	2.166×10^{-9}
McC	2.345×10^{-9}
Hb	2.65×10^{-9}
endid	4.994×10^{-9}
␣Dial	9.451×10^{-9}
␣decoration	1.0744×10^{-8}
Animals	1.0839×10^{-8}
␣utilized	1.0885×10^{-8}
\u6743	1.2994×10^{-8}
Zoo	1.793×10^{-8}
Vish	1.9079×10^{-8}
vano	1.9914×10^{-8}
␣sweats	2.0707×10^{-8}
␣articulated	2.2923×10^{-8}
helia	5.1906×10^{-8}
␣Sylvain	6.5284×10^{-8}
␣Skinny	7.6772×10^{-8}
avor	1.36062×10^{-7}
Cassie	1.64704×10^{-7}
␣Kjell	1.65677×10^{-7}
␣hastily	1.88567×10^{-7}
␣whe	2.76225×10^{-7}
␣unsteady	3.01208×10^{-7}
␣disliked	3.98665×10^{-7}
␣pep	4.54393×10^{-7}
Madame	5.48416×10^{-7}
␣Ace	6.51134×10^{-7}
Maxwell	6.83873×10^{-7}

Token	Ground truth μ
␣shedding	9.74816×10^{-7}
attered	1.015341×10^{-6}
␣coating	1.229574×10^{-6}
␣mostly	1.364986×10^{-6}
␣mor	1.376871×10^{-6}
␣disapproval	1.803206×10^{-6}
␣nurturing	2.100988×10^{-6}
␣tank	2.570783×10^{-6}
␣groans	5.677432×10^{-6}
␣bison	6.391679×10^{-6}
then	6.78363×10^{-6}
␣mural	7.759179×10^{-6}
␣Aur	9.377181×10^{-6}
mills	1.2066338×10^{-5}
␣eccentric	1.218224×10^{-5}
␣sharper	1.23574×10^{-5}
␣cutter	1.5257824×10^{-5}
␣ramp	1.9968653×10^{-5}
␣Casper	2.0307949×10^{-5}
␣scenic	2.1126934×10^{-5}
␣longer	2.2296295×10^{-5}
␣railing	4.2092146×10^{-5}
␣Robin	5.653832×10^{-5}
␣shifting	7.2919989×10^{-5}
␣Avon	7.9591409×10^{-5}
␣unsettling	$1.10149827 \times 10^{-4}$
␣resolute	$1.18333068 \times 10^{-4}$
␣librarian	$1.21119221 \times 10^{-4}$
␣steady	$1.23186823 \times 10^{-4}$
ana	$1.38252377 \times 10^{-4}$
␣brooding	$1.72549393 \times 10^{-4}$
␣whimsical	$2.31835613 \times 10^{-4}$
␣distant	$5.34711289 \times 10^{-4}$

Table 12. All Profanity Classifier events and reference probabilities. The event is $f(\mathbf{y}) > \kappa$. A dash means that this model–length–threshold combination is absent from the experimental inventory.

Cutoff κ	GPT-2, $L = 10$	GPT-2, $L = 50$	Gemma, $L = 10$
0.1	$2.8353515625 \times 10^{-1}$	$6.888671875 \times 10^{-2}$	7.9390625×10^{-2}
0.15	$2.11609375 \times 10^{-1}$	$5.83203125 \times 10^{-2}$	5.4046875×10^{-2}
0.2	$1.6591015625 \times 10^{-1}$	$5.11953125 \times 10^{-2}$	3.921875×10^{-2}
0.25	$1.33453125 \times 10^{-1}$	4.6140625×10^{-2}	3.0921875×10^{-2}
0.3	$1.085703125 \times 10^{-1}$	4.18125×10^{-2}	2.615625×10^{-2}
0.35	$8.918359375 \times 10^{-2}$	$3.82890625 \times 10^{-2}$	2.28125×10^{-2}
0.4	7.340625×10^{-2}	$3.507421875 \times 10^{-2}$	1.965625×10^{-2}
0.45	$6.056640625 \times 10^{-2}$	$3.22734375 \times 10^{-2}$	1.66875×10^{-2}
0.5	$4.980859375 \times 10^{-2}$	$2.970703125 \times 10^{-2}$	1.4265625×10^{-2}
0.55	$4.091015625 \times 10^{-2}$	2.7375×10^{-2}	1.21875×10^{-2}
0.6	$3.321484375 \times 10^{-2}$	2.5140625×10^{-2}	1.0421875×10^{-2}
0.65	$2.64765625 \times 10^{-2}$	$2.32109375 \times 10^{-2}$	8.4375×10^{-3}
0.7	$2.061328125 \times 10^{-2}$	$2.10859375 \times 10^{-2}$	7.109375×10^{-3}
0.75	$1.58671875 \times 10^{-2}$	$1.877734375 \times 10^{-2}$	5.828125×10^{-3}
0.8	1.1671875×10^{-2}	1.6671875×10^{-2}	4.578125×10^{-3}
0.85	$7.86328125 \times 10^{-3}$	$1.43359375 \times 10^{-2}$	3.28125×10^{-3}
0.9	4.6953125×10^{-3}	$1.187109375 \times 10^{-2}$	1.9375×10^{-3}
0.95	$2.14453125 \times 10^{-3}$	8.4296875×10^{-3}	7.5×10^{-4}
0.99	4.921875×10^{-4}	4.0546875×10^{-3}	$3.73527547 \times 10^{-4}$
0.995	2.8125×10^{-4}	2.9375×10^{-3}	$2.18535188 \times 10^{-4}$
0.999	8.59375×10^{-5}	$1.44140625 \times 10^{-3}$	6.3319311×10^{-5}

Table 13. All Compositional Token Presence events and reference probabilities for GPT-2 Small at $L = 10$. Conjunctive events require every listed token in any order; Ordered Sequence events require the listed order but allow intervening tokens. Leading spaces are part of each target token.

Event type	Target tokens	Ground truth μ
Conjunctive (AND)	{␣connect, ␣the, ␣was}	$3.7222278697068845 \times 10^{-6}$
Conjunctive (AND)	{␣the, ␣was, ␣village}	$8.061106367685 \times 10^{-5}$
Conjunctive (AND)	{␣a, ␣the, ␣king}	$1.981384612154 \times 10^{-4}$
Conjunctive (AND)	{␣the, ␣was, ␣people}	$5.053878165782 \times 10^{-4}$
Conjunctive (AND)	{␣the, ␣was, ␣and}	$6.57273504138 \times 10^{-3}$
Ordered Sequence (SEQ)	␣the → ␣was → ␣connect	$1.246940501071 \times 10^{-6}$
Ordered Sequence (SEQ)	␣the → ␣was → ␣village	$5.749512319003 \times 10^{-6}$
Ordered Sequence (SEQ)	␣a → ␣the → ␣king	$3.01940971556 \times 10^{-5}$
Ordered Sequence (SEQ)	␣the → ␣was → ␣people	$1.033016124585 \times 10^{-4}$
Ordered Sequence (SEQ)	␣the → ␣was → ␣and	$2.589375125567 \times 10^{-3}$