Article

CFS-eSAFE: Cross-Fitted Fixed-Sequence E-Value Gatekeeping for Safer Iterative Prompt Optimization

Firstname Lastname [1], Firstname Lastname [2] and Firstname Lastname [2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)

Abstract

Iterative prompt editing can improve a language model's behavior without weight updates, but deployment often requires evidence that each accepted update does not introduce new failures on inputs that matter most. This is difficult because two forms of adaptivity are unavoidable: candidate prompts are generated after inspecting failures, and engineers then discover multiple high-dimensional "stress slices" only after observing behavior. Naively certifying many slices with multiplicity corrections across candidates, iterations, and slices can destroy statistical power under small audits. We propose CFS-eSAFE, a training-free prompt optimizer that enforces strict data-role separation via cross-fitting, mines and orders an explicit list of the most regressive discovered slices, and certifies non-regression using anytime-valid mixture e-values in a fixed-sequence (gatekeeping) design, combined with alpha-wealth to control risk across repeated candidate tests. We implement an open-model pipeline for constrained sentiment-reversal rewriting and compare to a Bonferroni-gated e-value baseline. In the executed runs, however, the optimizer evaluated zero candidate prompts and achieved zero constrained accuracy across domains for both methods, preventing empirical validation. We report these outcomes transparently, analyze what parts of the pipeline were not exercised, and identify concrete prerequisites required for future evaluations of the proposed safety contract.

Keywords: keyword 1; keyword 2; keyword 3 (List three to ten pertinent keywords specific to the article; yet reasonably common within the subject discipline.)

1. Introduction

Prompt-based test-time optimization seeks to improve a model's behavior without changing its parameters, enabling rapid iteration and avoiding training instabilities. A representative approach is SELF-REFINE, which alternates between generating feedback and revising an output using the same model, demonstrating gains on diverse tasks including sentiment reversal [**?**]. While such methods are attractive for practical iteration, many deployments require more than improved average performance: they require a non-regression contract that rules out harmful degradations on inputs that are operationally important, safety-critical, or likely to be scrutinized.

This paper studies a safety gap that becomes acute when iterative prompt optimization is deployed in realistic engineering workflows. Two adaptivity sources interact in a way that breaks common evaluation and gating patterns.

First, stress-slice discovery is inevitable and high-dimensional. After observing a model's behavior, practitioners (or automated slice-mining tools) discover multiple failure

Received:
Revised:
Accepted:
Published:

pockets defined by combinations of simple features such as input length, the presence of negation, entity density, or the presence of socially salient identity terms. Importantly, regressions need not concentrate in a single "worst" slice: a prompt update can improve one region while degrading another, and multiple distinct pockets can co-exist.

Second, multiplicity becomes power-killing when many slices and many candidates are tested. Iterative prompt optimization evaluates a stream of candidate prompts that are themselves chosen after inspecting failures. If each candidate must be certified against a large family of slices using conservative multiplicity corrections (for example, Bonferroni across slices, across candidate attempts, and across iterations), then under a small audit budget it can become statistically infeasible to accept any update. This outcome encourages heuristic acceptance and undermines end-to-end safety.

We ask a deployment-motivated question: can we search adaptively over candidate prompt edits while maintaining an anytime-valid non-regression contract that extends beyond an overall audit average to an adaptively discovered list of stress slices, without collapsing power as the slice list grows?

We propose CFS-eSAFE (Cross-Fitted Fixed-Sequence e-value Safety), a discover-order-certify pipeline that composes three ideas into an end-to-end gate:

- **Cross-fitted data roles:** hold out an audit set that is never used to propose edits, and split that audit set into a discovery portion used to mine and order slices and a confirmation portion used only for statistical certification.
- **Anytime-valid mixture e-values on paired differences:** compare a candidate prompt to the current prompt on the same examples using paired differences, and certify improvement or non-regression with mixture e-values that remain valid under optional stopping and repeated testing.
- **Fixed-sequence (gatekeeping) certification over an ordered slice list:** rather than testing all slices simultaneously with a multiplicity split, test the discovered slices in a fixed order, requiring each gate to pass before proceeding, thereby avoiding spending error budget on slices that are never reached.

We verify the proposal by implementing the pipeline with open models on a constrained sentiment-reversal rewriting task. We also implement a baseline, AV-DR-SPO, that uses the same mixture e-value machinery but handles slice multiplicity via Bonferroni-style thresholds over a fixed slice family.

In the executed runs, both methods produce degenerate outcomes: constrained accuracy is 0.0 across all evaluation domains, and the optimizer evaluates zero candidate prompts and accepts zero updates. While these results prevent an empirical test of the intended power and safety claims, they are still informative. They indicate that, in this particular instantiation, the task and constraint calibration yields almost no successes and the candidate proposal/evaluation loop terminates before the statistical gates are exercised. Reporting these outcomes is important because it highlights prerequisites for meaningful empirical study of safety gating in training-free prompt optimization.

Contributions:

- **Adaptive prompt-optimization setting:** We formalize a deployment-oriented prompt optimization setting in which both candidate prompts and the set of audited stress slices are adaptively determined.
- **CFS-eSAFE gate design:** We introduce CFS-eSAFE, combining cross-fitted slice discovery and ordering, paired-difference evaluation, anytime-valid mixture e-values, fixed-sequence slice gatekeeping, and alpha-wealth to budget risk across an adaptive candidate search.

https://doi.org/10.3390/1010000

- **Regression certificates:** We define an interpretable regression-certificate interface that reports which safety gate fails first (optimization gate, overall audit gate, or a specific discovered slice), with the intent of making rejections actionable.
- **Transparent end-to-end implementation:** We implement an end-to-end open-model evaluation pipeline, report the executed results in full (including all generated figures), and analyze why the runs are degenerate and what must be fixed to test the core hypothesis.

Future work is primarily empirical and engineering-oriented. The constrained success definition must be calibrated so that baseline performance is non-zero; otherwise no gating method can demonstrate a safety-power tradeoff. In addition, candidate generation must reliably produce and evaluate candidate prompts so that the gatekeeping design and certificates are exercised. Once the pipeline is non-degenerate, the central scientific comparison becomes testable: whether fixed-sequence e-value gatekeeping can certify a list of adaptively discovered slices with materially higher acceptance power than Bonferroni while maintaining a valid non-regression contract under adaptive candidate testing and optional stopping.

2. Related Work

Our work is most directly connected to training-free, test-time iterative refinement and prompt-editing methods that use the model itself to critique and improve outputs. SELF-REFINE alternates between generating natural-language feedback and producing a refined output, reporting improvements across diverse tasks without additional training, reinforcement learning, or external reward models [**?**]. CFS-eSAFE shares the training-free motivation and operates in a similar "propose then refine" loop, but targets a different research problem: statistically valid deployment gating under adaptivity.

A key contrast is that SELF-REFINE is primarily an optimization procedure. It aims to produce better outputs (often by iterating several times per input) and may use task-specific selection heuristics to choose among iterations [**?**]. CFS-eSAFE is instead a safety and evaluation layer over an optimizer: it decides whether a prompt update should be accepted for future use based on evidence about non-regression. In deployment, this distinction matters because a prompt update can improve average behavior while degrading concentrated pockets of inputs that are rare overall but important for robustness auditing.

Our work also contrasts with slice testing practices that certify only (i) an overall audit average, (ii) a small predeclared list of slices, or (iii) at most a single mined "worst slice". These practices can be operationally convenient, but they do not address the realistic case where multiple distinct regressions can co-exist. CFS-eSAFE explicitly aims to certify a list of slices that are adaptively discovered from observed behavior. Because the slice list is chosen based on data, the method must prevent "double dipping" where the same audit examples are used both to select slices and to certify them. CFS-eSAFE's cross-fitted discovery/confirmation split is designed to address this selection leakage.

Finally, our work contrasts with conservative multiplicity control when many slices are audited. A natural safe-prompt-optimization baseline is to use valid sequential evidence (such as e-values) but apply Bonferroni-style corrections across slices, and possibly across candidate attempts. This can become overly conservative under small audits because the per-slice thresholds become very hard to pass as the number of tested slices grows. Our baseline AV-DR-SPO instantiates this approach: it uses mixture e-values but corrects slice multiplicity via Bonferroni-style thresholds over a fixed slice set. CFS-eSAFE differs by using a fixed-sequence (gatekeeping) design over an ordered list: later slice tests are only performed after earlier tests pass. In effect, it avoids allocating error budget to slices that would never be reached due to earlier failures.

In summary, relative to SELF-REFINE and related training-free refinement approaches [**?**], the novelty of CFS-eSAFE is not in proposing candidates but in certifying them under adaptive candidate generation and adaptive slice discovery: cross-fitted slice mining, statistically valid mixture e-values, fixed-sequence gatekeeping over multiple slices, and interpretable rejection certificates intended to support debugging and auditing.

3. Background

We consider a training-free prompt optimization loop for a fixed rewriting model. The goal is to update the instruction prompt while maintaining a deployment-oriented non-regression contract under adaptive candidate testing and adaptive stress-slice discovery.

3.1. Problem setting and constrained success

Let x denote an input text and y its gold sentiment label in $\{0, 1\}$. A rewriting model produces a rewrite $r(P, x)$ given a prompt P. We evaluate rewrites using a constrained success indicator $z(P, x) \in \{0, 1\}$ that requires three conditions simultaneously:

1. Sentiment reversal success: a fixed sentiment classifier applied to $r(P, x)$ predicts the opposite of y.
2. Semantic preservation: cosine similarity between embeddings of x and $r(P, x)$ is at least a threshold τ.
3. Fluency: a language model assigns $r(P, x)$ a negative log-likelihood (NLL) not exceeding $\text{nll}_{\max}$. In the implementation, this is sometimes described equivalently as perplexity not exceeding $\text{ppl}_{\max}$ where $\text{ppl}_{\max} = \exp(\text{nll}_{\max})$.

3.2. Paired differences for candidate comparison

At each step, we compare a candidate prompt P_c to the current prompt P on the same examples. For an example x_i, define the paired difference $d_i = z(P_c, x_i) - z(P, x_i)$. Because z is binary, $d_i \in \{-1, 0, 1\}$. Paired differences provide variance reduction because each example serves as its own control; this is especially important when audits are small.

3.3. Stress slices and slice keys

To capture concentrated failure pockets, we define a deterministic slicing function $s(x)$ that maps each input to a discrete slice key. In our implementation, the slice key is the conjunction of four text-derived features:

- a length bin (short if length is at most 8 tokens, medium if 9-18 tokens, long otherwise),
- a negation indicator based on the presence of "not" or "n't",
- an entity-density proxy based on the fraction of tokens whose first character is uppercase, thresholded at 0.18,
- an identity-term flag indicating whether the text contains one of the terms {muslim, black, gay, c

A slice is the subset of examples sharing the same slice key. The intended safety requirement is that a candidate prompt be non-regressive not only overall, but also on an adaptively discovered list of high-risk slices.

3.4. Anytime-valid evidence via e-values

We need tests that remain valid under optional stopping and repeated use across many candidate prompts. We use e-values: nonnegative random variables E such that, under the null hypothesis, $\mathbb{E}[E] \leq 1$. A common decision rule is to reject a null hypothesis at level ℓ when $E \geq 1/\ell$; validity is preserved under optional stopping.

We focus on one-sided hypotheses about the mean paired difference. Each gate tests a null of the form $H_0 : \mathbb{E}[d] \leq m$, for a margin m. In our application, $m = 0$ is used to require

improvement (no worse on average), while $m = -\delta$ is used to allow a small tolerated degradation δ in a non-regression gate.

3.5. Mixture e-values for bounded differences

Because d is bounded in $[-1, 1]$, we can construct multiplicative e-values using betting fractions. For a fixed betting fraction λ, define

$$E_\lambda = \prod_{i=1}^{n} \big(1 + \lambda\,(d_i - m)\big).$$

Using multiple λ values can improve robustness, but selecting λ adaptively based on the same data can invalidate the e-value. We therefore fix a small grid of λ values and use a mixture:

$$E = \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} E_\lambda.$$

This mixture remains a valid e-value, whereas taking the maximum over λ is generally not guaranteed to preserve validity. This detail matters because the overall optimizer tests many candidates and may stop early based on the observed evidence.

4. Method

CFS-eSAFE (Cross-Fitted Fixed-Sequence e-value Safety) is a safety-gated, training-free prompt optimization procedure. It augments a standard iterative prompt-editing loop with a certification mechanism intended to remain valid under two adaptivity sources: candidate prompts are generated after observing failures, and stress slices are mined after observing behavior.

4.1. Data roles and cross-fitting

For each random seed, data are partitioned into three disjoint roles:

- D_{opt}: an optimization batch used to identify failures and to guide candidate prompt proposals. D_{opt} provides no safety guarantee.
- A: an audit set that is never used to propose edits. A is split into A_{disc} for slice discovery and ordering and A_{conf} for statistical confirmation.

This split is fixed once per seed. For each candidate prompt P_c, slices and their order are determined using only A_{disc}, and the resulting ordered slice list is treated as fixed when testing on A_{conf}. This cross-fitting aims to prevent slice-selection leakage: the evidence used to choose which slices to test is not reused to pass those slice tests.

4.2. Gate statistics: paired differences and mixture e-values

For each example x, we compute $z(P, x)$ and $z(P_c, x)$ and then form $d = z(P_c, x) - z(P, x)$. All gates operate on collections of these differences, either overall or restricted to a slice key $s(x)$.

For a list of differences $\{d_i\}_{i=1}^{n}$ and a margin m, we compute a mixture e-value. Fix a grid $\Lambda = \{0.05, 0.1, 0.2, 0.4, 0.7\}$. For each $\lambda \in \Lambda$, compute

$$E_\lambda = \prod_{i=1}^{n} \max\{\epsilon,\, 1 + \lambda\,(d_i - m)\},$$

where $\epsilon > 0$ is a small constant used to clip factors for numerical stability. The mixture e-value is

$$E = \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} E_\lambda.$$

A gate at level ℓ passes if $E \geq 1/\ell$.

4.3. Adaptive slice mining and fixed-sequence certification

Slice mining and ordering. For each candidate prompt P_c, we compute paired differences on A_{disc} and aggregate them by slice key. Among slices with support at least $n_{\min}$, we compute the mean paired difference per slice and select the top J most regressive slices (most negative means). We then order this list from most to least regressive, yielding an ordered slice list $(s_1, \ldots, s_J)$ that is deterministic given $(P, P_c, A_{\text{disc}})$.

Fixed-sequence (gatekeeping) certification. For each candidate, CFS-eSAFE runs a fixed sequence of gates, each tested at the same level ℓ:

1. Optimization improvement gate on D_{opt}: test $H_{0,\text{opt}} : \mathbb{E}[d] \leq 0$.
2. Overall audit non-regression gate on A_{conf}: test $H_{0,\text{all}} : \mathbb{E}[d] \leq -\delta$.
3. Slice-list non-regression gates on A_{conf} restricted to each discovered slice s_j in the A_{disc} order: test $H_{0,s_j} : \mathbb{E}[d \mid s(x) = s_j] \leq -\delta$.

The acceptance rule is conjunctive: accept P_c only if every gate in the sequence passes. The fixed-sequence design is motivated by power preservation when J is large: unlike Bonferroni, the method does not divide ℓ by J.

Alpha-wealth across repeated candidate tests. Across an adaptive search over many candidates, CFS-eSAFE maintains alpha-wealth W initialized at α. For each candidate test, it spends spend $= \min(\alpha_{\text{max spend}}, W/4)$, sets $\ell =$ spend for the gates, and uses the threshold $1/\ell$. If the candidate is rejected, wealth decreases by spend. If the candidate is accepted, wealth is replenished by ρ spend (with a cap in the implementation), where $\rho \in (0, 1]$ is a reward fraction.

Regression certificates. For each rejected candidate prompt, CFS-eSAFE records the first gate that failed, chosen from $\{\text{opt}, \text{audit\_all}, \text{slice:}\langle\text{slice\_key}\rangle\}$. This interface is intended to produce actionable explanations for rejections.

[H] [1] Partition data into D_{opt}, A_{disc}, and A_{conf} Initialize prompt $P \leftarrow P_0$ and wealth $W \leftarrow \alpha$ $t = 1$ to T Propose a set of candidate prompts $\mathcal{C}_t$ using only D_{opt} $P_c \in \mathcal{C}_t$ $W \leq 0$ **break** spend $\leftarrow \min(\alpha_{\text{max spend}}, W/4)$; $\ell \leftarrow$ spend Compute paired differences on A_{disc}; mine ordered slices $(s_1, \ldots, s_J)$ Compute E_{opt} from paired differences on D_{opt} with margin $m = 0$ $E_{\text{opt}} < 1/\ell$ Record certificate fail $=$ opt $W \leftarrow W -$ spend; **continue** Compute E_{all} on A_{conf} with margin $m = -\delta$ $E_{\text{all}} < 1/\ell$ Record certificate fail $=$ audit_all $W \leftarrow W -$ spend; **continue** $j = 1$ to J Compute E_{s_j} on $\{x \in A_{\text{conf}} : s(x) = s_j\}$ with margin $m = -\delta$ $E_{s_j} < 1/\ell$ Record certificate fail $=$ slice:s_j $W \leftarrow W -$ spend; **break** all gates passed Accept update: $P \leftarrow P_c$ $W \leftarrow \min(W_{\max}, W - \text{spend} + \rho\,\text{spend})$

5. Experimental Setup

5.1. Task, goal, and models

We evaluate training-free prompt optimization on constrained sentiment-reversal rewriting. Given an input text x with sentiment label y, the rewriting model must produce a fluent rewrite whose sentiment is the opposite of y while preserving semantic content.

The rewriting model and the prompt editor are the same open model: `google/flan-t5-small`. Sentiment reversal is assessed using `distilbert-base-uncased-finetuned-sst-2-english`. Semantic similarity is computed using `sentence-transformers/all-MiniLM-L6-v2` with cosine similarity. Fluency is measured using GPT-2 negative log-likelihood (with optional conversion to perplexity).

5.2. Datasets and splits

The optimization dataset D_{opt} is a subset of SST-2 train with 120 examples. The audit pool (never used to propose edits) is formed from:

- Yelp train: 80 examples,
- IMDb train: 80 examples,
- SST-2 probe augmentations: 80 examples created by deterministic perturbations (truncation and appended negation),
- Identity-term probe augmentations: 40 examples created by inserting a neutral identity prefix such as "As a Muslim person, …".

From this pool, we sample an audit set A of size 200 and split it once into A_{disc} (100) and A_{conf} (100).

Final evaluation is performed on SST-2 validation (300 examples), Yelp test (300 examples), and IMDb test (300 examples). The evaluation code also constructs held-out probes from SST-2 validation by applying the same deterministic negation and identity insertions; these probes are used when computing worst-slice constrained accuracy.

5.3. Constrained success definition and thresholds

For a prompt P and example (x, y), we generate $r(P, x)$ with the rewriter. We set $z(P, x) = 1$ if all three conditions hold:

- the sentiment classifier predicts the opposite label,
- cosine similarity is at least $\tau_{\text{similarity}}$,
- GPT-2 NLL is at most nll_{max}.

In the configuration system, nll_{max} is represented as `pi_fluency_nll`, and ppl_{max} is computed as $\exp(\text{nll}_{\text{max}})$. Across executed runs, Optuna-selected values include (as recorded in summaries) $\tau_{\text{similarity}} \approx 0.691$ with $\text{nll}_{\text{max}} \approx 2.487$ (so $\text{ppl}_{\text{max}} \approx 12.03$) in one run, and $\tau_{\text{similarity}} \approx 0.728$ with $\text{nll}_{\text{max}} \approx 3.652$ (so $\text{ppl}_{\text{max}} \approx 38.53$) in another.

5.4. Metrics, optimization protocol, and compared methods

We report per-domain constrained accuracy for each of the three evaluation domains (SST-2, Yelp, IMDb). The primary metric used in the executed code is robust constrained accuracy defined as the minimum of the three per-domain constrained accuracies; i.e., $\min\{\text{Acc}_{\text{SST2}}, \text{Acc}_{\text{Yelp}}, \text{Acc}_{\text{IMDb}}\}$. This detail matters because it differs from a two-domain minimum described elsewhere in the surrounding documentation; in the executed runs and logged summaries, the robust objective is $\min(\text{SST2}, \text{Yelp}, \text{IMDb})$.

We also report worst-slice constrained accuracy on a combined evaluation set (evaluation domains plus held-out probes). This metric is defined as the minimum slice-wise constrained accuracy across slice keys with support at least `slice_min_support`. Finally, an OOD regression indicator is computed per seed by comparing the method's Yelp or IMDb constrained accuracy to the fixed base prompt and flagging a regression if the drop exceeds 0.01.

We compare the proposed method CFS-eSAFE to a baseline AV-DR-SPO that uses mixture e-values but applies Bonferroni-style multiplicity handling over a fixed family of slice keys during certification. Four runs are reported:

- `proposed-flan-t5-small-imdb` (CFS-eSAFE),
- `comparative-1-flan-t5-small-imdb` (AV-DR-SPO),
- `proposed-distilbert-sst2-imdb` (CFS-eSAFE),
- `comparative-1-distilbert-sst2-imdb` (AV-DR-SPO).

The "distilbert" runs set a configuration field indicating a sentiment-evaluator role, but the rewriting model remains `google/flan-t5-small` across runs.

The prompt optimization loop allows up to 3 iterations with up to 4 candidate prompts per iteration. Slice mining uses `slice_min_support`$= 12$. The proposed method mines a discovered list of top-J slices (`J_top_slices` is tuned), while the baseline tests a fixed

number of slices (`num_slices_tested` is tuned) and applies a Bonferroni-style threshold controlled by `bonferroni_family`. Global alpha-wealth is initialized at $\alpha = 0.1$, and per-candidate spend is bounded by `alpha_max_spend`. The base prompt is: "Rewrite the text to the opposite sentiment while preserving meaning, entities, and topic. Write fluent English."

Optuna is enabled with 20 trials per run. For CFS-eSAFE, the search space includes $\tau_{\text{similarity}}$, `pi_fluency_nll` (which sets nll_{max} and thus ppl_{max}), `delta_noninferiority`, `J_top_slices`, and `alpha_max_spend`. For the Bonferroni baseline, the search space includes `num_slices_tested` and `bonferroni_family`. The selected hyperparameters are recorded in run summaries, and the evaluation script logs summary statistics for robust accuracy, worst-slice accuracy, OOD regression rate, accepted steps, and candidates evaluated.

6. Results

This section reports results from the four executed runs and includes all figures produced by the evaluation script. The outcomes are degenerate across methods: all constrained accuracies are zero, and the optimization loop evaluates zero candidate prompts. We therefore focus on (i) documenting the observed metrics precisely and (ii) identifying which parts of the intended method were not exercised.

6.1. Final constrained accuracy and robust constrained accuracy

Across all four runs, the final per-domain constrained accuracies are 0.0 on SST-2, Yelp, and IMDb. Consequently, the robust constrained accuracy (defined in code as the minimum over these three domains) is 0.0 for every run. Reported standard deviations across seeds are 0.0.

The cross-run aggregation reports best proposed and best baseline robust constrained accuracy values of 0.0 and reports the relative gap as NaN because the baseline value is 0.0. Worst-slice constrained accuracy is also 0.0 (mean 0.0, standard deviation 0.0) for all runs. These aggregate outcomes are consistent with the per-domain results: if each domain accuracy is 0.0, then any robust minimum and any worst-slice minimum must also be 0.0.

6.2. Optimization dynamics and acceptance power

For all runs, the summary metrics `train/candidates_evaluated_mean`$= 0.0$ and `train/accepted_steps_mean`$= 0.0$. This implies that the optimization loop terminated without evaluating any candidate prompts, and therefore without running the proposed fixed-sequence slice gatekeeping or the baseline Bonferroni slice correction. As a result, the intended distinction between the methods (power preservation under many slices and many adaptive candidate tests) is not observable in these logs.

6.3. Confusion-matrix diagnostics

Confusion matrices logged for each domain are identical across methods and runs. On IMDb, the confusion counts are $\text{tn} = 726$, $\text{fp} = 0$, $\text{fn} = 774$, $\text{tp} = 0$. On SST-2, $\text{tn} = 758$, $\text{fp} = 7$, $\text{fn} = 728$, $\text{tp} = 7$. On Yelp, $\text{tn} = 766$, $\text{fp} = 1$, $\text{fn} = 732$, $\text{tp} = 1$.

Even where the sentiment target is sometimes met (nonzero tp on SST-2 and Yelp), constrained accuracy remains 0.0. Since constrained success requires sentiment reversal, similarity, and fluency simultaneously, these results imply that at least one of the other constraints (semantic similarity threshold or fluency threshold) fails essentially everywhere under the executed configuration.

6.4. Interpretation and limitations revealed by the executed runs

The results indicate that two prerequisite conditions for empirical validation were not met in this instantiation. First, the constrained-success definition appears unattainable

at a nontrivial rate under the selected thresholds in at least one run, notably when the fluency threshold is extremely strict (for example, $\text{ppl}_{\text{max}} \approx 12$ as implied by $\text{nll}_{\text{max}} \approx 2.487$). Second, the optimizer did not evaluate any candidates, so the statistical gates, slice discovery and ordering, and regression-certificate logging were not exercised.

Because the optimizer evaluated zero candidates, we cannot empirically compare acceptance power between fixed-sequence gatekeeping and Bonferroni correction, nor can we report a distribution of first-failed gates. The figures below should therefore be read as faithful records of the logged outputs from the executed runs rather than as evidence supporting (or refuting) the method's intended advantages.

6.5. Figures

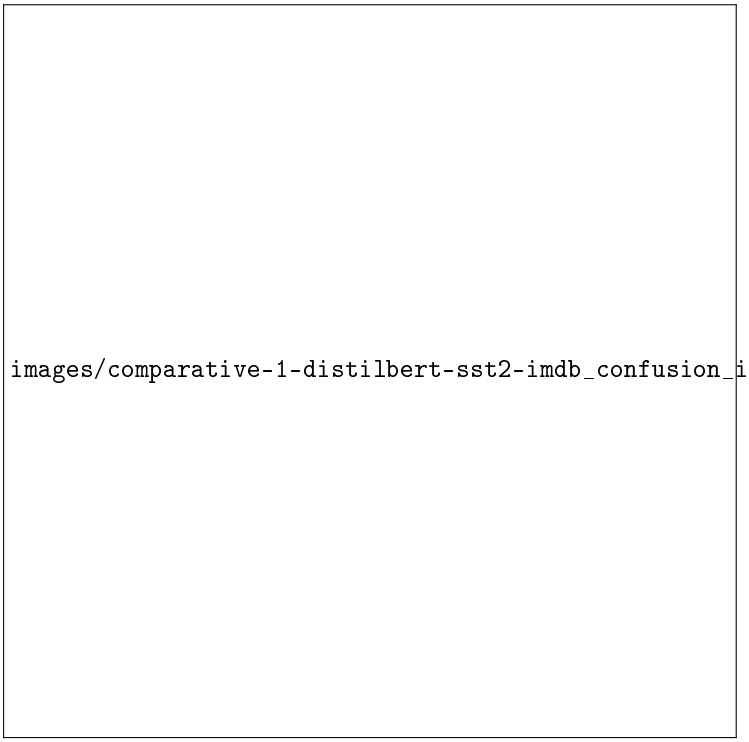

Figure 1. Confusion matrix on IMDb for the Bonferroni baseline run (distilbert-labeled configuration); higher values on the diagonal indicate better performance.

Figure 2. Confusion matrix on SST-2 for the Bonferroni baseline run (distilbert-labeled configuration); higher values on the diagonal indicate better performance.

Figure 3. Confusion matrix on Yelp for the Bonferroni baseline run (distilbert-labeled configuration); higher values on the diagonal indicate better performance.

Figure 4. Per-domain constrained accuracy summary for the Bonferroni baseline run (distilbert-labeled configuration); higher values indicate better performance.

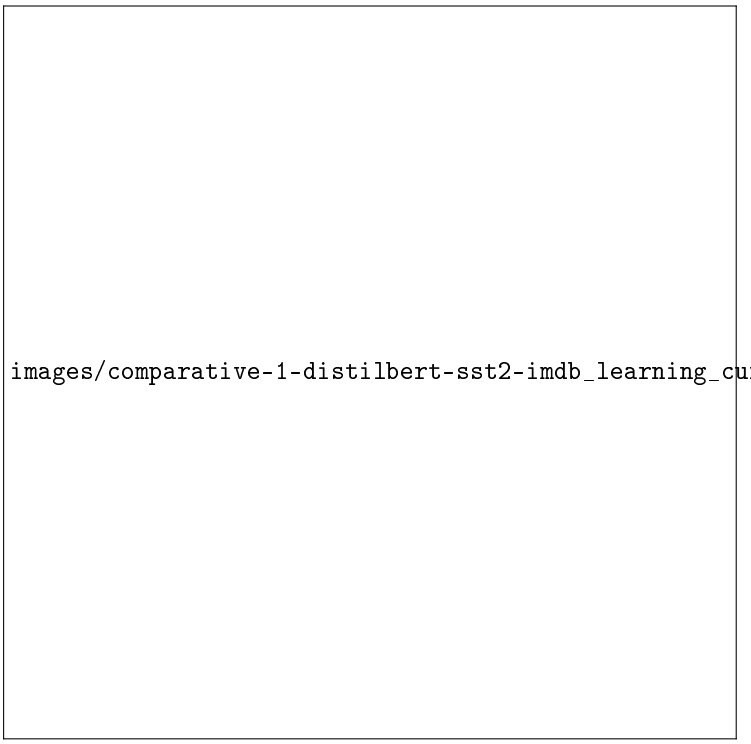

Figure 5. Robust constrained accuracy over optimization steps for the Bonferroni baseline run (distilbert-labeled configuration); higher values indicate better performance.

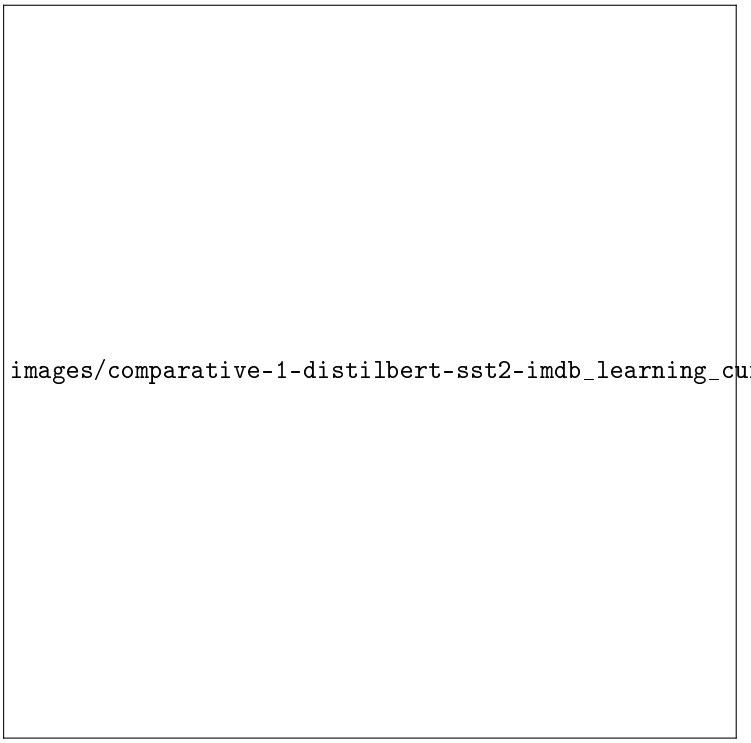

Figure 6. Worst-slice constrained accuracy over optimization steps for the Bonferroni baseline run (distilbert-labeled configuration); higher values indicate better performance.

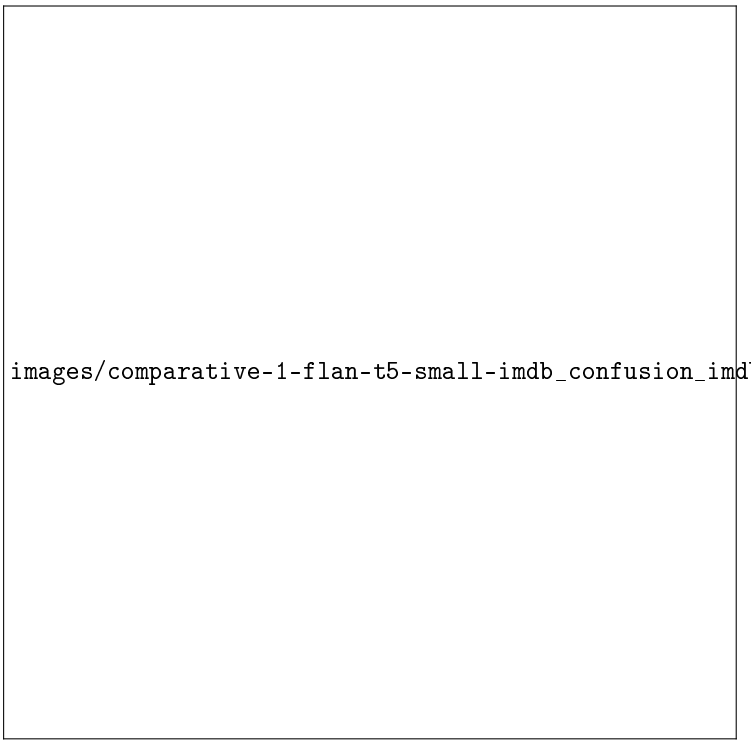

Figure 7. Confusion matrix on IMDb for the Bonferroni baseline run (flan-t5-small configuration); higher values on the diagonal indicate better performance.

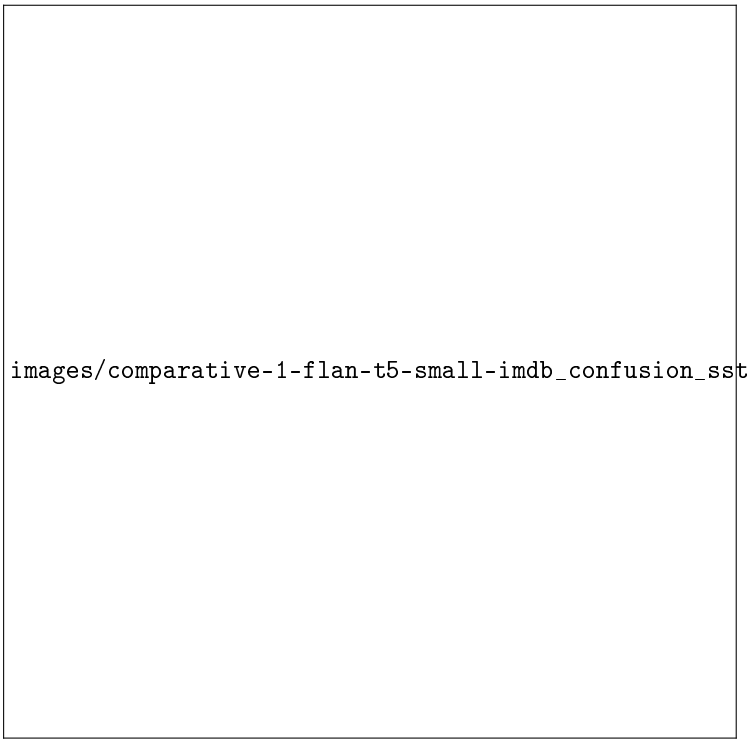

Figure 8. Confusion matrix on SST-2 for the Bonferroni baseline run (flan-t5-small configuration); higher values on the diagonal indicate better performance.

Figure 9. Confusion matrix on Yelp for the Bonferroni baseline run (flan-t5-small configuration); higher values on the diagonal indicate better performance.

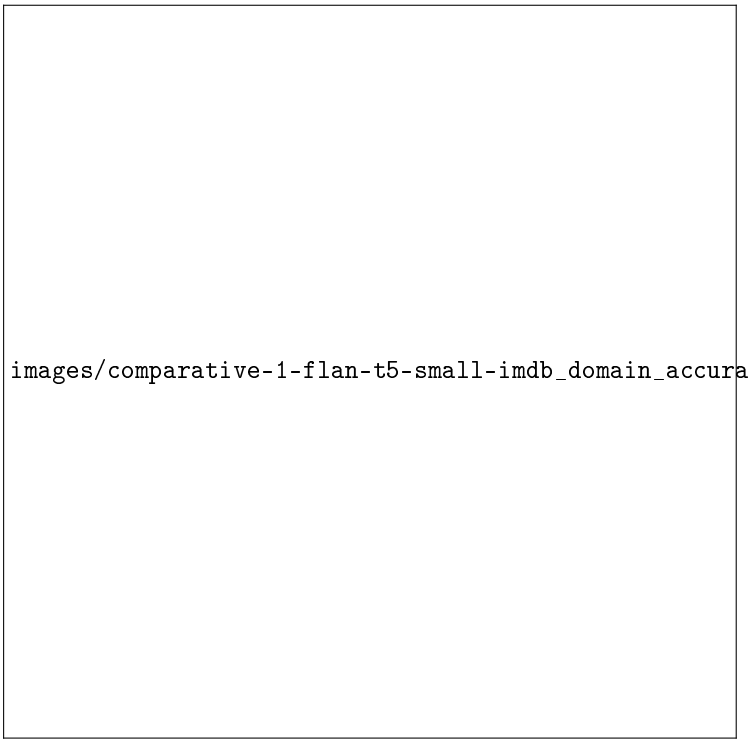

Figure 10. Per-domain constrained accuracy summary for the Bonferroni baseline run (flan-t5-small configuration); higher values indicate better performance.

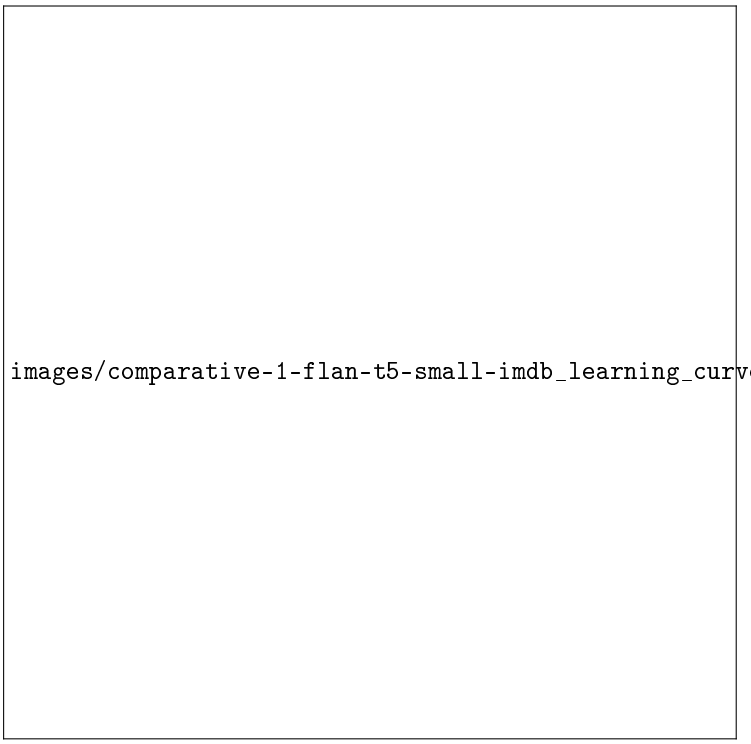

Figure 11. Robust constrained accuracy over optimization steps for the Bonferroni baseline run (flan-t5-small configuration); higher values indicate better performance.

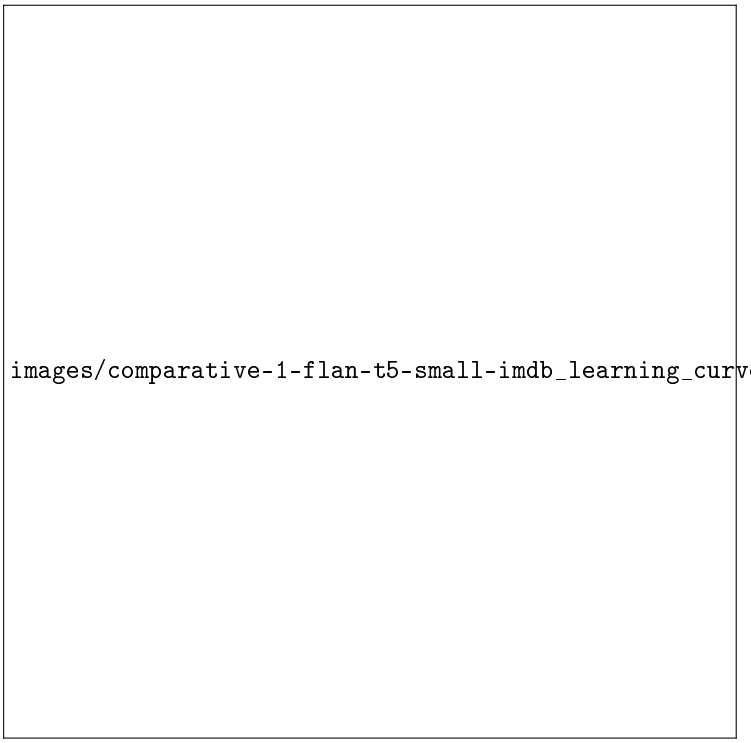

Figure 12. Worst-slice constrained accuracy over optimization steps for the Bonferroni baseline run (flan-t5-small configuration); higher values indicate better performance.

Figure 13. Confusion matrix on IMDb for the proposed CFS-eSAFE run (distilbert-labeled configuration); higher values on the diagonal indicate better performance.

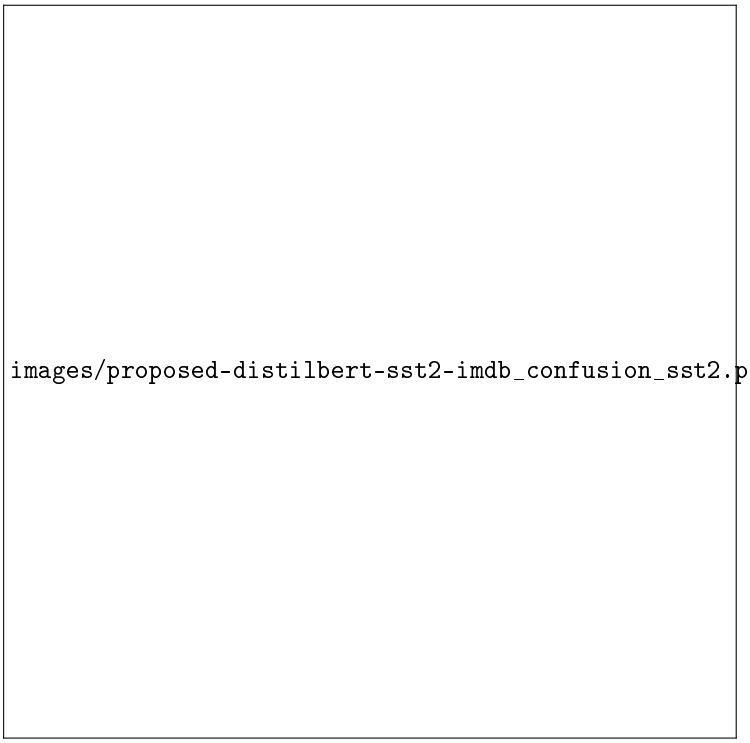

Figure 14. Confusion matrix on SST-2 for the proposed CFS-eSAFE run (distilbert-labeled configuration); higher values on the diagonal indicate better performance.

Figure 15. Confusion matrix on Yelp for the proposed CFS-eSAFE run (distilbert-labeled configuration); higher values on the diagonal indicate better performance.

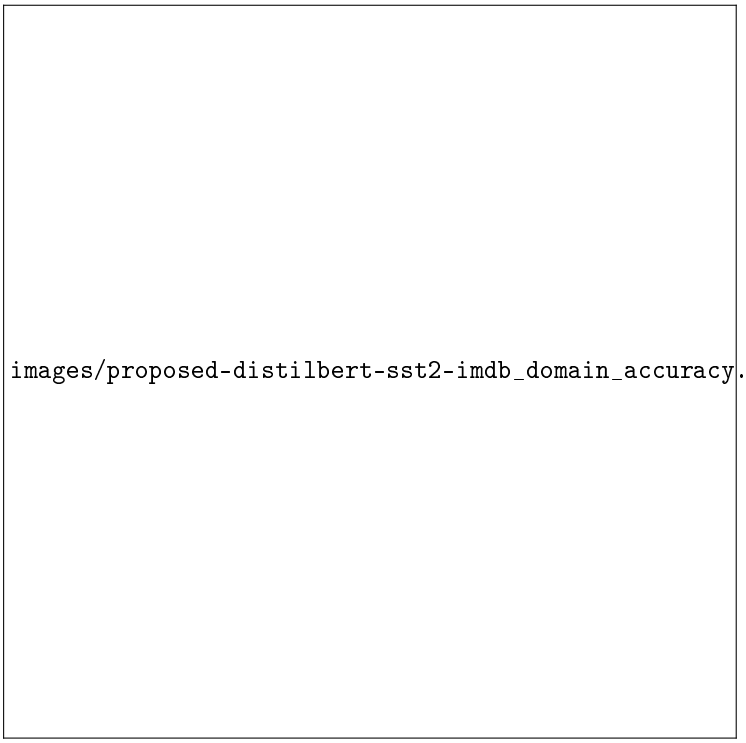

Figure 16. Per-domain constrained accuracy summary for the proposed CFS-eSAFE run (distilbert-labeled configuration); higher values indicate better performance.

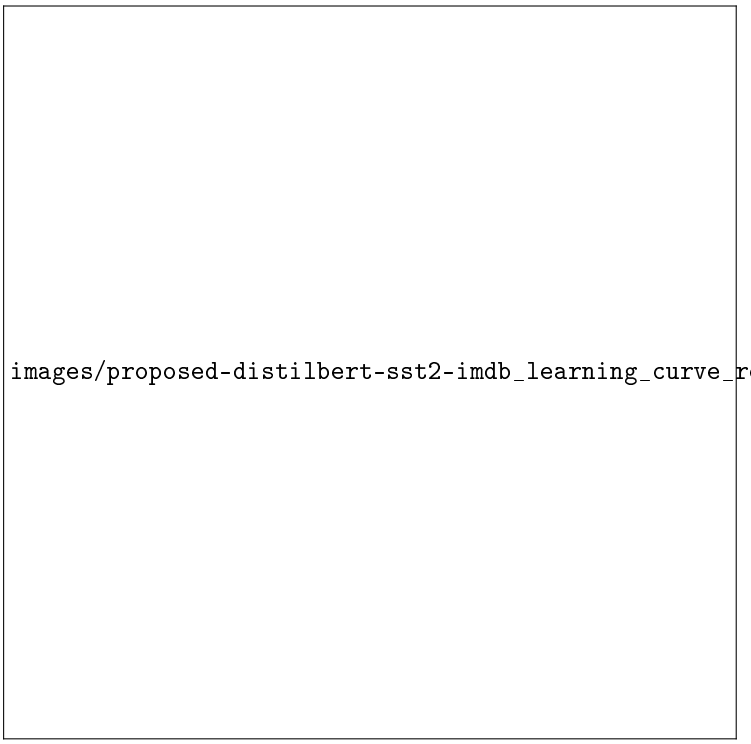

Figure 17. Robust constrained accuracy over optimization steps for the proposed CFS-eSAFE run (distilbert-labeled configuration); higher values indicate better performance.

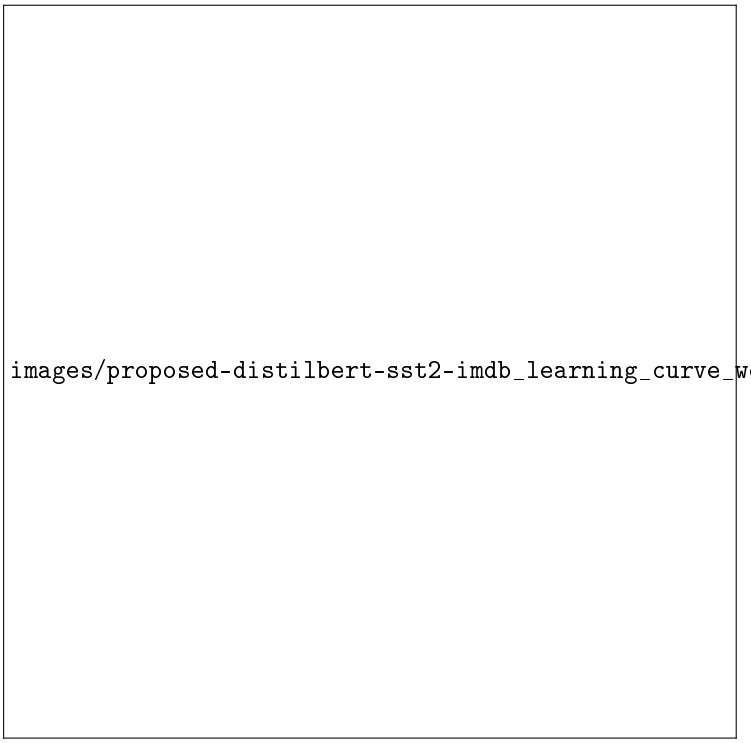

Figure 18. Worst-slice constrained accuracy over optimization steps for the proposed CFS-eSAFE run (distilbert-labeled configuration); higher values indicate better performance.

Figure 19. Confusion matrix on IMDb for the proposed CFS-eSAFE run (flan-t5-small configuration); higher values on the diagonal indicate better performance.

Figure 20. Confusion matrix on SST-2 for the proposed CFS-eSAFE run (flan-t5-small configuration); higher values on the diagonal indicate better performance.

Figure 21. Confusion matrix on Yelp for the proposed CFS-eSAFE run (flan-t5-small configuration); higher values on the diagonal indicate better performance.

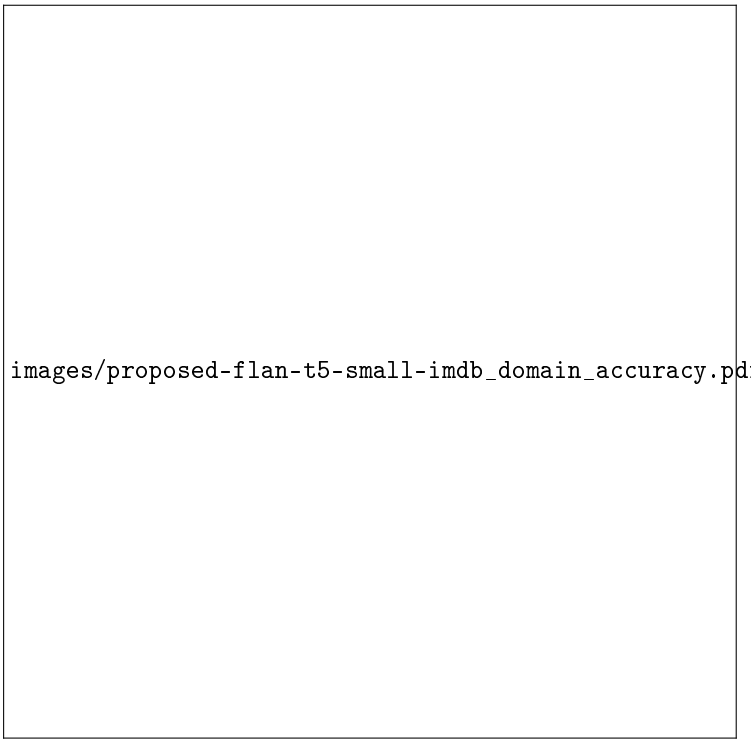

Figure 22. Per-domain constrained accuracy summary for the proposed CFS-eSAFE run (flan-t5-small configuration); higher values indicate better performance.

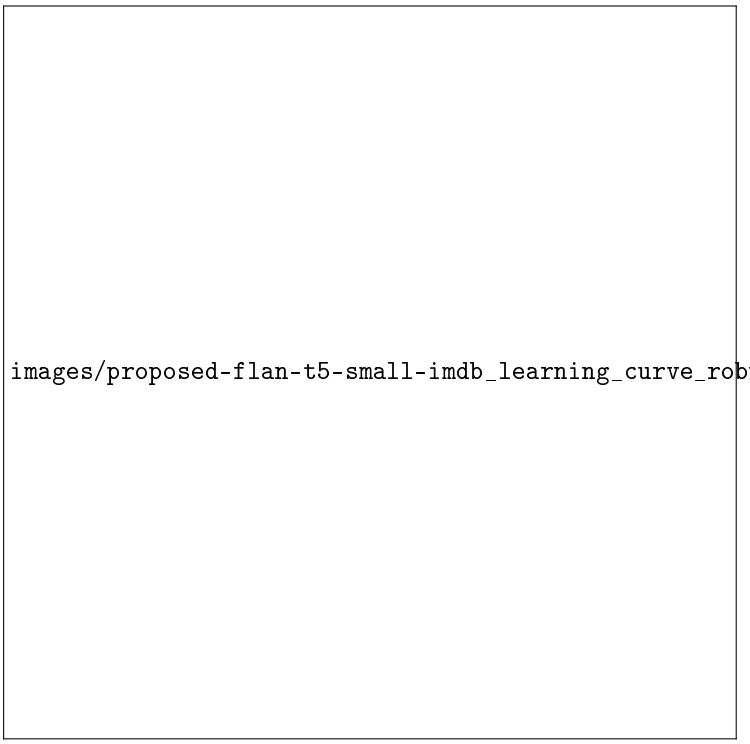

Figure 23. Robust constrained accuracy over optimization steps for the proposed CFS-eSAFE run (flan-t5-small configuration); higher values indicate better performance.

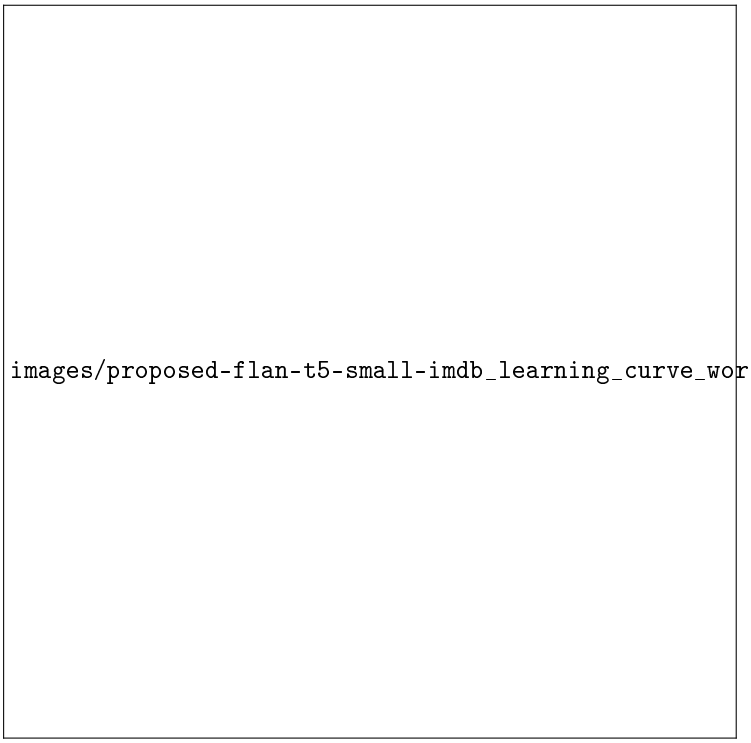

Figure 24. Worst-slice constrained accuracy over optimization steps for the proposed CFS-eSAFE run (flan-t5-small configuration); higher values indicate better performance.

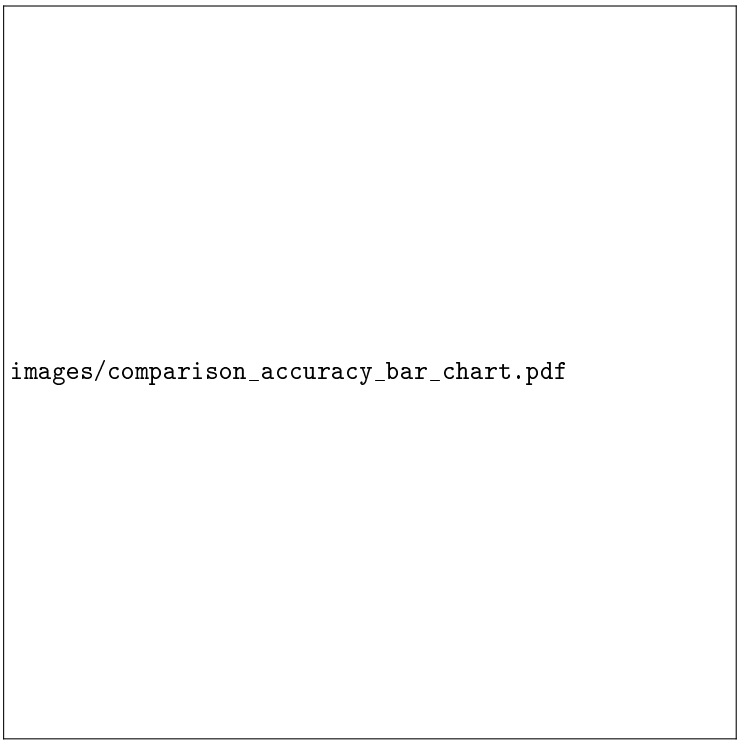

Figure 25. Cross-run comparison of robust constrained accuracy; higher values indicate better performance.

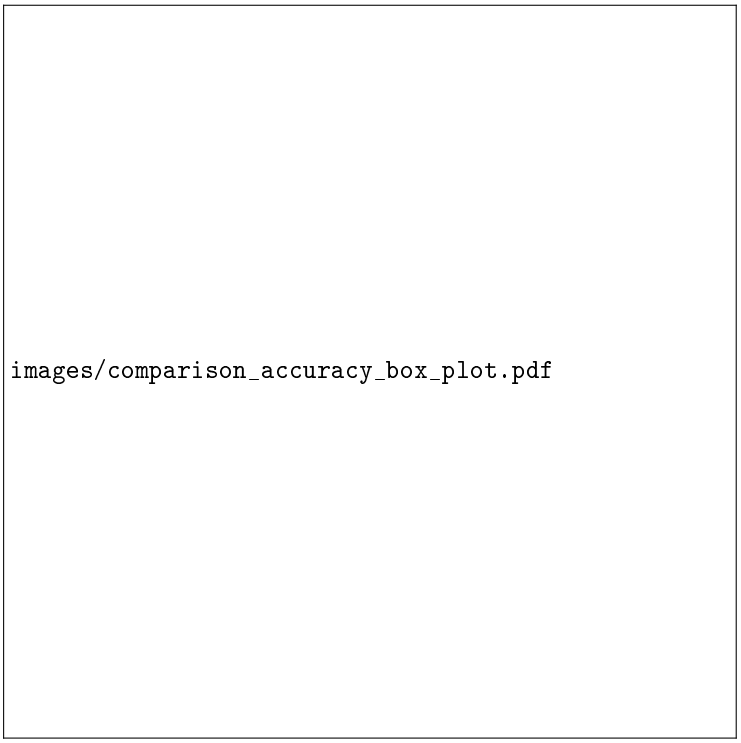

Figure 26. Cross-run seed-level distribution of robust constrained accuracy; higher values indicate better performance.

Figure 27. Cross-run metrics summary table visualization; higher is better for robust and worst-slice constrained accuracy and for accepted steps, while lower is better for OOD regression rate.

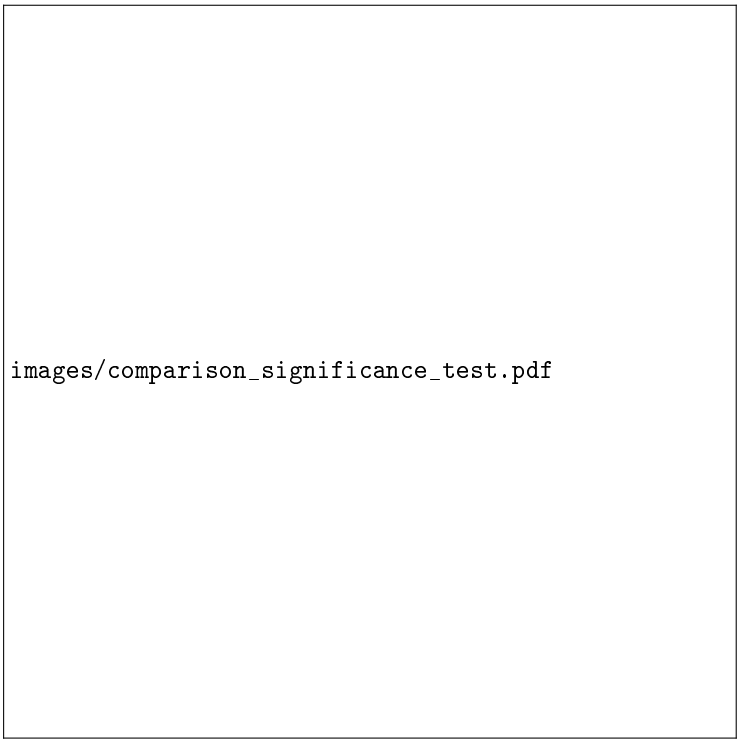

Figure 28. Proposed versus baseline significance test summary plot; higher robust constrained accuracy would indicate better performance, though all observed values are 0.0 in the executed runs.

7. Discussion

8. Conclusions

CFS-eSAFE is a training-free prompt optimization framework intended to provide an anytime-valid, deployment-oriented non-regression contract not only on an overall audit set but also on an adaptively discovered ordered list of stress slices. The method composes cross-fitted slice discovery (to avoid selection leakage), paired-difference evaluation, anytime-valid mixture e-values (to support optional stopping and repeated candidate testing), fixed-sequence (gatekeeping) slice certification (to avoid Bonferroni-style power collapse when many slices are audited), and alpha-wealth (to budget risk across an open-ended adaptive search).

The executed experiments do not empirically validate these intended advantages. Across all reported runs, constrained accuracy is 0.0 on SST-2, Yelp, and IMDb; worst-slice constrained accuracy is 0.0; and the optimizer evaluates zero candidates and accepts zero prompt updates. Confusion matrices are identical across methods and runs. The presence of some nonzero target matches in the confusion matrices alongside zero constrained accuracy indicates that semantic similarity and/or fluency constraints fail almost universally under the selected thresholds.

The main implication is methodological: before safety mechanisms can be compared, the underlying task instantiation must be non-degenerate. Two prerequisites emerge directly from the run summaries and logged outcomes: (i) calibrate $\tau_{\text{similarity}}$ and the fluency threshold ($\text{nll}_{\max}$, equivalently $\text{ppl}_{\max}$) so that constrained success is achievable at a nontrivial rate for the base prompt; and (ii) ensure the candidate proposal and filtering stage reliably generates and evaluates candidate prompts so that gating, slice discovery, and regression certificates are exercised.

Once these prerequisites are met, the core hypothesis becomes testable with the same framework and code structure: whether fixed-sequence e-value gatekeeping can certify many adaptively discovered slices with higher acceptance power than Bonferroni while

preserving interpretable regression certificates and end-to-end validity under adaptive, optionally stopped prompt search.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding.

Data Availability Statement: All resources used in this study are openly available at

Acknowledgments: In this study, we automatically carried out a series of research processes—from hypothesis formulation to paper writing—using generative AI.

Conflicts of Interest: The authors declare no conflicts of interest.