DIFFCONTEXT: LEARNING ADAPTIVE BUDGETS FOR RISK-CONTROLLED CONTEXT COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long contexts expand the information available to language models, but increase the cost of inference. Context compression offers a way to reduce this cost, yet its savings may come at the expense of evidence essential to the downstream task. The amount of context needed to preserve task performance varies across inputs: some contain repeated evidence, whereas others require complementary information from several passages. Applying the same retention budget across inputs can therefore discard necessary evidence in some cases while retaining unnecessary content in others. These differences motivate adaptive budget selection that balances token savings against the risk of task degradation. However, directly evaluating task quality at multiple budgets for each new input would require repeated task-model calls, potentially offsetting the savings. To address the challenge, we introduce **Diff**erentiated **Context** Compression (**DiffContext**), a task-aware, plug-in budget controller for existing compressors with adjustable retention budgets. Keeping the compressor and task model frozen, DiffContext uses offline task feedback to estimate harm and token cost across budgets and construct input-dependent policies. Because predicted harm alone does not guarantee risk control, independent calibration selects a deployment policy that limits the population frequency of utility drops beyond a specified tolerance relative to full-context inference, with a finite-sample guarantee under i.i.d. calibration and deployment sampling. At deployment, the selected policy chooses a budget from input features, requiring at most one compressor call and one task-model call per input. Across twelve configurations spanning three datasets and five compression backbones, DiffContext increases mean task-model token-cost savings from 3.61% for certified fixed budgets to 12.14% on the evaluated budget grid, under the same 10% limit on harm frequency. These results support adaptive budget control as a reusable way to retain compression savings under an explicit task-risk requirement. The code and data are available at `https://anonymous.4open.science/r/DiffContext-v1`.

1 INTRODUCTION

Long contexts enable language models to integrate evidence across passages for retrieval-augmented question answering (Tan et al., 2023; Rau et al., 2024; Yue et al., 2025) and document understanding (Bai et al., 2024), but increase the token cost of inference (Mu et al., 2023; Zhang et al., 2025). Context compression reduces this cost by removing or summarizing input content (Li et al., 2025; Liskavets et al., 2025). However, compression may also discard evidence required for the task and thereby degrade downstream performance (Nagle et al., 2024).

A retention budget specifies how much context a compressor should preserve, but the amount needed for the task varies across inputs. As illustrated in Figure 1, repeated evidence can allow substantial compression, whereas removing a necessary link between facts from different passages may make an input unanswerable. Applying one budget across inputs can therefore discard essential evidence in some cases while retaining unnecessary content in others, motivating adaptive budget selection.

LongLLMLingua prioritizes content according to its relevance to the question (Jiang et al., 2024), while AdaComp predicts how many retrieved documents are needed to answer a query (Zhang et al., 2024). Building on this input-dependent perspective, we study budget policies for existing compres-

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.

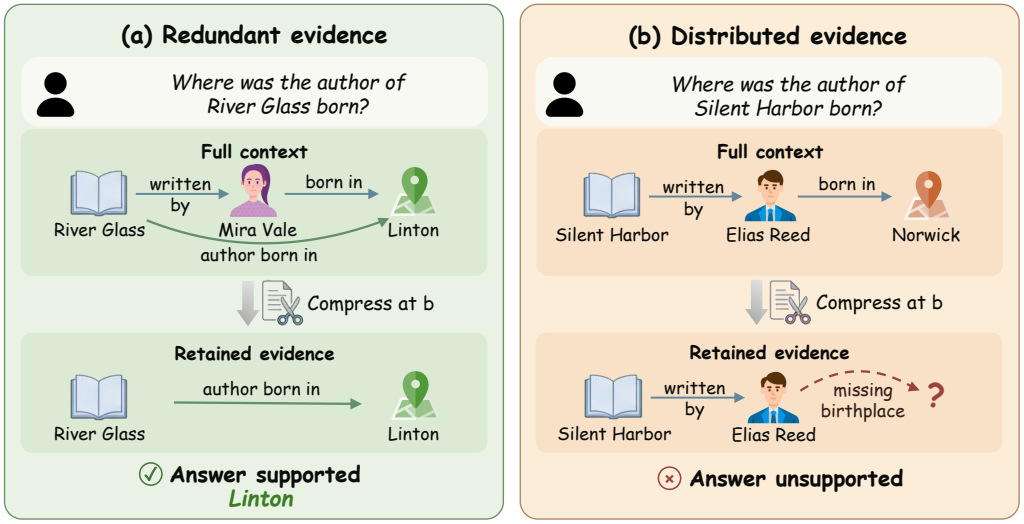

Figure 1: **The same retention budget can affect inputs differently.** Schematic examples using the same frozen compressor G and retention budget b. (a) Redundant evidence allows the compressed context to retain the information needed to answer the question. (b) Compression removes a necessary link between distributed facts, leaving insufficient evidence to answer the question.

sors under an explicit task-risk requirement. The challenge is to make these decisions from the input, before the resulting task outcome is available. Directly comparing budgets for each new input would require repeated task-model calls, potentially offsetting the savings.

We introduce **Diff**erentiated **Context** Compression (**DiffContext**), a task-aware, plug-in budget controller for compressors with adjustable retention budgets. Its design separates content selection from budget allocation: the compressor determines what to retain, while the controller learns how much to request from offline task feedback. With the compressor and task model frozen, we seek to reduce expected token cost while keeping the population frequency of harm below a prescribed limit. Here, harm denotes a utility drop beyond a specified tolerance relative to full-context inference. For each compressor–task configuration, offline evaluations across budgets support a harm predictor and a token-cost estimator, whose scores generate a finite family of input-dependent policies with different risk–cost preferences.

Because predicted harm alone does not ensure risk control, DiffContext certifies the frozen candidate policies on independent calibration data through Learn then Test fixed-sequence testing (Angelopoulos et al., 2025). Under i.i.d. calibration and deployment sampling, the selected policy satisfies the population-harm limit with a finite-sample guarantee, even when harm varies non-monotonically across budgets. Full context provides a fallback if no candidate passes. Once calibrated, the controller selects a budget from each input's features, followed by at most one compressor call and one task-model call.

Our main contributions are:

- We formulate task-aware budget allocation as a modular decision layer over existing compressors. Learning from offline task feedback, DiffContext adapts retention budgets to individual inputs while keeping the compressor and task model frozen.
- We integrate learned budget allocation with independent policy certification, providing a finite-sample population-harm guarantee under i.i.d. calibration and deployment sampling. This design accommodates prediction errors and non-monotonic compression effects.
- We conduct extensive experiments on three datasets across five compression backbones, covering twelve dataset–compressor configurations. The results show that DiffContext recovers token

savings where certified fixed budgets revert to full context, demonstrating the benefit of adaptive allocation under a common harm-frequency constraint.

2 PROBLEM FORMULATION

We consider a task instance $x = (q, C, y) \sim \mathcal{D}$, where q is the query, C is the full context, and y is the reference answer used for offline training and evaluation. Given a frozen compressor G and task model M, we seek a policy that chooses how much context to retain for each input (q, C). The policy π maps (q, C) to a retention budget $b \in \mathcal{B}$, where $\mathcal{B} \subset (0, 1]$ is a finite budget set containing the full-context action $b = 1$.

For a selected budget $b = \pi(q, C)$, the compressor produces the context passed to the task model. Smaller budgets request less retained content, whereas $b = 1$ uses the full context directly. Thus, the resulting context and prediction are

$$C_b = \begin{cases} G(q, C; b), & b < 1, \\ C, & b = 1, \end{cases} \qquad \hat{y}_b = M(q, C_b). \tag{1}$$

Compression-induced harm. To assess the effect of compression, we compare its utility with that of the same task model given the full context. Let $u(x, b) = \mathrm{Eval}(\hat{y}_b, y) \in [0, 1]$ denote task utility at budget b. Compression causes harm when the utility loss exceeds a tolerance $\epsilon \geq 0$:

$$\Delta u(x, b) = [u(x, 1) - u(x, b)]_+, \qquad h_\epsilon(x, b) = \mathbf{1}\{\Delta u(x, b) > \epsilon\}. \tag{2}$$

This definition gives $h_\epsilon(x, 1) = 0$ because full context is the reference. For compressed actions, we allow harm to vary arbitrarily with the retention budget. Applying a policy to the input population then gives the harm rate

$$R_\epsilon(\pi) = \mathbb{E}_{x \sim \mathcal{D}} \left[h_\epsilon\big(x, \pi(q, C)\big) \right]. \tag{3}$$

Risk-constrained budget selection. Our goal is to reduce token cost while keeping this harm rate below a prescribed limit. Let $c_{\mathrm{tok}}(x, b)$ be the task model's total input and output token count at budget b, normalized by its count under full-context inference for the same instance. Thus, $c_{\mathrm{tok}}(x, 1) = 1$. For policy class Π and risk limit $\alpha \in (0, 1)$, the objective is

$$\begin{aligned} \min_{\pi \in \Pi} \quad & \mathbb{E}_{x \sim \mathcal{D}} \left[c_{\mathrm{tok}}\big(x, \pi(q, C)\big) \right] \\ \text{subject to} \quad & R_\epsilon(\pi) \leq \alpha. \end{aligned} \tag{4}$$

Here, ϵ specifies how much utility loss constitutes harm, while α limits how frequently that harm may occur. The policy class includes $\pi_{\mathrm{full}}(q, C) = 1$, providing a feasible fallback with zero compression-induced harm.

3 DIFFCONTEXT

DiffContext selects a retention budget before task inference while keeping the compressor and task model frozen. As illustrated in Figure 2, it separates the use of task feedback from deployment through offline preparation and online inference. Offline evaluations at multiple budgets record task outcomes and token costs relative to full-context inference. These observations support a harm predictor and a cost estimator, whose scores are used to construct input-dependent candidate policies. The candidate policies are fixed before independent calibration assesses their harm and selects one for deployment. For a new input, the selected policy scores the available budgets using input features and chooses how much context to retain. The task model then receives either the full context or the output of one call to the frozen compressor.

3.1 OFFLINE BUDGET SCORING

Let $z = (q, C)$ denote the query and context available when a budget is chosen, and let $x = (z, y)$ include the reference answer available during offline evaluation. To choose a budget before compression, DiffContext evaluates the available budgets on a fit split $\mathcal{D}_{\mathrm{fit}}$, recording task outcomes and realized token costs alongside the full-context reference. These observations support a harm predictor $\hat{r}_\theta(\phi(z), b)$ and a cost decision score $\hat{c}(z, b)$; neither uses y at deployment.

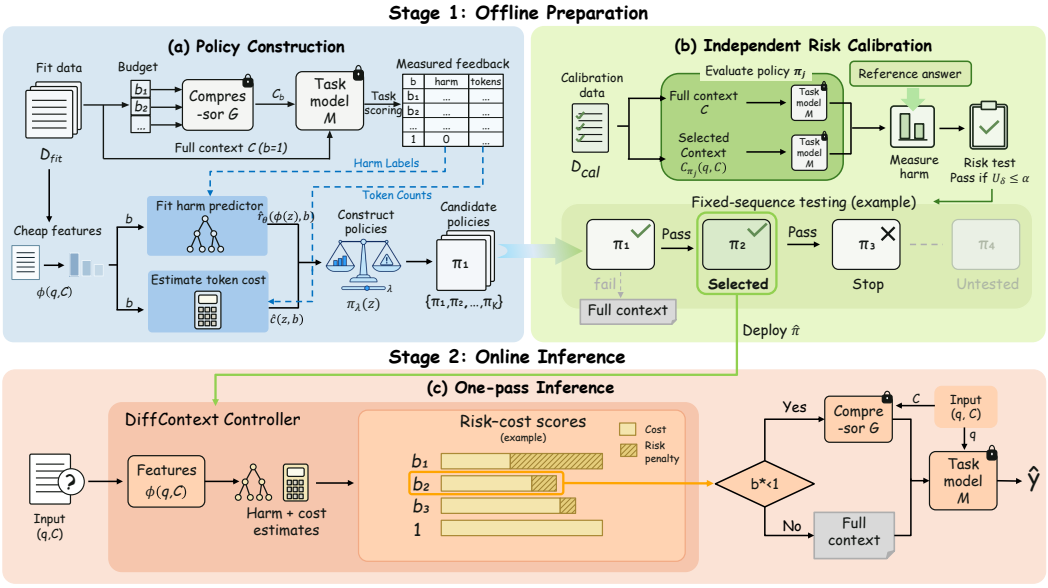

Figure 2: **Overview of DiffContext**. Offline preparation comprises (a) policy construction and (b) independent risk calibration; (c) shows one-pass online inference. In (b), testing stops at the first failure and selects the last passing policy, with full context as the fallback if none passes. Lock icons indicate the frozen compressor G and task model M.

Harm estimation. To score the compressed budgets, DiffContext feeds input and budget features into a shared classifier. Input features include context length, query length, and lexical overlap; budget features include the requested retention ratio and length proxies. The classifier is trained on binary labels $\widetilde{h}_i(b)$ stored with the fit evaluations, and a Platt map is fitted to out-of-fold scores on the same split. In six QA configurations, the fit labels encode drops in binary answer correctness, whereas calibration evaluates the F1-drop event in Eq. 2. Full context receives predicted harm zero because it is the reference action. Appendix B.3 reports label agreement and a nested comparison of the two supervision targets within the fit split.

Cost estimation. Alongside harm, budget selection must account for the cost savings available from each action. Let L_q denote the query-token count and L_C the sum of context-unit token counts. The requested retained context length gives an initial approximation of normalized input length:

$$a(z, b) = \frac{L_q + bL_C}{L_q + L_C}. \tag{5}$$

Since this approximation may differ from realized token cost, we use the offline observations to fit a median residual for each budget:

$$d_b = \text{median}_{x \in \mathcal{D}_{\text{fit}}} \left[c_{\text{tok}}(x, b) - a(z, b) \right]. \tag{6}$$

We apply a scaling fixed on the fit split to the corrected estimate $a(z, b) + d_b$, yielding $\hat{c}(z, b)$. This scaled value is used to rank budget actions; actual token savings are evaluated from realized c_{tok}, not from the scaled score. Appendix B gives the feature, model-selection, and scaling details.

3.2 Risk–Cost Policy Construction

Using these scores, DiffContext selects budgets by minimizing a weighted combination of estimated cost and predicted harm. For a risk weight $\lambda \geq 0$, we define

$$\pi_\lambda(z) = \arg\min_{b \in \mathcal{B}} \left[\hat{c}(z, b) + \lambda \hat{r}_\theta(\phi(z), b) \right]. \tag{7}$$

Although a policy uses the same weight for all inputs, their different scores can lead to different budget choices. Meanwhile, full context enters the comparison with zero predicted harm, allowing

the policy to retain all content when appropriate. By comparing budget scores directly, the rule also accommodates non-monotone predicted risks across retention levels.

Different values of λ express different risk–cost preferences, but not every weight produces a new budget decision. For a given input, the selected action can change only where two budget scores intersect. We therefore use these intersections to construct a finite candidate family. For budgets b and b' with unequal predicted risks, the intersection is

$$\lambda_{b,b'}(z) = \frac{\hat{c}(z, b') - \hat{c}(z, b)}{\hat{r}_\theta(\phi(z), b) - \hat{r}_\theta(\phi(z), b')}. \tag{8}$$

The positive finite intersections on fit inputs partition the weight range into intervals with unchanged fit-set decisions in their interiors. Using out-of-fold fit scores, we choose representative weights, remove policies with identical fit-set actions, and order the remaining candidates by decreasing measured fit cost. The selected classifier is refitted on all fit inputs while retaining the fitted Platt map. Each ordered candidate weight, the refitted predictor, the cost estimator, and the tie-breaking rule together define a final policy function $(\pi_1, \ldots, \pi_K)$. We freeze these functions and their order before examining calibration data. Calibration and deployment evaluate the same final functions, rather than the out-of-fold fit decisions. The order reflects *fit-set* savings; it does not assert monotone population costs. Appendix B.5 gives the construction details.

3.3 Independent Risk Calibration

Once the final candidate functions and their order are fixed, we evaluate their formal harm events on a calibration set $\mathcal{D}_{\text{cal}}$ independent of the fit data (Angelopoulos et al., 2024; 2025). For policy π_j, the number of harm events across the $n = |\mathcal{D}_{\text{cal}}|$ inputs is

$$m_j = \sum_{x_i \in \mathcal{D}_{\text{cal}}} h_\epsilon\big(x_i, \pi_j(z_i)\big). \tag{9}$$

Each calibration input is evaluated at the budget assigned by π_j, so the harm count summarizes the policy's input-dependent decisions across the calibration set. The resulting certificate applies to this complete decision rule, including its full-context choices, which contribute zero compression-induced harm. From this count, we compute the one-sided Clopper–Pearson upper bound $U_\delta(m_j, n)$ at confidence level $1-\delta$. A candidate passes when $U_\delta(m_j, n) \leq \alpha$. Fixed-sequence testing evaluates candidates in the predetermined order and stops at the first failure (Angelopoulos et al., 2025). We select the last passing policy, or fall back to full context if none passes.

Proposition 1 (Fixed-sequence risk control)**.** *Suppose the candidate policy sequence is fixed independently of the calibration data and calibration instances are sampled i.i.d. from the deployment distribution. The selected policy $\hat{\pi}$ satisfies*

$$\Pr_{\mathcal{D}_{\text{cal}}}\left[R_\epsilon(\hat{\pi}) \leq \alpha\right] \geq 1 - \delta. \tag{10}$$

Selecting an unsafe policy requires the first policy with risk above α to pass its test, an event with probability at most δ. Thus, fixed-sequence testing requires neither monotone policy risks nor independent tests. The upper-bound definition and full proof appear in Appendix C.

Deployment. After calibration, DiffContext computes predictor and decision-cost scores from each new input z and selects $b^* = \hat{\pi}(z)$. If $b^* < 1$, the compressor produces the corresponding context; otherwise, the full context is used directly. The task model then generates its output from that context. Because budget comparison uses these scores before task inference, each input requires at most one compressor call and one task-model call.

4 Experiments

To assess whether DiffContext reduces token costs while controlling compression-induced harm, we address the following four research questions:

- **RQ1:** How much compression does DiffContext retain under different risk limits?

- **RQ2:** How does adaptive budget selection compare with fixed budgets?
- **RQ3:** What does independent risk certification contribute to adaptive budget selection?
- **RQ4:** Do token savings reduce task-model expenditure after fit-label costs are amortized?

4.1 Experimental Setup

Tasks and models. We use Qwen-Plus as the frozen task model and evaluate on HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and ProcessBench (Zheng et al., 2025). HotpotQA includes BM25 (Robertson & Zaragoza, 2009), a dense selector, LongLLMLingua (Jiang et al., 2024), Selective Context (Li et al., 2023), and RECOMP (Xu et al., 2024); MuSiQue uses the latter four; and ProcessBench uses the dense selector, LongLLMLingua, and Selective Context. Compressor–dataset pairings are chosen based on compatibility with each task's input structure and evaluation protocol.

All configurations use retention budgets $\mathcal{B} = \{0.25, 0.50, 0.75, 1\}$, with 90 or 240 fit inputs and 180 inputs each for calibration and testing. ProcessBench uses MATH for fitting and Olympiad for calibration and testing. We set the utility-loss tolerance to $\epsilon = 0.10$ and the certification error level to $\delta = 0.01$. We use $\alpha = 0.10$ as a reference risk budget to study the compression–risk trade-off, and evaluate stricter budgets of 3%, 5%, and 7.5% in RQ1.

Comparisons and ablations. We compare DiffContext with three uncalibrated fixed budgets and a certified fixed-budget baseline selected by the same certification procedure, testing budgets in decreasing retention order. All policies use matched test inputs within each configuration. To examine the contribution of independent certification, we also evaluate a fit-only selector sharing DiffContext's fitted predictor, cost estimator, and frozen adaptive candidate family.

Evaluation metrics. We report Token-Cost Saving (CTS) relative to full-context inference and observed test harm. Let N_i^{π} and N_i^{full} denote the task model's total input and output token counts for instance i under policy π and full context, respectively:

$$\mathrm{CTS}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{N_i^{\pi}}{N_i^{\text{full}}}\right), \tag{11}$$

$$\widehat{H}_{\epsilon}(\pi) = \frac{1}{n}\sum_{i=1}^{n} h_{\epsilon}\big(x_i, \pi(q_i, C_i)\big). \tag{12}$$

Higher CTS and lower harm are preferred. Task utility is answer token F1 for HotpotQA and MuSiQue, and error-localization correctness for ProcessBench. To summarize the savings–harm trade-off, we additionally report a post hoc risk-priority score. With CTS and harm expressed as fractions, define

$$s_{\alpha}(\pi) = \begin{cases} 1 + \dfrac{\mathrm{CTS}(\pi)}{1 + \widehat{H}_{\epsilon}(\pi)/\alpha}, & \widehat{H}_{\epsilon}(\pi) \le \alpha, \\ \dfrac{\alpha}{\widehat{H}_{\epsilon}(\pi)}, & \widehat{H}_{\epsilon}(\pi) > \alpha, \end{cases} \tag{13}$$

$$T_{\alpha}^{\text{rel}}(\pi) = \frac{s_{\alpha}(\pi)}{s_{\alpha}(\pi_{\text{CF}})}. \tag{14}$$

Here, π_{CF} is the certified fixed-budget policy for the same configuration. The score rewards savings within the observed harm limit and penalizes exceedance. Certified fixed is normalized to one, with higher scores indicating a more favorable trade-off under this criterion. We average scores after configuration-level normalization. Appendix D.2 provides calculation details; mean harm excess is defined in Table 2.

4.2 RQ1: Compression under Different Risk Limits

Figure 3 examines compression savings under different risk limits while holding the fitted predictors and candidate sequences fixed. Relaxing the limit allows additional cost-saving policies to pass certification, increasing mean CTS on both question-answering datasets. ProcessBench retains smaller

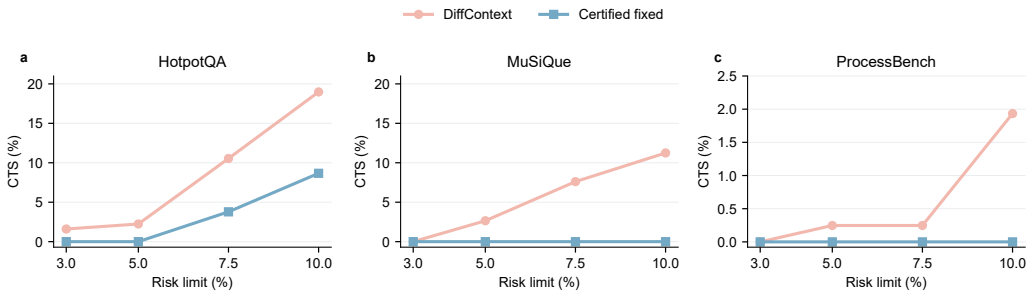

Figure 3: Token savings under a common certification requirement. Panels report mean test CTS across compressors within each dataset at risk limits of 3%, 5%, 7.5%, and 10%; higher CTS indicates greater savings. DiffContext and certified fixed budgets use the same fixed-sequence calibration protocol ($\delta = 0.01$), with fitted predictors and candidate families held fixed. Each configuration has 180 calibration and 180 test inputs. Markers identify evaluated settings; lines guide the eye.

Table 1: Fixed-budget and adaptive policies on 180 matched test inputs per configuration. CTS and observed Harm are percentages; higher CTS and lower Harm are preferred. The dimensionless descriptive score $T^{\text{rel}}_{0.10}$ is normalized to certified fixed within each configuration (baseline 1; higher is preferred). Bold marks the highest score in each row, including ties; † marks test Harm above 10%. Individual fixed budgets are uncalibrated. Certified fixed and DiffContext use $\alpha = 0.10$, $\delta = 0.01$; certified fixed selects 0.50 for HotpotQA–RECOMP and full context otherwise.

Dataset	Compressor	Fixed 0.25			Fixed 0.50			Fixed 0.75			Certified fixed			DiffContext (certified)		
		CTS	Harm	$T^{\text{rel}}_{0.10}$	CTS	Harm	$T^{\text{rel}}_{0.10}$	CTS	Harm	$T^{\text{rel}}_{0.10}$	CTS	Harm	$T^{\text{rel}}_{0.10}$	CTS	Harm	$T^{\text{rel}}_{0.10}$
HotpotQA	BM25	74.23	21.11†	0.474	50.61	9.44	**1.260**	27.16	8.89	1.144	0.00	0.00	1.000	22.66	6.67	1.136
	Dense	74.13	22.22†	0.450	50.56	15.00†	0.667	27.78	6.67	**1.167**	0.00	0.00	1.000	14.47	5.56	1.093
	LongLLMLingua	83.02	46.67†	0.214	67.28	40.56†	0.247	38.14	25.56†	0.391	0.00	0.00	1.000	3.07	2.22	**1.025**
	Selective Context	68.13	32.78†	0.305	38.25	19.44†	0.514	16.41	5.00	**1.109**	0.00	0.00	1.000	14.88	6.67	1.089
	RECOMP	68.77	16.11†	0.510	43.33	10.00	1.000	18.91	3.89	0.934	43.33	10.00	1.000	39.81	7.78	**1.006**
MuSiQue	Dense	73.87	25.00†	0.400	50.03	15.56†	0.643	25.26	8.89	**1.134**	0.00	0.00	1.000	15.29	6.11	1.095
	LongLLMLingua	88.19	43.33†	0.231	74.51	40.56†	0.247	47.36	31.11†	0.321	0.00	0.00	1.000	6.00	1.11	**1.054**
	Selective Context	69.67	42.22†	0.237	40.03	29.44†	0.340	17.13	11.67†	0.857	0.00	0.00	1.000	8.11	6.67	**1.049**
	RECOMP	70.55	18.33†	0.546	44.52	10.56†	0.947	19.33	6.67	1.116	0.00	0.00	1.000	15.58	2.22	**1.127**
ProcessBench	Dense	54.13	36.67†	0.273	38.92	34.44†	0.290	21.46	26.67†	0.375	0.00	0.00	1.000	5.06	2.78	**1.040**
	LongLLMLingua	50.23	44.44†	0.225	40.50	43.33†	0.231	29.18	42.22†	0.237	0.00	0.00	1.000	0.74	0.00	**1.007**
	Selective Context	49.94	39.44†	0.254	29.37	35.56†	0.281	11.61	27.78†	0.360	0.00	0.00	**1.000**	0.00	0.00	**1.000**
Mean across configurations		68.74	32.36	0.343	47.33	25.32	0.556	24.98	17.09	0.762	3.61	0.83	1.000	12.14	3.98	**1.060**

savings, showing that the compression available under a common risk requirement depends on the task–compressor combination.

At the reference risk limit, DiffContext retains nonzero compression in eleven of twelve configurations, with observed test harm below the limit throughout. Independent certification therefore permits useful compression across most evaluated settings, while the risk budget determines how much compression can be retained.

4.3 RQ2: Adaptive Selection versus Fixed Budgets

Under the same independent certification requirement, DiffContext increases mean CTS from 3.61% to 12.14%, with higher savings in most configurations (Table 1). Certified fixed returns full context in all but one configuration, whereas DiffContext retains positive savings in eleven. This difference highlights the benefit of input-dependent allocation: adaptive policies can preserve more context where predicted harm is high and compress more aggressively elsewhere, instead of applying one retention budget to every input. DiffContext also exceeds the certified-fixed reference on the relative score in most configurations. The full-context fallback of certified fixed budgets indicates that none of the tested uniform compression budgets passed the certification procedure in those configurations; adaptive allocation nevertheless recovers savings under the same requirement.

Table 2: Independent certification applied to the same frozen adaptive candidate family on 180 matched test inputs per configuration. CTS and observed Harm are percentages; $T^{\mathrm{rel}}_{0.10}$ is dimensionless and uses the same certified-fixed reference as Table 1. Higher CTS and scores and lower Harm are preferred. Bold marks the higher paired score; † marks observed test Harm above 10%. Mean excess is $\max(\mathrm{Harm} - 10, 0)$ in percentage points, averaged over all twelve configurations.

Dataset	Compressor	Fit-only (uncertified)			DiffContext (certified)		
		CTS	Harm	$T^{\mathrm{rel}}_{0.10}$	CTS	Harm	$T^{\mathrm{rel}}_{0.10}$
HotpotQA	BM25	38.19	8.89	**1.202**	22.66	6.67	1.136
	Dense	50.18	17.22†	0.581	14.47	5.56	**1.093**
	LongLLMLingua	19.78	15.00†	0.667	3.07	2.22	**1.025**
	Selective Context	35.70	18.89†	0.529	14.88	6.67	**1.089**
	RECOMP	42.95	8.33	**1.015**	39.81	7.78	1.006
MuSiQue	Dense	40.62	17.22†	0.581	15.29	6.11	**1.095**
	LongLLMLingua	36.11	16.11†	0.621	6.00	1.11	**1.054**
	Selective Context	27.92	16.67†	0.600	8.11	6.67	**1.049**
	RECOMP	42.16	11.11†	0.900	15.58	2.22	**1.127**
ProcessBench	Dense	24.46	16.11†	0.621	5.06	2.78	**1.040**
	LongLLMLingua	9.40	7.22	**1.055**	0.74	0.00	1.007
	Selective Context	20.18	15.56†	0.643	0.00	0.00	**1.000**
Mean across configurations		32.30	14.03	0.751	12.14	3.98	**1.060**

Against uncalibrated fixed budgets, DiffContext achieves a higher risk-priority relative score in most settings, including eight of twelve configurations against Fixed 0.75. Although this fixed budget yields greater mean token savings, its test harm exceeds the limit in half of the configurations, whereas DiffContext has no observed exceedances. Consequently, DiffContext obtains a higher mean relative score, 1.060 versus 0.762. The comparison shows how input-dependent selection preserves useful compression while reducing the harm associated with a uniform budget.

4.4 RQ3: The Role of Independent Risk Certification

Table 2 compares two policy-selection procedures using the same fitted predictor, cost estimator, and frozen adaptive candidate family. Fit-only selection uses fit-set outcomes, whereas DiffContext selects a policy through independent calibration. DiffContext reduces mean test harm from 14.03% to 3.98%, while the number of configurations exceeding the observed harm limit falls from nine to zero. Mean harm excess likewise falls to zero.

This reduction accompanies lower CTS, reflecting more conservative budget allocation. The mean relative score increases from 0.751 to 1.060, with DiffContext achieving a higher score in most configurations under the risk-priority criterion. The additional savings obtained by fit-only selection are accompanied by more frequent harm-limit exceedances, whereas DiffContext retains compression benefits at lower observed harm. Appendix E reports restricted-feature controls and complete configuration-level results across all evaluated dataset–compressor pairings.

4.5 RQ4: Task-Model Expenditure and Amortization

The preceding experiments measure compression benefits through token counts. Since input and output tokens have different prices, we price them separately to assess task-model expenditure. All eleven configurations with nonzero compression reduce online expenditure to 0.61–0.99 times that of full-context inference (Figure 4a). Generating fit labels introduces an upfront token expenditure, which these online savings offset after approximately 640–14,000 deployment inputs (Figure 4b). As deployment volume grows, the fit-label expenditure allocated to each input decreases, and the amortized expenditure ratio approaches the online ratio. Thus, per-input savings determine the long-run expenditure reduction, while deployment scale determines when the initial fit-label expenditure is fully recovered through cumulative savings on task-model inference.

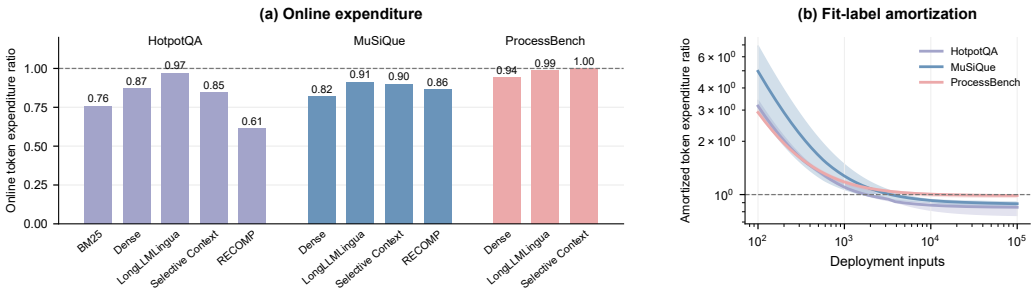

Figure 4: Task-model expenditure relative to full context. (a) Online expenditure ratios. (b) Ratios after amortizing fit-label token expenditure over deployment inputs; lines and bands show medians and interquartile ranges across compressors within each dataset. Lower ratios are better, and values below the dashed parity line indicate savings. This partial cost accounting excludes compressor compute, calibration, predictor fitting, and controller execution.

5 Related Work

Context compression. Context compression reduces task-model input by selecting or summarizing content (Li et al., 2025). Text-filtering methods decide what to retain (Chung et al., 2024; Jiang et al., 2023a; Liskavets et al., 2025; Zhang et al., 2025): Selective Context uses self-information, LongLLMLingua uses question relevance, and LLMLingua-2 learns token-level retention (Li et al., 2023; Jiang et al., 2024; Pan et al., 2024). Summarization and task-oriented reformulation provide other ways to preserve useful information, including RECOMP's task-feedback-trained compressor (Lee et al., 2024; Xu et al., 2024; Zhao et al., 2025; Cheng et al., 2024). DiffContext leaves content selection to these compressors and instead controls the requested retention budget.

Adaptive resource allocation. Input-dependent methods also vary the amount of context or computation used (Yue et al., 2025). AdaComp predicts how many retrieved documents a query needs (Zhang et al., 2024); RouteLLM routes between models, while Conformal Thinking calibrates reasoning-time computation (Ong et al., 2025; Wang et al., 2026). More broadly, adaptive allocation replaces one fixed resource setting with input-dependent decisions (Jiang et al., 2023b; Ma et al., 2023). DiffContext focuses on retention budgets for a fixed compressor and task model, with a specified limit on compression-induced harm.

Statistical risk control. Conformal Risk Control calibrates predictive rules for bounded losses, with monotonicity required in its basic formulation (Angelopoulos et al., 2024). Learn then Test instead constructs candidate rules before calibration and tests them against a risk requirement (Angelopoulos et al., 2025). DiffContext uses its fixed-sequence testing procedure to select among frozen budget policies; the resulting guarantee does not require their population risks to vary monotonically along the prespecified policy testing sequence.

6 Conclusion

DiffContext uses offline task feedback to construct input-dependent budget policies for a frozen compressor and task model, then selects a deployment policy through independent calibration. Under i.i.d. calibration and deployment sampling, fixed-sequence testing provides a finite-sample population-harm guarantee for the selected policy. Across twelve configurations, DiffContext retains useful compression in most settings and improves token reduction over certified fixed budgets. These results show that risk control need not impose one retention budget on every input. In future work, we will incorporate compressor overhead into budget selection and explore more informative, low-cost input signals for predicting compression-induced harm.

AI USE STATEMENT

We have not used generative AI tools to generate synthetic data, develop theoretical models, formulate or prove mathematical claims, propose or refine hypotheses, design or implement the methodology, or interpret results; these tasks were performed by the authors, and the remaining tasks requiring disclosure are not applicable to this work. We used generative AI tools to edit the manuscript for readability, to brainstorm phrasing and structure, and to format references. All AI-assisted edits were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anastasios Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International conference on learning representations*, volume 2024, pp. 55198–55218, 2024.

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516, 2024.

Tsz Ting Chung, Leyang Cui, Lemao Liu, Xinting Huang, Shuming Shi, and Dit-Yan Yeung. Selection-p: Self-supervised task-agnostic prompt compression for faithfulness and transferability. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11057–11070, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023a.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023b.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*, 2024.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 6342–6353, 2023.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7182–7195, 2025.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K Luke. Prompt compression with context-aware sentence encoding for fast and improved llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24595–24604, 2025.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5303–5315, 2023.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

Alliot Nagle, Adway Girish, Marco Bondaschi, Michael Gastpar, Ashok Vardhan Makkuva, and Hyeji Kim. Fundamental limits of prompt compression: A rate-distortion framework for black-box language models. *Advances in Neural Information Processing Systems*, 37:94934–94970, 2024.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024.

David Rau, Hervé Déjean, Nadezhda Chirkova, Thibault Formal, Shuai Wang, Stéphane Clinchant, and Vassilina Nikoulina. Bergen: A benchmarking library for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7640–7663, 2024.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174, 2009.

Wang-Chiew Tan, Yuliang Li, Pedro Rodriguez, Richard James, Xi Victoria Lin, Alon Halevy, and Wen-tau Yih. Reimagining retrieval augmented language models for answering queries. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6131–6146, 2023.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Xi Wang, Anushri Suresh, Alvin Zhang, Rishi More, William Jurayj, Mehrdad Farajtabar, Daniel Khashabi, and Eric Nalisnick. Conformal thinking: Risk control for reasoning on a compute budget. In *Forty-third International Conference on Machine Learning*, 2026.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with context compression and selective augmentation. In *International Conference on Learning Representations*, volume 2024, pp. 43478–43502, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. In *International Conference on Learning Representations*, volume 2025, pp. 72914–72938, 2025.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *International Conference on Learning Representations*, volume 2025, pp. 101932–101948, 2025.

Qianchi Zhang, Hainan Zhang, Liang Pang, Hongwei Zheng, and Zhiming Zheng. Adacomp: Extractive context compression with adaptive predictor for retrieval-augmented large language models. *arXiv preprint arXiv:2409.01579*, 2024.

Yi Zhao, Zuchao Li, Hai Zhao, Baoyuan Qi, and Liu Guoming. Dac: A dynamic attention-aware approach for task-agnostic prompt compression. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19395–19407, 2025.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, 2025.

A Experimental Configuration

Table S1 provides the configuration-level split sizes and retention budgets used in the experiments. It complements the dataset and model descriptions in the main text.

Table S1: Experimental configurations with the unified retention grid. The task model is Qwen-Plus. ProcessBench fitting uses MATH and calibration/testing use Olympiad.

Dataset	Compressor	n_{fit}	n_{cal}	n_{test}	Budgets
HotpotQA	BM25	240	180	180	0.25, 0.50, 0.75, 1.00
	Dense	90	180	180	0.25, 0.50, 0.75, 1.00
	LongLLMLingua	90	180	180	0.25, 0.50, 0.75, 1.00
	Selective Context	90	180	180	0.25, 0.50, 0.75, 1.00
	RECOMP	90	180	180	0.25, 0.50, 0.75, 1.00
MuSiQue	Dense	90	180	180	0.25, 0.50, 0.75, 1.00
	LongLLMLingua	90	180	180	0.25, 0.50, 0.75, 1.00
	Selective Context	240	180	180	0.25, 0.50, 0.75, 1.00
	RECOMP	240	180	180	0.25, 0.50, 0.75, 1.00
ProcessBench	Dense	90	180	180	0.25, 0.50, 0.75, 1.00
	LongLLMLingua	90	180	180	0.25, 0.50, 0.75, 1.00
	Selective Context	90	180	180	0.25, 0.50, 0.75, 1.00

B Predictors and Policy Construction

B.1 Input Features

The controller uses eight input statistics: context-token count, number of context units, query-token count, mean and standard deviation of unit-token counts, and mean, maximum, and standard deviation of query–unit lexical overlap. Lexical terms are lowercased alphanumeric strings. For query-term set Q and unit-term set T_j, overlap is $|Q \cap T_j| / \max(1, |Q|)$. Empty collections receive zero summary statistics.

For each compressed action, the predictor additionally receives b, bL_C, and $(L_q + bL_C)/(L_q + L_C)$, where L_C is the sum of context-unit token counts and L_q is the query-token count. These feature lengths support pre-compression scoring; realized context reduction is measured from the resulting context. This gives eleven features per input–budget pair. Context length is lower-bounded by one in the length ratios. Full context is excluded from classifier fitting and assigned predicted harm zero.

B.2 Harm Predictor

Three base classifiers are considered: standardized logistic regression ($C = 1$, balanced class weights, 3,000 maximum iterations), a random forest (500 trees, maximum depth 5, minimum leaf size 5, balanced-subsample class weights), and extra trees (500 trees, maximum depth 6, minimum leaf size 5, balanced class weights). Each is combined with either a global or a budget-specific Platt map.

Five shuffled folds are formed at the example level, keeping all budget observations for an example in the same fold. The resulting out-of-fold probabilities supply the inputs to the Platt fit. Probabilities are clipped to $[10^{-6}, 1 - 10^{-6}]$ before taking logits. The Platt map is logistic regression on these logits with $C = 10^6$ and a maximum of 2,000 iterations; a single-class target uses its empirical prevalence. The post-Platt fit diagnostics reuse the labels that estimate this map; they measure fit-stage behavior.

For each classifier–Platt-map combination, candidate policies are evaluated on fit data using calibrated out-of-fold scores. For this model-selection step, a candidate is admissible when its measured fit harm is at most 0.05. Model selection first minimizes the measured fit cost of an admissible candidate, then Brier score, then negative area under the precision–recall curve. The selected base classifier is refitted on all fit examples, retaining the Platt map learned from out-of-fold scores. The out-of-fold matrix is used for fit-stage model selection and candidate construction; the refitted classifier supplies the scores of the final candidate functions used on calibration and deployment inputs. The policy risk certificate comes from the independent calibration set, not from Platt scaling.

B.3 Fit Supervision and Label Alignment

The harm predictor is trained on the stored binary `induced_error` labels, denoted $\widetilde{h}_i(b)$. In six question-answering configurations, these labels were generated from drops in binary answer correctness rather than the answer-F1 utility used by the formal harm event in Eq. 2. Consequently, 5–30 of the 270 compressed fit records per configuration disagree with the formal event (Table S2). These stored labels serve only as fit-stage proxy supervision; independent calibration and test evaluation use the formal harm event defined in the main text.

A nested comparison confined to the fit split examines the effect of replacing those labels. Three outer folds reserve 30 of the 90 inputs for evaluation in each fold, while five inner folds support classifier selection and Platt scaling on the remaining 60. Stored-label and F1-drop-label variants share folds, seeds, and model-selection settings. Both rich and length-only feature sets are refitted within this comparison. The mean per-budget AUROC of the rich predictor, computed from pooled outer-fold predictions, does not improve in any of the six configurations. These fits characterize label alignment; the reported deployment policies retain their stored-label supervision.

Table S2: Stored fit-label disagreement and a nested comparison confined to the fit split. The last two columns report mean per-budget AUROC for rich features, using pooled outer-fold predictions against the F1-drop target. Each affected configuration has 90 fit inputs and 270 compressed input–budget records.

Dataset	Compressor	Disagreeing labels	Stored labels	F1-drop labels
HotpotQA	Dense	8/270	0.461	0.450
	LongLLMLingua	20/270	0.475	0.453
	Selective Context	7/270	0.465	0.371
	RECOMP	5/270	0.392	0.386
MuSiQue	Dense	15/270	0.538	0.526
	LongLLMLingua	30/270	0.459	0.424

B.4 Token-Cost Estimate

The cost branch combines a length proxy with an action-specific residual correction. For fit example i, let $c_i(b)$ be its recorded normalized token cost and define

$$a(z_i, b) = \frac{L_{q,i} + bL_{C,i}}{L_{q,i} + L_{C,i}}, \qquad d_b = \mathrm{median}_{i \in \mathcal{D}_{\mathrm{fit}}}[c_i(b) - a(z_i, b)]. \tag{15}$$

Writing $z = (q, C)$ for the online input, the raw estimate is $\widetilde{c}(z, b) = a(z, b) + d_b$. If $c_{\min}$ and $c_{\max}$ are its extrema over fit input–action pairs, the score used by the controller is

$$\widehat{c}(z, b) = \mathrm{clip}\left(\frac{\widetilde{c}(z, b) - c_{\min}}{\max(c_{\max} - c_{\min}, 10^{-12})}, 0, 1.5\right). \tag{16}$$

The residuals and normalization constants are frozen after fitting. This branch uses lengths and cost feedback independently of the harm predictor. Because scaling can change the numerical value of the full-context score, $\widehat{c}(z, 1)$ need not equal the realized normalized cost $c_{\mathrm{tok}}(x, 1) = 1$; the former is a decision score and the latter is used to measure actual savings.

B.5 Finite Policy Family

To instantiate the cost–risk policy family in the main text, we search the action-switching points induced by the fit data. Ties in action scores follow the stored action order, and the full-context policy is maintained separately as a fallback.

For a fit input z and actions b, b' with unequal predicted risks, their score-intersection multiplier is

$$\lambda_{z,b,b'} = \frac{\widehat{c}(z, b') - \widehat{c}(z, b)}{\widehat{r}_\theta(\phi(z), b) - \widehat{r}_\theta(\phi(z), b')}. \tag{17}$$

The search retains positive, finite intersections across fit examples. It rounds these values to twelve decimal places and removes duplicates. Representatives include zero, a point below the first intersection, midpoints between successive intersections, and a point above the last intersection. If no positive intersection exists, the search uses zero.

Policies with identical fit-set action vectors are deduplicated. Final candidate construction uses a fit-harm cap of 1.0 and retains policies improving measured fit cost over full context; this cap does not exclude any binary-harm policy on its own. It selects representatives across a grid of fit compression coverage values and empirical coverage quantiles. For a target coverage, ties are resolved by distance to that target, lower fit harm, and then higher fit cost. The resulting candidates are ordered by decreasing fit cost, then increasing fit harm and coverage. If the candidate limit is exceeded, evenly spaced positions in this order are retained. The limit is twelve candidates. This weight sequence is constructed without calibration outcomes. The refitted classifier, retained Platt map, cost estimator, candidate weights, and action tie-breaking then define the final functions evaluated on calibration inputs and later used for deployment; no candidate is revised after inspecting calibration harm.

Fit-only policy selection. The fit-only selector uses the same fitted predictor, cost estimator, and frozen adaptive candidate family as DiffContext. It chooses the candidate with the lowest measured fit cost among those with measured fit harm at most 0.10.

C Calibration Algorithm and Guarantee

For frozen policy π_j, let m_j be the number of harm events among n calibration examples. Its one-sided binomial upper bound is

$$U_\delta(m_j, n) = \begin{cases} \mathrm{Beta}^{-1}(1-\delta; m_j+1, n-m_j), & m_j < n, \\ 1, & m_j = n. \end{cases} \tag{18}$$

The policy passes when $U_\delta(m_j, n) \le \alpha$. Testing proceeds through the frozen order and stops at the first failure.

The policy outcomes can be computed from cached input–budget evaluations, so certification does not require repeating task-model calls for actions already evaluated. The following proposition establishes the guarantee for the stopping rule in the main text.

Proposition. Condition on the fit data and all choices used to construct the policy sequence. Suppose calibration examples are independent draws from the deployment distribution and each frozen policy defines a pointwise Bernoulli harm event. Then the returned policy satisfies

$$\Pr_{D_{\mathrm{cal}}}\{R_\epsilon(\hat{\pi}) > \alpha\} \le \delta, \qquad R_\epsilon(\pi) = \mathbb{E}[h_\epsilon(X, \pi(Q, C))]. \tag{19}$$

Proof. If every candidate has risk at most α, the claim follows immediately, since full context also has zero risk. Otherwise, let j_0 be the first candidate in the fixed order with $R_\epsilon(\pi_{j_0}) > \alpha$. Returning any unsafe candidate requires passing candidate j_0: a failure there stops the procedure before any later candidate can be returned. Thus

$$\Pr\{R_\epsilon(\hat{\pi}) > \alpha\} \le \Pr\{U_\delta(m_{j_0}, n) \le \alpha\} \tag{20}$$

$$\le \Pr\{U_\delta(m_{j_0}, n) < R_\epsilon(\pi_{j_0})\} \le \delta, \tag{21}$$

where the final inequality is the coverage property of the one-sided binomial bound. The argument does not require independence between candidate tests or monotonicity of their true risks. Averaging over the fit data gives the unconditional statement. □

This is a fixed-sequence testing guarantee: with probability at least $1-\delta$ over the calibration sample, the returned policy has population harm at most α. It does not assert that the selected policy minimizes population cost over all possible policies. The confidence level applies separately to each certified method–configuration pair. For stochastic task execution, the harm event includes the execution randomness under the evaluation protocol. The distribution and execution protocol in this statement are those used for calibration.

D Evaluation Metrics and Paired Comparisons

D.1 Token Savings, Harm, and Coverage

In addition to CTS and test harm reported in the main text, the complete results include context-token reduction and compression coverage:

$$\mathrm{TRed}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\left[1 - \frac{L(C_{b_i})}{L(C_i)}\right], \qquad \mathrm{Cov}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{b_i < 1\}, \tag{22}$$

where $b_i = \pi(q_i, C_i)$. Coverage counts a selected compression action even if its realized reduction is small.

D.2 Relative Score and Aggregation

Equations 13 and 14 define a post hoc descriptive score with a risk-priority preference: for the nonnegative CTS values in these comparisons, observed harm above α gives $0 < s_\alpha < 1$, whereas positive savings within the limit give $s_\alpha > 1$. Within the latter region, larger CTS increases the score and larger harm decreases it. The piecewise rule has a downward jump at α when CTS is positive; it explicitly prioritizes meeting the observed harm limit over retaining savings above that limit.

The relative score divides by the certified fixed-budget policy's score separately in each configuration. This reference selects full context in eleven configurations, for which $s_\alpha = 1$. On HotpotQA–RECOMP it selects budget 0.50: $s_{0.10} = 1 + 0.4333/(1 + 0.1000/0.10) = 1.21665$. DiffContext has $s_{0.10} = 1 + 0.3981/(1 + 0.0778/0.10) \approx 1.223903$, giving $T^{\mathrm{rel}}_{0.10} \approx 1.005962$, displayed as 1.006.

A relative score below one means a lower score than the reference, not necessarily a risk-limit exceedance. For example, Fixed 0.75 on HotpotQA–RECOMP has 3.89% observed harm but a relative score of 0.934. Conversely, an uncertified policy can score above one if its test outcomes meet the threshold. Certification depends on the independent calibration bound, not the test-based score.

CTS and harm are expressed as fractions in the formulas and percentages in the tables; the relative score is dimensionless. Each input receives equal weight within its configuration. We compute the nonlinear score and reference normalization separately for each configuration, then average all twelve scores with equal weights. The recalculation uses the displayed two-decimal CTS and Harm; final relative scores are rounded to three decimal places. A mean relative score of 1.060 is a score ratio, not a 6% token-cost reduction. The raw CTS and Harm remain available in the main tables.

D.3 Paired Bootstrap

For each configuration, both policies are evaluated on the same 180 test inputs. Each of 10,000 bootstrap replicates samples 180 matched input IDs with replacement. For CTS, harm, and task score, the replicate difference is the mean of the paired per-input differences. The auxiliary bootstrap score is $T^{\mathrm{soft}}_{\alpha}(\pi) = \mathrm{CTS}(\pi)/(1 + \widehat{H}_\epsilon(\pi)/\alpha)$; it is a continuous discount and can remain positive above the risk limit. CTS and harm are recomputed within each resample before evaluating this auxiliary score. The 2.5th and 97.5th percentiles give the reported intervals, using seed 20260921. Intervals condition on the fitted and selected policies and are computed separately for each comparison without multiplicity correction. The relative-score recalculation is descriptive; no bootstrap intervals or significance claims for $T^{\mathrm{rel}}_{0.10}$ are reported, and the auxiliary intervals cannot be transferred to it.

Figure S1 shows the bootstrap distributions of CTS and harm differences against certified fixed budgets. The box spans the interquartile range, the internal line marks the median, the dot marks the observed mean difference, and the whiskers span the percentile 95% interval. Table S3 reports the corresponding $T^{\mathrm{soft}}_{0.10}$ comparisons, and Table S4 reports task-score differences.

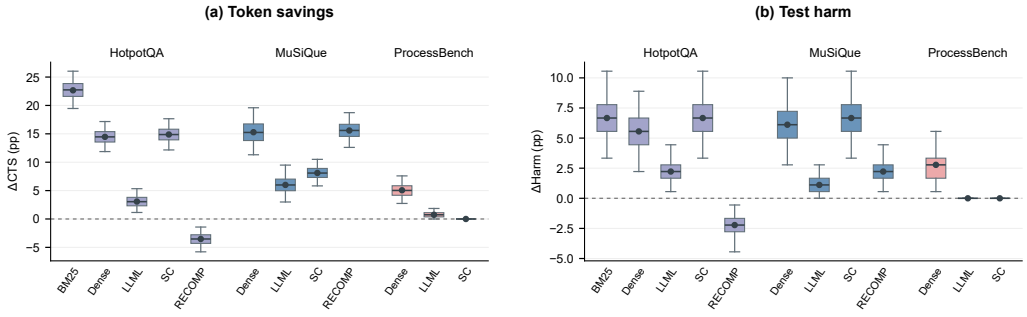

Figure S1: Paired bootstrap distributions of DiffContext minus certified fixed budgets: (a) CTS and (b) test harm, in percentage points. Positive CTS differences indicate more savings; negative harm differences indicate less observed harm. Boxes show interquartile ranges, lines indicate medians, dots mark observed mean differences, and whiskers span percentile 95% intervals from 10,000 paired resamples of 180 test inputs per configuration. These intervals describe CTS and harm, not the relative score. LLML: LongLLMLingua; SC: Selective Context.

Table S3: Paired differences in the auxiliary soft score $T^{\text{soft}}_{0.10}$: DiffContext minus each baseline, in percentage points. Intervals are percentile 95% intervals from 10,000 matched-input bootstrap resamples; the soft score is recomputed in every resample. These intervals do not apply to the relative score $T^{\text{rel}}_{0.10}$.

Dataset	Compressor	Baseline	ΔT^{soft}	95% CI
HotpotQA	BM25	Fixed	13.60	[10.79, 17.34]
	BM25	Budget-only	1.11	[-1.82, 4.19]
	BM25	Length-only	2.46	[0.07, 5.16]
	Dense	Fixed	9.30	[7.49, 11.69]
	Dense	Budget-only	3.55	[0.73, 6.61]
	Dense	Length-only	-3.24	[-6.13, -0.70]
	LongLLMLingua	Fixed	2.51	[1.06, 3.94]
	LongLLMLingua	Budget-only	0.75	[-1.41, 2.84]
	LongLLMLingua	Length-only	-2.97	[-5.07, -0.92]
	Selective Context	Fixed	8.93	[7.10, 11.29]
	Selective Context	Budget-only	1.31	[-1.55, 4.19]
	Selective Context	Length-only	-1.20	[-4.10, 1.55]
	RECOMP	Fixed	0.73	[-1.57, 3.46]
	RECOMP	Budget-only	0.73	[-1.57, 3.46]
	RECOMP	Length-only	-0.92	[-4.44, 2.43]
MuSiQue	Dense	Fixed	9.49	[7.35, 12.16]
	Dense	Budget-only	3.60	[0.66, 6.75]
	Dense	Length-only	0.78	[-1.62, 3.07]
	LongLLMLingua	Fixed	5.40	[2.83, 8.19]
	LongLLMLingua	Budget-only	2.31	[-0.80, 5.80]
	LongLLMLingua	Length-only	-1.58	[-3.25, -0.27]
	Selective Context	Fixed	4.87	[3.68, 6.32]
	Selective Context	Budget-only	1.57	[-0.12, 3.41]
	Selective Context	Length-only	3.40	[1.99, 4.99]
	RECOMP	Fixed	12.75	[9.97, 16.02]
	RECOMP	Budget-only	-3.89	[-7.84, -0.39]
	RECOMP	Length-only	4.54	[1.57, 7.76]
ProcessBench	Dense	Fixed	3.96	[2.37, 5.65]
	Dense	Budget-only	-2.05	[-4.34, 0.15]
	Dense	Length-only	-1.07	[-2.94, 0.70]
	LongLLMLingua	Fixed	0.74	[0.00, 1.85]
	LongLLMLingua	Budget-only	-0.84	[-2.29, 0.48]
	LongLLMLingua	Length-only	-4.65	[-6.25, -3.20]
	Selective Context	Fixed	0.00	[0.00, 0.00]
	Selective Context	Budget-only	-5.24	[-6.94, -3.78]
	Selective Context	Length-only	-6.25	[-8.25, -4.47]

Table S4: Paired task-score differences: DiffContext minus each baseline (percentage points), with percentile 95% intervals from 10,000 matched-input bootstrap resamples.

Dataset	Compressor	Baseline	Difference	95% CI
HotpotQA	BM25	Fixed	-4.58	[-8.14, -1.45]
	BM25	Budget-only	0.13	[-3.33, 3.41]
	BM25	Length-only	-0.52	[-2.59, 1.37]
	Dense	Fixed	-4.20	[-7.47, -1.23]
	Dense	Budget-only	-2.63	[-6.00, 0.70]
	Dense	Length-only	-3.19	[-6.38, -0.37]
	LongLLMLingua	Fixed	-1.20	[-2.78, -0.19]
	LongLLMLingua	Budget-only	-0.28	[-2.22, 1.67]
	LongLLMLingua	Length-only	3.72	[0.72, 6.77]
	Selective Context	Fixed	-2.16	[-4.84, 0.49]
	Selective Context	Budget-only	0.76	[-3.35, 4.77]
	Selective Context	Length-only	-0.56	[-3.88, 2.78]
	RECOMP	Fixed	1.81	[0.14, 3.89]
	RECOMP	Budget-only	1.81	[0.14, 3.89]
	RECOMP	Length-only	-0.02	[-1.79, 1.78]
MuSiQue	Dense	Fixed	-4.06	[-7.03, -1.57]
	Dense	Budget-only	-2.34	[-5.44, 0.57]
	Dense	Length-only	-1.48	[-5.02, 1.98]
	LongLLMLingua	Fixed	-0.63	[-1.70, 0.00]
	LongLLMLingua	Budget-only	1.54	[-0.52, 3.97]
	LongLLMLingua	Length-only	0.86	[0.00, 2.27]
	Selective Context	Fixed	-3.89	[-7.02, -1.17]
	Selective Context	Budget-only	-4.00	[-7.45, -0.92]
	Selective Context	Length-only	-3.70	[-6.84, -0.94]
	RECOMP	Fixed	-0.65	[-3.01, 1.66]
	RECOMP	Budget-only	-0.17	[-3.56, 3.19]
	RECOMP	Length-only	-3.40	[-6.38, -0.68]
ProcessBench	Dense	Fixed	-1.67	[-4.44, 1.11]
	Dense	Budget-only	1.11	[-3.33, 5.56]
	Dense	Length-only	-1.11	[-5.00, 2.78]
	LongLLMLingua	Fixed	0.00	[0.00, 0.00]
	LongLLMLingua	Budget-only	0.00	[0.00, 0.00]
	LongLLMLingua	Length-only	4.44	[0.56, 8.89]
	Selective Context	Fixed	0.00	[0.00, 0.00]
	Selective Context	Budget-only	5.00	[2.22, 8.33]
	Selective Context	Length-only	3.33	[-0.56, 7.22]

E Complete Policy Results and Feature Controls

E.1 Complete Test Results

Tables S5–S7 give CTS, test harm, $T^{\mathrm{rel}}_{0.10}$, context-token reduction, coverage, and task score for DiffContext and the three certified comparison pipelines. All use the same 180 test inputs within a configuration. Table S8 additionally reports DiffContext harm counts and inputs for which both DiffContext and certified fixed choose full context. In ProcessBench–Selective Context, the selected DiffContext policy assigns full context to every test input; this describes its decisions rather than a failure of every candidate's calibration test.

Table S9 collects the relative scores for all six main comparison policies, including fit-only selection, and reports configuration-level win, tie, and loss counts.

E.2 Feature-Control Construction

Budget-only estimates its budget-specific risks from the empirical prevalence of fit labels. Because it retains the input-dependent cost branch, constant risk predictions at a given budget need not produce identical decisions across inputs. Length-only selects a learner and Platt map from the same search space as the rich-feature predictor. Both controls undergo the same candidate construction and independent risk certification as DiffContext. Figure S2 compares dataset-level mean test CTS for DiffContext against each control separately. The full feature set does not show a consistent advantage: Length-only has higher mean CTS and higher mean observed harm across configurations, while Budget-only is close to DiffContext in mean CTS. Table S10 reports CTS, harm, and relative scores for every configuration.

Table S5: Complete certified-policy results on HotpotQA. CTS, Harm, TRed, Coverage, and task Score are percentages. The dimensionless $T_{0.10}^{\mathrm{rel}}$ uses the configuration's certified fixed-budget policy as reference (1).

Compressor	Policy	CTS	Harm	$T_{0.10}^{\mathrm{rel}}$	TRed	Coverage	Score
BM25	DiffContext	22.66	6.67	1.136	23.58	55.00	73.20
BM25	Fixed	0.00	0.00	1.000	0.00	0.00	77.78
BM25	Budget-only	21.51	7.22	1.125	22.33	78.89	73.07
BM25	Length-only	17.95	6.11	1.111	18.64	48.89	73.72
Dense	DiffContext	14.47	5.56	1.093	15.15	42.22	73.58
Dense	Fixed	0.00	0.00	1.000	0.00	0.00	77.78
Dense	Budget-only	7.35	2.78	1.058	7.58	14.44	76.21
Dense	Length-only	18.12	4.44	1.125	19.06	54.44	76.78
LongLLMLingua	DiffContext	3.07	2.22	1.025	3.16	4.44	76.58
LongLLMLingua	Fixed	0.00	0.00	1.000	0.00	0.00	77.78
LongLLMLingua	Budget-only	1.95	1.11	1.018	2.00	2.22	76.86
LongLLMLingua	Length-only	9.74	7.78	1.055	10.07	14.44	72.86
Selective Context	DiffContext	14.88	6.67	1.089	15.87	38.89	75.62
Selective Context	Fixed	0.00	0.00	1.000	0.00	0.00	77.78
Selective Context	Budget-only	12.71	6.67	1.076	13.38	32.78	74.86
Selective Context	Length-only	15.76	5.56	1.101	16.81	73.89	76.17
RECOMP	DiffContext	39.81	7.78	1.006	41.37	89.44	74.04
RECOMP	Fixed	43.33	10.00	1.000	45.00	100.00	72.23
RECOMP	Budget-only	43.33	10.00	1.000	45.00	100.00	72.23
RECOMP	Length-only	38.86	6.67	1.014	40.33	85.56	74.06

Table S6: Complete certified-policy results on MuSiQue. CTS, Harm, TRed, Coverage, and task Score are percentages. The dimensionless $T_{0.10}^{\mathrm{rel}}$ uses the configuration's certified fixed-budget policy as reference (1).

Compressor	Policy	CTS	Harm	$T_{0.10}^{\mathrm{rel}}$	TRed	Coverage	Score
Dense	DiffContext	15.29	6.11	1.095	15.44	23.33	38.89
Dense	Fixed	0.00	0.00	1.000	0.00	0.00	42.96
Dense	Budget-only	7.52	2.78	1.059	7.65	15.00	41.23
Dense	Length-only	14.52	6.67	1.087	14.43	47.78	40.37
LongLLMLingua	DiffContext	6.00	1.11	1.054	6.09	6.67	42.33
LongLLMLingua	Fixed	0.00	0.00	1.000	0.00	0.00	42.96
LongLLMLingua	Budget-only	3.96	2.78	1.031	4.03	4.44	40.79
LongLLMLingua	Length-only	8.54	2.22	1.070	8.66	9.44	41.47
Selective Context	DiffContext	8.11	6.67	1.049	8.39	22.22	39.06
Selective Context	Fixed	0.00	0.00	1.000	0.00	0.00	42.96
Selective Context	Budget-only	3.84	1.67	1.033	3.89	21.67	43.07
Selective Context	Length-only	1.63	1.11	1.015	1.75	9.44	42.77
RECOMP	DiffContext	15.58	2.22	1.127	15.93	35.56	42.31
RECOMP	Fixed	0.00	0.00	1.000	0.00	0.00	42.96
RECOMP	Budget-only	24.03	4.44	1.166	24.57	53.89	42.48
RECOMP	Length-only	9.58	1.67	1.082	9.84	32.22	45.71

E.3 Evaluation Records

The unified-grid comparison retains the existing fit, calibration, and test input IDs. Available input–budget outcomes and full-context references are reused, and missing outcomes at the added budgets are evaluated to complete the grid. The original model search and candidate-construction rule are retained. The unified-grid and risk-limit analyses are retrospective analyses of these existing splits.

Table S7: Complete certified-policy results on ProcessBench. CTS, Harm, TRed, Coverage, and task Score are percentages. The dimensionless $T^{\mathrm{rel}}_{0.10}$ uses the configuration's certified fixed-budget policy as reference (1).

Compressor	Policy	CTS	Harm	$T^{\mathrm{rel}}_{0.10}$	TRed	Coverage	Score
Dense	DiffContext	5.06	2.78	1.040	6.33	8.33	46.67
Dense	Fixed	0.00	0.00	1.000	0.00	0.00	48.33
Dense	Budget-only	8.68	4.44	1.060	10.94	14.44	45.56
Dense	Length-only	6.42	2.78	1.050	7.74	10.00	47.78
LongLLMLingua	DiffContext	0.74	0.00	1.007	0.87	1.11	48.33
LongLLMLingua	Fixed	0.00	0.00	1.000	0.00	0.00	48.33
LongLLMLingua	Budget-only	1.58	0.00	1.016	1.95	2.78	48.33
LongLLMLingua	Length-only	8.69	6.11	1.054	10.64	14.44	43.89
Selective Context	DiffContext	0.00	0.00	1.000	0.00	0.00	48.33
Selective Context	Fixed	0.00	0.00	1.000	0.00	0.00	48.33
Selective Context	Budget-only	7.87	5.00	1.052	10.04	14.44	43.33
Selective Context	Length-only	9.37	5.00	1.062	11.44	15.56	45.00

Table S8: Observed DiffContext harm counts and inputs assigned full context by both DiffContext and certified fixed.

Dataset	Compressor	Harm cases	Both full
HotpotQA	BM25	12	81
	Dense	10	104
	LongLLMLingua	4	172
	Selective Context	12	110
	RECOMP	14	0
MuSiQue	Dense	11	138
	LongLLMLingua	2	168
	Selective Context	12	140
	RECOMP	4	116
ProcessBench	Dense	5	165
	LongLLMLingua	0	178
	Selective Context	0	180

Table S9: Dimensionless relative scores $T^{\mathrm{rel}}_{0.10}$ for all twelve configurations, recomputed from the displayed aggregate CTS and Harm. Each row uses its certified fixed-budget policy as reference (1). Higher is preferred under the risk-priority scoring rule. These point estimates do not establish risk certification or statistical significance.

Dataset	Compressor	Fixed 0.25	Fixed 0.50	Fixed 0.75	Certified fixed	Fit-only	DiffContext
HotpotQA	BM25	0.474	1.260	1.144	1.000	1.202	1.136
	Dense	0.450	0.667	1.167	1.000	0.581	1.093
	LongLLMLingua	0.214	0.247	0.391	1.000	0.667	1.025
	Selective Context	0.305	0.514	1.109	1.000	0.529	1.089
	RECOMP	0.510	1.000	0.934	1.000	1.015	1.006
MuSiQue	Dense	0.400	0.643	1.134	1.000	0.581	1.095
	LongLLMLingua	0.231	0.247	0.321	1.000	0.621	1.054
	Selective Context	0.237	0.340	0.857	1.000	0.600	1.049
	RECOMP	0.546	0.947	1.116	1.000	0.900	1.127
ProcessBench	Dense	0.273	0.290	0.375	1.000	0.621	1.040
	LongLLMLingua	0.225	0.231	0.237	1.000	1.055	1.007
	Selective Context	0.254	0.281	0.360	1.000	0.643	1.000
Mean across configurations		0.343	0.556	0.762	1.000	0.751	1.060
DiffContext wins / ties / losses		12 / 0 / 0	11 / 0 / 1	8 / 0 / 4	11 / 1 / 0	9 / 0 / 3	—

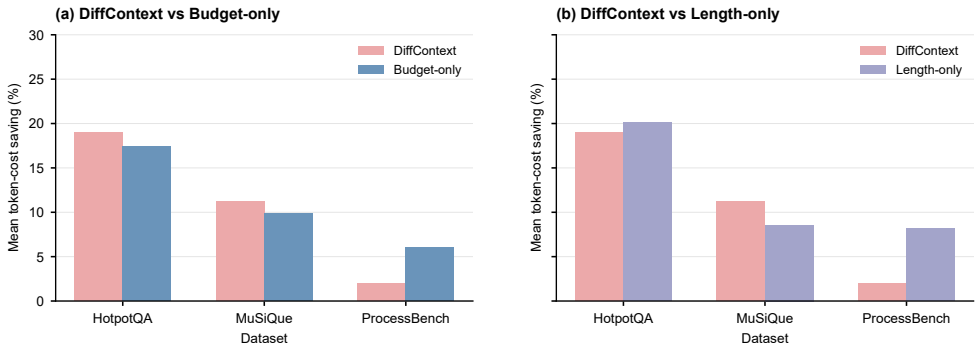

Figure S2: Token-cost saving (CTS) by dataset under the same independent certification protocol. (a) DiffContext (pink) versus Budget-only (blue). (b) DiffContext (pink) versus Length-only (purple). Bars average configuration-level test CTS equally across the five HotpotQA, four MuSiQue, and three ProcessBench compressors; each configuration uses 180 matched test inputs. These descriptive means do not represent uncertainty intervals. Observed harm and complete per-configuration results are reported in Table S10.

Table S10: Input-feature ablations under the same certification protocol. CTS and Harm are percentages on 180 matched test inputs per configuration; the relative score is dimensionless. CTS is token-cost saving, Harm is observed compression-induced harm, and $T^{\mathrm{rel}}_{0.10}$ is normalized to the configuration's certified fixed-budget policy (1). Budget-only and Length-only restrict harm-predictor features while retaining the cost estimator and the original fitting and policy-selection pipeline.

Dataset	Compressor	DiffContext			Budget-only			Length-only		
		CTS ↑	Harm ↓	$T^{\mathrm{rel}}_{0.10}$ ↑	CTS ↑	Harm ↓	$T^{\mathrm{rel}}_{0.10}$ ↑	CTS ↑	Harm ↓	$T^{\mathrm{rel}}_{0.10}$ ↑
HotpotQA	BM25	22.66	6.67	1.136	21.51	7.22	1.125	17.95	6.11	1.111
	Dense	14.47	5.56	1.093	7.35	2.78	1.058	18.12	4.44	1.125
	LongLLMLingua	3.07	2.22	1.025	1.95	1.11	1.018	9.74	7.78	1.055
	Selective Context	14.88	6.67	1.089	12.71	6.67	1.076	15.76	5.56	1.101
	RECOMP	39.81	7.78	1.006	43.33	10.00	1.000	38.86	6.67	1.014
MuSiQue	Dense	15.29	6.11	1.095	7.52	2.78	1.059	14.52	6.67	1.087
	LongLLMLingua	6.00	1.11	1.054	3.96	2.78	1.031	8.54	2.22	1.070
	Selective Context	8.11	6.67	1.049	3.84	1.67	1.033	1.63	1.11	1.015
	RECOMP	15.58	2.22	1.127	24.03	4.44	1.166	9.58	1.67	1.082
ProcessBench	Dense	5.06	2.78	1.040	8.68	4.44	1.060	6.42	2.78	1.050
	LongLLMLingua	0.74	0.00	1.007	1.58	0.00	1.016	8.69	6.11	1.054
	Selective Context	0.00	0.00	1.000	7.87	5.00	1.052	9.37	5.00	1.062

F CALIBRATION OUTCOMES AND RISK-LIMIT SENSITIVITY

Table S11 reports the selected policies' calibration outcomes at the main 10% limit. With 180 calibration inputs and $\delta = 0.01$, eight harm events give an upper bound of approximately 9.42%, while nine give approximately 10.15%. A candidate with eight events passes, whereas one with nine fails certification. The zero-risk full-context fallback is known from the harm definition and does not require estimation.

Risk-limit sensitivity varies α over $\{0.03, 0.05, 0.075, 0.10\}$ while retaining $\epsilon = 0.10$, $\delta = 0.01$, fitted predictors, and the original candidate order. At each limit, the procedure starts at the beginning of the sequence, stops at the first failed test, and returns the last passing policy or full context. Its performance is then measured on the matched test inputs. The reported CTS and harm in Table S12 are test outcomes; selection uses calibration outcomes.

Figure S3 shows all twelve configurations separately. The main figure averages their CTS values within each dataset, giving each compressor equal weight. Markers identify the four evaluated risk limits; connecting lines are visual guides. The certification guarantee applies separately at each fixed risk limit.

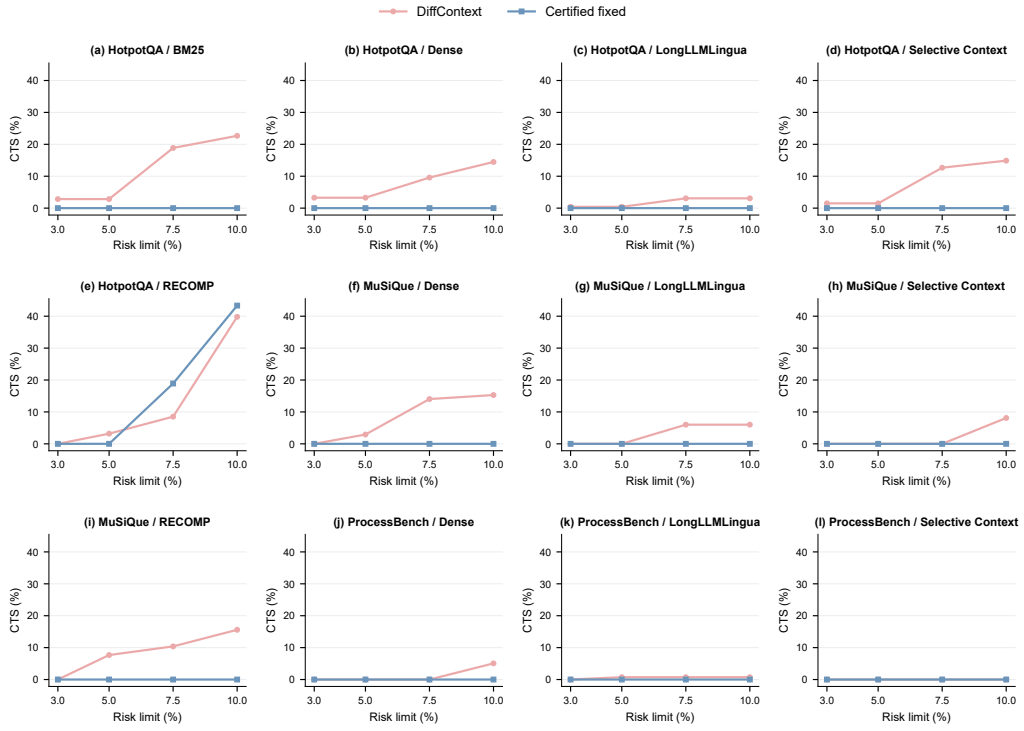

Figure S3: Configuration-level test CTS at certified risk limits of 3%, 5%, 7.5%, and 10%, corresponding to the dataset averages in Figure 3. Higher CTS indicates greater token savings. Candidate families and fitted predictors are held fixed; policies are recalibrated at each limit with $\delta = 0.01$. Lines connect evaluated settings.

Table S11: Selected calibration outcomes at $\alpha = 0.10$, $\delta = 0.01$, and $n = 180$. Rank refers to the frozen DiffContext sequence. Full-context fallback has known zero harm; other upper bounds are one-sided Clopper–Pearson bounds.

Dataset	Compressor	Policy	Selection	Harm count	Upper bound (%)
HotpotQA	BM25	DiffContext	Rank 7	7	8.67
	BM25	Certified fixed	Full	0	0.00
	Dense	DiffContext	Rank 5	7	8.67
	Dense	Certified fixed	Full	0	0.00
	LongLLMLingua	DiffContext	Rank 2	4	6.31
	LongLLMLingua	Certified fixed	Full	0	0.00
	Selective Context	DiffContext	Rank 5	8	9.42
	Selective Context	Certified fixed	Full	0	0.00
	RECOMP	DiffContext	Rank 10	7	8.67
	RECOMP	Certified fixed	0.50	8	9.42
MuSiQue	Dense	DiffContext	Rank 5	7	8.67
	Dense	Certified fixed	Full	0	0.00
	LongLLMLingua	DiffContext	Rank 2	3	5.47
	LongLLMLingua	Certified fixed	Full	0	0.00
	Selective Context	DiffContext	Rank 2	8	9.42
	Selective Context	Certified fixed	Full	0	0.00
	RECOMP	DiffContext	Rank 5	8	9.42
	RECOMP	Certified fixed	Full	0	0.00
ProcessBench	Dense	DiffContext	Rank 2	7	8.67
	Dense	Certified fixed	Full	0	0.00
	LongLLMLingua	DiffContext	Rank 1	1	3.63
	LongLLMLingua	Certified fixed	Full	0	0.00
	Selective Context	DiffContext	Rank 1	0	2.53
	Selective Context	Certified fixed	Full	0	0.00

Table S12: Risk-limit sensitivity with frozen candidates. Values are test CTS / test harm (percent), following calibration at each stated risk limit.

Dataset	Compressor	Policy	3%	5%	7.5%	10%
HotpotQA	BM25	DiffContext	2.83 / 0.56	2.83 / 0.56	18.86 / 5.56	22.66 / 6.67
	BM25	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	Dense	DiffContext	3.26 / 0.00	3.26 / 0.00	9.58 / 3.33	14.47 / 5.56
	Dense	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	LongLLMLingua	DiffContext	0.41 / 0.56	0.41 / 0.56	3.07 / 2.22	3.07 / 2.22
	LongLLMLingua	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	Selective Context	DiffContext	1.50 / 0.56	1.50 / 0.56	12.67 / 4.44	14.88 / 6.67
	Selective Context	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	RECOMP	DiffContext	0.00 / 0.00	3.19 / 1.67	8.51 / 2.78	39.81 / 7.78
	RECOMP	Certified fixed	0.00 / 0.00	0.00 / 0.00	18.91 / 3.89	43.33 / 10.00
MuSiQue	Dense	DiffContext	0.00 / 0.00	2.92 / 0.56	14.04 / 5.00	15.29 / 6.11
	Dense	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	LongLLMLingua	DiffContext	0.00 / 0.00	0.00 / 0.00	6.00 / 1.11	6.00 / 1.11
	LongLLMLingua	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	Selective Context	DiffContext	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	8.11 / 6.67
	Selective Context	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	RECOMP	DiffContext	0.00 / 0.00	7.67 / 1.11	10.39 / 1.67	15.58 / 2.22
	RECOMP	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
ProcessBench	Dense	DiffContext	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	5.06 / 2.78
	Dense	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	LongLLMLingua	DiffContext	0.00 / 0.00	0.74 / 0.00	0.74 / 0.00	0.74 / 0.00
	LongLLMLingua	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	Selective Context	DiffContext	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
	Selective Context	Certified fixed	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00

G Task-Model Expenditure and Fit-Label Amortization

G.1 Token Pricing

The pricing scenario uses USD 0.115 per million input tokens and USD 0.287 per million output tokens. For instance i and policy π, the task-model expenditure is

$$d_i^\pi = \frac{0.115\, N_{i,\mathrm{in}}^\pi + 0.287\, N_{i,\mathrm{out}}^\pi}{10^6}. \tag{23}$$

These prices are accounting assumptions. Let $\bar{d}_\pi = n^{-1} \sum_i d_i^\pi$ and define $\bar{d}_{\text{full}}$ analogously. The online expenditure ratio is $\bar{d}_\pi / \bar{d}_{\text{full}}$. Unlike CTS, it uses separately priced input and output tokens and is a ratio of aggregate expenditure. Table S13 gives the mean per-input amounts and their ratios.

Table S13: Mean task-model token expenditure per input (USD) and the online/full-context ratio. This table excludes compressor compute.

Dataset	Compressor	Full (USD)	Online (USD)	Ratio
HotpotQA	BM25	0.000169	0.000128	0.757
	Dense	0.000169	0.000147	0.871
	LongLLMLingua	0.000169	0.000164	0.969
	Selective Context	0.000169	0.000143	0.845
	RECOMP	0.000169	0.000104	0.613
MuSiQue	Dense	0.000315	0.000257	0.818
	LongLLMLingua	0.000315	0.000287	0.911
	Selective Context	0.000315	0.000283	0.899
	RECOMP	0.000315	0.000272	0.865
ProcessBench	Dense	0.001035	0.000973	0.940
	LongLLMLingua	0.001035	0.001021	0.986
	Selective Context	0.001035	0.001035	1.000

G.2 Fit-Label Expenditure

Let F_{fit} denote the task-model expenditure for evaluating the four budgets on the fit split, counting the full-context reference once per fit input. For a deployment volume of N inputs, the amortized ratio is

$$A(N) = \frac{\bar{d}_\pi + F_{\text{fit}}/N}{\bar{d}_{\text{full}}}. \tag{24}$$

When $\bar{d}_\pi < \bar{d}_{\text{full}}$, the smallest integer break-even volume is

$$N^* = \left\lceil \frac{F_{\text{fit}}}{\bar{d}_{\text{full}} - \bar{d}_\pi} \right\rceil. \tag{25}$$

Otherwise there is no finite break-even volume for positive F_{fit}. Table S14 lists the fit-label expenditure and break-even volumes. The curves in Figure 4 summarize configuration-level ratios by dataset medians and interquartile ranges; these bands describe variation across compressors rather than confidence intervals. This accounting includes task-model tokens for fit-label generation and online inference. Compressor compute, independent calibration, predictor fitting, and controller execution are outside its scope.

Table S14: Task-model token expenditure for fit-label generation and its break-even deployment volume under the stated token prices. Compressor compute and other preparation costs are excluded.

Dataset	Compressor	Fit inputs	Fit actions	Fit labels (USD)	Break-even inputs
HotpotQA	BM25	240	960	0.1007	2453
	Dense	90	360	0.0388	1777
	LongLLMLingua	90	360	0.0332	6349
	Selective Context	90	360	0.0434	1658
	RECOMP	90	360	0.0420	643
MuSiQue	Dense	90	360	0.0708	1234
	LongLLMLingua	90	360	0.0535	1907
	Selective Context	240	960	0.1936	6087
	RECOMP	240	960	0.1889	4439
ProcessBench	Dense	90	360	0.2031	3282
	LongLLMLingua	90	360	0.1988	13945
	Selective Context	90	360	0.2191	–