OPTIMAL DEPTH OF NEURAL NETWORKS: RESOURCE-CORRECT STOPPING WITH BOUNDED LOGIT INCREMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

When can a network stop without executing the suffix that justifies the stop? For a fixed inspection schedule, finite observed record, confidence-resource utility, declared charges, and coupled bounded logit increments, we derive an information-optimal stopping certificate. One aligned completion attains every future confidence support. A separate sharp guard preserves the full-recall returned label, not unknown-label correctness. We compile both tests into checked integer kernels and prove their equivalence to arbitrary-precision comparisons. Five-seed experiments select schedules separately for scalar and return-conjoined policies, publicly freezing 870 choices before new evaluation. Return protection changes two of 300 main-grid schedule choices and eliminates all returned-label disagreements; precision and temperature are evaluated with newly constructed scores and supports. Actual image-to-return execution on Xeon and Neoverse CPUs makes the compiled multi-exit path faster than full execution for the registered 1,000-image workload, while retaining adverse small-batch results. A finite-family work bound and a separated design procedure distinguish pointwise certification, task risk, and statistical selection. All prior proofs, saturated-training diagnostics, and unfavorable results are preserved behind an active-claim and artifact index.

1 INTRODUCTION

Adaptive depth needs a prediction and a reason not to execute another layer. Learned halting and routing estimate task–compute trade-offs (Graves, 2016; Teerapittayanon et al., 2016; Huang et al., 2018; Wang et al., 2018). A certificate asks whether *any* suffix consistent with the information already available can improve the declared objective enough to matter. Recall makes the accounting nontrivial: executing three corrections and returning the prefix prediction still pays for all three corrections and every inspection.

Our architectural contract bounds additive corrections to persistent logits. Raising one currently maximal coordinate and lowering every competitor attains all future confidence supports simultaneously. This coupled witness proves soundness and information-relative maximality, not merely a sufficient bound. With fixed inspection order, full finite observed information, specified confidence-resource utility and charges, the first successful certificate is the minimum certifiable executed depth. This is an operational notion of optimal depth, not optimal architecture search or unknown-label correctness.

Three distinctions determine whether the theorem becomes useful computation. First, current confidence alone loses the recall and past-regret state; a general controller must retain it. Second, a score-regret gate need not preserve the full network's returned label. We provide a separate sharp return guard and select schedules for its conjunction with the scalar test rather than attach a task guarantee to scalar-only schedule results. Third, declared work savings need not reduce runtime. We compile the exact history-aware tests, prove a finite integer-width bound, and time actual masked image inference rather than substitute the first-exit shortcut's timing for a richer controller.

The contribution is a sharp contract-to-controller construction with three separate layers: pointwise stopping and return certificates; training-only selection within a fixed schedule family; and measured implementation cost. Controlled five-seed training repairs severe saturation without relaxing the output bound. New guarded-schedule, precision/temperature, and two-CPU studies connect that

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.

construction to decisions and execution. The main text contains the active results; a one-page appendix map separates current proofs, full tables, runnable evidence, and byte-preserved archival developments.

2 Related work

The depth-stopping formulation of Qi (2025) motivates the with-recall problem. Adaptive prediction and conditional computation (Bolukbasi et al., 2017; Graves, 2016; Huang et al., 2018; Wang et al., 2018) learn task–compute trade-offs. Regression and neural optimal stopping (Longstaff & Schwartz, 2001; Becker et al., 2019) estimate continuation values; recall-aware threshold models study loss trajectories (Yang et al., 2026b;a). Our guarantee instead quantifies all completions of an architectural increment contract, conditional on an observed prefix. Statistical early-exit prediction sets and compute-risk control (Jazbec et al., 2024; Wang et al., 2026) provide different sampling-based guarantees. A score-regret certificate is not a replacement for their population-risk statements.

Elboher et al. (2025) directly verify early-exit networks, including conditional exit guards and local input robustness. This is a close antecedent. A general verifier can encode our recall and regret predicate; our contribution is the sharp closed form obtained from the output contract, not a new generic solver or an input-robustness theorem. Interval bounds and abstract interpretation (Cousot & Cousot, 1977; Gowal et al., 2018), CROWN and optimized linear relaxations (Zhang et al., 2018; Xu et al., 2021), and split-domain verification (Wang et al., 2021) exploit more of the trained network. We compare the authors' auto_LiRPA implementations on the same Fashion-MNIST suffix and conditional domains. They have access to weights and hidden-state boxes that the structural rule deliberately ignores. The earlier MNIST comparison and historical grid, proof-library, and shortest-path developments remain in the appendices, without being relabeled as new experiments.

3 Sharp stopping with recall and inspection charges

A network has exits $0, \dots, L$, scores $g_j \in [0,1]$, and logits $z_j \in \mathbb{R}^K$. Write $C_0 = 0$, $C_t = \sum_{s=1}^{t} c_s$, $c_s \geq 0$, $B_t = \max_{j \leq t} g_j$, and $R_t = B_t - C_t$. The returned exit is the shallowest maximizer of the observed score; the executed depth is t. Let $Z_t \geq 0$ be the cumulative charge for all inspections through t, including the current check and earlier rejections. We compare the charged candidate $R_t - Z_t$ with the uncharged network-depth comparator $\max_{0 \leq j \leq L} R_j$. This demanding comparator does not silently give competing depths the controller's overhead. Its regret is

$$\begin{aligned} \Gamma_t^Z &= \max_{0 \leq j \leq L} R_j - R_t + Z_t = \max\{P_t, S_t\} + Z_t, \qquad P_t = \max_{j \leq t} R_j - R_t, \\ S_t &= \Big[\max_{j > t} (g_j - B_t - D_j) \Big]_+ . \end{aligned} \tag{1}$$

where $D_j = C_j - C_t$. Nonnegative transition costs justify the reduction: a future record is best charged when it appears, and an old record cannot create positive continuation improvement. The past term P_t is already observed; no estimate of the suffix can repair it.

The architectural contract is

$$z_j = z_{j-1} + \eta_j, \qquad \|\eta_j\|_\infty \leq \beta_j, \qquad r_j = \sum_{s=t+1}^{j} \beta_s. \tag{2}$$

Let $\mathcal{A}_t$ contain every completion satisfying this coupled contract, not just those reachable through particular trained weights. For max-softmax confidence, put $p = \max_k \operatorname{softmax}(z_t)_k$ and

$$Q_j(p) = \frac{e^{2r_j} p}{1 + (e^{2r_j} - 1)p}, \qquad U_t = \Big[\max_{j > t} \{ Q_j(p) - B_t - D_j \} \Big]_+ . \tag{3}$$

An empty future maximum contributes zero.

Theorem 1 (Sharp charged certificate)**.** *Under* (2), $\sup_{\mathcal{A}_t} \Gamma_t^Z = \max\{P_t, U_t\} + Z_t$. *Thus stopping is sound at tolerance λ exactly when*

$$\max\{P_t, U_t\} + Z_t \leq \lambda. \tag{4}$$

No rule using only the same observed scores and logits, increment budgets, charges, and comparator can soundly accept a larger set of prefixes. On a fixed inspection schedule, the first successful test has the smallest certifiable executed depth along that observed trajectory.

Proof. For a fixed class, its softmax probability increases with its own logit and decreases with every competitor. Raising it by r_j and lowering all others by r_j gives the largest class probability in the endpoint cube. Choosing a currently maximal class maximizes that expression over classes and yields $Q_j(p)$. Apply the same signed increment of magnitude β_s at every stage s. This one admissible sequence attains all Q_j simultaneously, so the maximum of the endpoint improvements in (3) is attained by a coupled completion. The past and inspection terms are fixed. If (4) fails, this completion is a witness against any rule with the stated information. Earliest acceptance follows on the fixed schedule; it does not compare different inspection schedules. □

This distinction between a certificate and a fallback is important. An input that fails every inspection executes the final stage and returns its best observed prediction. That fallback is *not* automatically certified: $P_L + Z_L$ can exceed λ even though there is no remaining suffix. Our implementations and tables distinguish certified early stops from compulsory final execution. Charging an inspection before acceptance also means that $Z_t > \lambda$ precludes any certificate, irrespective of confidence.

3.1 When the rule becomes a confidence threshold

At a record prefix $B_t = p$, each future constraint has the form $h_s(p) \le a$, where $s = e^{2r_j}$, $a = \lambda - Z_t + D_j$, and

$$h_s(p) = \frac{(s-1)p(1-p)}{1+(s-1)p}, \quad \max_{p\in[0,1]} h_s(p) = \frac{\sqrt{s}-1}{\sqrt{s}+1}, \quad p_\pm = \frac{1-a \pm \sqrt{(1-a)^2 - 4a/(s-1)}}{2}. \tag{5}$$

For a nonvacuous positive constraint, acceptance lies below p_- or above p_+. Intersect these sets over future exits and enforce the observed-past constraint. On $p \ge 1/2$, the future part is a single upper-threshold test; its cutoff can be compiled offline. Retaining only this branch is conservative when a lower branch exists. At nonrecord prefixes, replacing B_t by current p or discarding P_t changes the problem. Our multi-exit controller therefore retains the score record and past net rewards rather than applying a first-exit cutoff at every stage.

4 Finite implementation and task-level return preservation

Deployment uses $S = 64$ logit units, prefix integers $q_0 \in [-512, 512]^K$, and additive updates bounded by 16 units when $\beta = 0.25$. Let $M = 2^{48}$ and $E^-[d]$ be the committed monotone integer lower approximation of $Me^{-d/S}$, with $E^-[0] = M$. Define

$$d(q) = \sum_k E^-[\max_\ell q_\ell - q_k], \qquad \widehat{g}(q) = M/d(q). \tag{6}$$

Rational Taylor bounds verify the table independently; the uniform score error is at most $\epsilon_E = 730/M < 2.594 \cdot 10^{-12}$ on the declared domain. A regret statement transfers to real softmax with $2\epsilon_E$ added to its tolerance. The finite claim itself uses the rational score, not rounded floating-point softmax comparisons.

For a remaining integer radius b, add b to a currently maximal logit and subtract b from the others. Monotonicity of the table gives the exact finite-score endpoint support, attained simultaneously by aligned integer updates. A score difference is compared with rational $a = u/v$ through $Mv(d_2 - d_1) \le ud_1d_2$, using exact integer arithmetic. No tolerance is introduced to decide acceptance. Prefix normalization and table lookup are real operations; the "one comparison" first-exit implementation refers only to the additional cutoff decision after normalization and offline compilation.

Proposition 2 (Compiled arithmetic refinement)**.** *For ten classes, four exits, $S \in \{16, 32, 64\}$, $q_0 \in [-8S, 8S]^{10}$, per-stage integer budget $S/4$, a valid monotone table with $E[0] = M$, and rational rule operands $|u|, v \le 2^{40}$ with $v > 0$, the checked 256-bit kernel makes exactly the arbitrary-precision scalar and return decisions at every legal prefix. Invalid table, logit or operand domains fail closed through the checked interface.*

Every endpoint gap is below $32S$, every denominator below 2^{52}, and the largest cross product below 2^{144}. Thus compilation introduces no rounding or overflow within this validated domain (Appendix B.2). The kernel retains nonrecord history, earliest ties and all rejected-check charges. It is a general exact gate, not a one-comparison cutoff: normalization, table validation, future supports, return tests and masking are still paid operations. The refinement assumes live correctly shaped buffers and a faithful compiler; it is not a whole-software or operating-system proof.

4.1 Why recall does not prevent compilation

For a fixed utility, the complete observed record has a sufficient recurrence. Let a_t be the earliest label attaining B_t, and define the largest past net reward $A_t = \max_{j \le t}(B_j - C_j)$. On observing the next score,

$$B_{t+1} = \max\{B_t, g_{t+1}\}, \qquad A_{t+1} = \max\{A_t, B_{t+1} - C_{t+1}\}, \qquad P_t = A_t - B_t + C_t. \quad (7)$$

Update a_t only on a strict score increase, and add every attempted fee to Z_t before testing. Induction shows that $(q_t, B_t, a_t, A_t, Z_t)$ reproduces the full observed comparator and return guard; no approximation to the history is necessary. It is $O(K)$ scalar/logit state excluding hidden activations, with bit cost depending on the rational representation. Changing the utility after discarding history need not preserve this sufficiency.

The implemented four-exit kernel instead reconstructs the same quantities from its short history buffer, which keeps the implementation auditable and allows direct comparison with the independent reference. At each scheduled inspection it validates the state, builds the four rational rule pairs, compares past records and future endpoints, and optionally tests alternate returns. Accepted rows are removed before another residual stage; all others keep their paid check count. This remains exact at nonrecord prefixes. A single confidence cutoff cannot in general encode the independent past-regret and record-confidence state, but those extra rational comparisons do not prevent compilation. The measured implementation and the sufficient-state alternative are explicitly distinguished, rather than attributing an unimplemented memory optimization to the timing result.

4.2 A sharp guarantee for the returned decision

Confidence regret and task loss are different quantities. A useful bridge is to preserve the decision made by the entire full-recall network, without requiring every intermediate head to keep the same label. Let a be the label currently recalled, let $v = \max_{k \ne a} \mathrm{softmax}(z_t)_k$, and put $r = \sum_{s>t} \beta_s$. The current prefix need not be a record.

Theorem 3 (Full-recall return preservation)**.** *With earliest-exit tie-breaking, every completion in $\mathcal{A}_t$ has full-recall returned label a if and only if*

$$\frac{e^{2r} v}{1 + (e^{2r} - 1)v} \le B_t. \quad (8)$$

When this holds, stopping preserves $\ell(a, y)$ relative to full recall for every *possible true label y and every loss depending only on the returned decision. It does not establish that $a = y$ or preserve a probability-dependent loss such as cross-entropy.*

Proof. Every future competing-class probability is at most its boosted endpoint probability, and the largest such value is the left side of (8). If it is at most B_t, no competing label can create a strictly higher record; equality preserves the earlier observed return. If it exceeds B_t, boost a maximizing competing coordinate and lower all others at every remaining step. That coordinate's probability increases to the endpoint, while each other probability decreases from a value at most B_t. The competing coordinate is therefore the unique endpoint winner and creates a strictly better record, producing a violating completion. This one coupled trajectory proves necessity as well as an endpoint bound. The argument uses the complete observed record but does not require the current prefix itself to be a record. □

Complete finite-table and tie details are in Appendix G.

The exact finite analogue checks alternate-class boosted endpoints: a harmful endpoint has a label other than a and a denominator strictly smaller than the observed record. Testing all alternate classes

Table 1: Optimization under the same $\beta = 0.25$ deployment contract, five paired seeds. Accuracy is in percent; saturation is the percentage of final-stage coordinates above 99% of the cap magnitude before rounding. Every registered endpoint is retained. The unconstrained row is a comparator, not a certified bounded model.

Readout/training	Epochs	Terminal	Recall	Saturation
Hard, continued	12	85.416	85.410	100.0
Small init., hard	4	86.170	85.866	84.1
Small init., hard	12	86.472	86.172	86.1
Small init., smooth	4	86.308	85.980	23.1
Small init., smooth	12	86.712	86.348	50.3
Straight-through	4	85.444	85.420	56.4
Straight-through	12	85.600	85.518	10.3
Projected + straight-through	12	85.576	85.538	37.9
Unbounded, continued	12	86.854	86.706	–
Unbounded then projected	12	86.276	85.894	61.0

handles quantization and the deterministic class tie rule. Conjoining this guard with (4) yields both a charged score-regret certificate and full-recall decision-loss preservation. It does not certify the unknown label or remove the architectural approximation to the trained suffix. Appendix G gives the finite argument, state sufficiency, and exhaustive nonrecord-prefix checks.

Corollary 4 (Whole-policy decision-loss preservation)**.** *Assume early and full execution share the same observed-prefix semantics. A controller that conjoins the scalar and return tests, and otherwise falls back to full recall, returns the full-recall label on every valid input. Its expected decision loss therefore equals full recall under any joint input/label distribution for which the expectation exists, although its scalar regret guarantee remains selective to certified stops.*

5 Learning and executing the same contract

The bound restricts output amendments, not their training rule. A persistent feature with arbitrary cube-valued readouts can represent every amendment in $[-\sum_j \beta_j, \sum_j \beta_j]^{10}$ by splitting it proportionally over stages. This capacity statement does not prove trainability. Original hard $\beta = .25$ models clip approximately 99.8–100% of raw update coordinates; even $\beta = .5$ clips roughly 96–98%. Their terminal accuracy is 85.380% versus 86.070% for paired unconstrained fits (-0.690 ± 0.487 percentage points, paired sample standard deviation). The constraint is not free.

The retained five-seed study holds prefixes, splits, feature width and integer deployment budget fixed. Small initialization uses training features; $\beta \tanh(u/\beta)$ avoids flat hard-clamp derivatives, while straight-through estimation changes only the training backward map. All 45 fixed checkpoints and original four-epoch endpoints are preserved. At twelve epochs, smooth small initialization improves on continued hard training by 1.296 points, paired 95% interval $[.865, 1.727]$. Smooth minus small-hard initialization is .240 points $[-.108, .588]$, so this does not isolate smoothness from suitable initialization. Smooth minus unconstrained is $-.142$ $[-.521, .237]$; containing zero does not establish equivalence. The weaker STE and projection results remain in Table 1, not only in the archive.

The inherited image-execution study already makes intermediate decisions: at price .04, ratio $1/3$ and fee 4,096, hard models stop 117.8 and 28.6 inputs at exits 1 and 2 per 10,000; smooth models stop 238.6 and 181.0. Smooth terminal accuracy is 86.712%, whereas its full-recall accuracy is 86.348%; the hard full-recall baseline is 85.446%. Returning the same recall decision is not terminal-head accuracy dominance. Complete depth tables, projection bounds, calibration/derivative diagnostics and actual masked-execution checks remain in the technical appendices and preserved preceding main text.

6 Selecting the policy that carries the guarantee

An inspection schedule chooses where to check, not which executed heads to retain. We consider every $\sigma \subseteq \{0, 1, 2\}$ and compulsory endpoint 3. A prefix costs 172,192 MACs, each correction

Table 2: Training-selected schedules, scalar/return-conjoined modes (S/R). Counts are five-seed means per 10,000 test inputs; changed schedules compare the two guard-specific selectors. $\Delta W = W_R - W_S$ is declared work, not latency. Every guarded return agrees with its own full-recall comparator.

Fee	Price	λ/p_c	Changed	Early S/R	Labels S/R	ΔW
0	0.04	$1/3$	0/5	5611.2/5606.2	1.2/0	8.83
4,096	0.04	$1/3$	0/5	4906.0/4906.0	0.0/0	0.00
4,096	0.08	1	2/5	8847.0/8719.4	22.0/0	269.35
0	0.08	1	0/5	9268.0/9057.4	28.4/0	517.91

17,664, and full execution 225,184. For total suffix price p_c, each transition costs $p_c/3$; a check costs f declared work credits, or $fp_c/52992$ utility units. These charges include rejected checks. Let G be either the scalar gate (S) or its conjunction with the return guard (R). Its work function $W_{\sigma,G}(x)$ changes when G changes, so schedule selection is performed separately:

$$\widehat{\sigma}_G \in \arg\min_{\sigma \subseteq \{0,1,2\}} \frac{1}{n} \sum_{i=1}^{n} W_{\sigma,G}(X_i). \tag{9}$$

Ties prefer fewer checks, then lexicographic order. Every candidate work sum, not only the winner, is frozen. Adding free checks cannot delay certification; adding paid checks can, because an earlier failure consumes tolerance needed at a later prefix. The constructive counterexample is retained in Appendix E.

Theorem 5 (Guard-specific finite-family work bound)**.** *Fix the candidate, guard and m schedules independently of n iid calibration inputs from P. Suppose incremental work lies in $[0, H]$. With probability at least $1 - \delta$, simultaneously for distributions Q with $\mathrm{TV}(P, Q) \le \rho$,*

$$\mathbb{E}_Q W_{\widehat{\sigma}_G,G} - \min_{\sigma} \mathbb{E}_Q W_{\sigma,G} \le 2H\sqrt{\frac{\log(2m/\delta)}{2n}} + 2H\rho. \tag{10}$$

Hoeffding plus a union bound controls every calibration mean, and the selected and comparator expectations each incur at most $H\rho$ under shift. Here $m = 8$, $H = 3(17664 + f)$. The proof is in Appendix B.5. Independence from candidate fitting and a fixed family are essential. This is not a confidence interval for an observed test-oracle gap, and our shared benchmark rows/five seeds are not independent datasets. Joint candidate design requires separate data or a correction for the full considered family.

6.1 Public freeze and guarded-schedule evidence

Five original hard $\beta = .25$ candidates each use their 5,000 reserved controller images to evaluate three fees $\{0, 4096, 16384\}$, four prices $\{.01, .02, .04, .08\}$ and five ratios $\{0, 1/6, 1/3, 1/2, 1\}$, for both guards. Selection reads no labels. The separate calibration object also includes the precision study below: all 870 selected policies and 6,960 candidate sums were committed publicly before the new held-out replay began. Code, model, split, score-table and calibration hashes are in that object. This closes chronology for this extension, not retroactively for the older schedule study, whose combined artifact was uploaded after evaluation. Both use an already public benchmark; neither is presented as untouched confirmation.

The return guard changes 2 of 300 main-grid schedule choices: at fee 4,096, price .08 and ratio 1, seeds 3101 and 3103 switch from dense inspection to $\{0, 2\}$. The other seeds already choose $\{0, 2\}$. In that cell, average work increases by 269.35 credits, paired 95% interval $[177.05, 361.64]$, while 22 scalar-policy label differences per 10,000 disappear. At the reference fee 4,096, price .04, ratio $1/3$, both selectors choose $\{0\}$ and no extra work is needed. With free checks at the same price/ratio, both choose all exits, but protection removes 1.2 label differences at an average cost of 8.832 credits $[2.628, 15.036]$. Equal average accuracy in that cell was not pointwise equality.

Reselection and enforcing the guard are not the same intervention. We also run the guarded predicate using the scalar-selected schedule, then compare with the guarded-selected schedule on the same test trajectories. This holds the guarantee fixed while changing only the inspection plan. All such

Table 3: Precision/temperature sensitivity at price .04, ratio $1/3$, fee 4,096. Scalar and conjoined selectors choose $\{0\}$ and have identical actions in these nine cells. Guarded coverage is relative to each transformed candidate. Last column: full-recall label changes per 10,000 relative to the original $S = 64, \tau = 1$ candidate; this is not a guard violation.

S	τ	Early (%)	Work (10^3)	Changed full labels
16	$1/2$	68.632	192.911	32.2
16	1	49.082	203.270	26.6
16	2	20.922	218.193	38.4
32	$1/2$	68.574	192.941	22.6
32	1	49.076	203.274	12.8
32	2	20.922	218.193	32.2
64	$1/2$	68.550	192.954	18.2
64	1	49.060	203.282	0.0
64	2	20.922	218.193	28.0

paired differences are recorded separately; the principal work differences in Table 2 include both the additional refusals required by return preservation and any change in schedule. The only cell with reselected schedules saves 106.25 credits on average relative to guarded execution under the scalar-selected schedule, with a paired interval $[-94.33, 306.82]$; this is not a resolved population gain. This decomposition makes clear why an unchanged schedule can still have a nonzero guarantee cost, and why zero schedule changes alone would not establish cost-free protection.

All guarded candidate policies have zero full-recall label disagreement; all accepted stops pass the scalar tolerance. Compulsory fallback remains uncertified when accumulated charges or observed regret exceed the budget. All 60 grouped main conditions, including zero-coverage and adverse cases, are in Appendix C. Paired intervals use five seeds, df 4, and are unadjusted descriptive summaries, not 50,000 training replications.

The numerical workflow cross-checks calibration outputs before joining test outcomes. Python object integers and the checked native kernel agree on every registered calibration prefix; the frozen file stores all candidate sums, all ties and all winners. Evaluation reads the freeze by its exact hash and cannot select a different winner from the test records. The full all-schedule arrays also separate two costs: enabling return protection at a fixed schedule, and reselecting a schedule for the protected policy. Their difference is not silently attributed to the guard alone. When no early certificate is affordable, a no-check schedule remains a legitimate full-execution alternative.

6.2 Precision and temperature change the problem

For $S \in \{16, 32, 64\}$ we round the saved prefix and *each* additive increment separately, with ties to even. The integer bound is $S/4$, retaining $\beta = .25$. These are transformed frozen trajectories, not newly trained or recovered high-precision networks. For $\tau \in \{1/2, 1, 2\}$, the table bounds $Me^{-d/(S\tau)}$; records, endpoint supports, return tests and selected schedules are rebuilt at each setting. Prices and tolerance are not rescaled after looking at results. The 27 precision/temperature/fee configurations are evaluated with both guards and all schedules. At fee 4,096, changing S has little effect on reference-temperature coverage (49.060–49.082%), whereas changing τ moves coverage between 20.922% and 68.632%. This is sensitivity to a different score utility, not evidence that lowering temperature improves calibration. Full-recall labels can change because temperatures can reorder confidence records across exits even though a head's argmax is unchanged. The guard preserves each transformed candidate's own full recall, not the original candidate's labels. Appendix B.4 gives rounding and score-transfer bounds; all settings, including unfavorable work and transformed-baseline differences, are retained.

Proposition 6 (Rebuilding a finite score under coarsening)**.** *For the separate prefix/increment rounding above, at exit j, $\|z_j - \widetilde{z}_j\|_\infty \le (j+1)/(2S)$. At fixed temperature τ, real max-softmax scores differ by at most $(j+1)/(4S\tau)$; include each finite table's proved error when comparing finite scores. A fixed-depth regret with common charges changes by at most twice the maximum per-exit score error. No unchanged stopping or label decision follows without an additional margin or exact guard check.*

Table 4: New image-to-return CPU timings (ms): full recall, Python dense guarded, C++ dense guarded, and training-selected C++ guarded schedules. Speedup is the median of paired full/selected ratios, not the ratio of the displayed medians. One fixed model, one thread, nine randomized-order repetitions; batches 1/32/1,000 process 128/992/1,000 fixed test images respectively. Initialization is separate.

CPU	Batch	Full	Py dense	C++ dense	C++ selected	Speedup
x86	1	29.610	76.052	30.676	26.716	1.107
x86	32	19.552	43.867	24.015	21.425	0.915
x86	1000	6.866	10.403	6.358	6.181	1.104
arm	1	84.246	133.111	74.638	66.643	1.267
arm	32	68.283	98.105	73.219	68.800	0.991
arm	1000	20.667	23.641	19.535	19.260	1.073

This follows from rounding error and the $1/(2\tau)$ confidence Lipschitz constant (Appendix B.4). In particular, sensitivity to precision and sensitivity to a new calibrated utility are not interchangeable. The former approximates a trajectory under a fixed score definition; the latter changes which score differences the objective rewards. The tables therefore report both policy disagreements relative to each candidate and between-candidate full-recall changes, rather than combine them into one "safety violation" count.

7 Compiled execution, not inferred speedup

We time the original seed-3101 model, frozen selected schedules, dense schedules and full recall on Intel Xeon 6973P-C and ARM Neoverse-N2 cores. The latter is reported by `lscpu`; the commercial system-on-chip is not identified. Both runs use Torch 2.10.0, NumPy 2.3.5, Python 3.13 and one Torch/BLAS thread. Nine repetitions randomize mode order on fixed official-test subsets. There is no test-time admission or fastest-configuration selection. Timings include image prefix inference, score normalization, gate, masks, executed suffix and return; loading and compilation are separate.

At batch 1,000, the compiled dense *return-conjoined* path is faster than full execution on both CPUs: 6.358 versus 6.866 ms on Xeon and 19.535 versus 20.667 ms on Neoverse. The Python dense guarded path takes 10.403 and 23.641 ms. The selected guarded path takes 6.181 and 19.260 ms, with median paired speedups 1.104 and 1.073. This directly times the selected policy; it does not convert the previous 3,658-credit difference into milliseconds. At batch 32 both selected paths remain slower than full execution, and the Xeon dense path is also slower at batch 1. These adverse regimes are retained.

Gate-only timing on 1,000 fixed calibration histories is reported separately from image inference. The new C++ and Python decisions, executed depths and check counts agree in actual masked execution; guarded labels agree with full recall on each timed workload. Across the full replay, 26.1 million prefix flag comparisons match the independent arbitrary-precision reference, with 7,830 additional Fraction checks and 27 invalid-call tests. Counts repeat the same inputs over registered settings and are not independent examples.

Initialization is not free. The release measures 2.863/3.934 seconds for fresh native compilation on these runners, 1.239/1.559 ms for table proof/rule setup, and 5.349/6.951 ms for model construction/loading. Ahead-of-time deployment need not compile per launch; neither convention is hidden in a steady-state ratio. For measured setup s and positive paired workload saving d, the strict repeated-workload break-even is

$$k_{\mathrm{BE}} = \lfloor s/d \rfloor + 1; \qquad d \le 0 \implies \text{no finite break-even.} \tag{11}$$

For the 1,000-image selected guarded workload, charging the entire measured table/rule setup as incremental cost gives a strict break-even of two repeated workloads, or 2,000 inputs, on both hosts. Including fresh source compilation instead gives 4,436,000 inputs on Xeon and 2,816,000 on Neoverse. These counts use the median paired savings, not differences rounded from the displayed millisecond table. The distinction is consequential: compiling a native module on every short invocation can erase the observed steady-state advantage, whereas distributing an ahead-of-time binary has a different setup profile. Neither scenario is selected retrospectively to suppress the other.

Costs common to full and early execution are retained separately; all raw measured setup components remain available for evaluating a deployment's binary distribution, model-loading requirements and workload horizon.

The full setup-inclusive tables preserve both conventions and no-break-even regimes. They exclude common framework import/data input and are not complete cold-start or worst-case execution-time guarantees. Timing repetitions quantify variability of these fixed workloads, not variation across models or machines.

8 Design implications and limits

Joint selection with independent roles. Budget, precision and fitted calibration may be selected on an independent labeled design split using a predeclared finite family and a simultaneous risk bound. Freeze that choice, then use a separate unlabeled split to select its guarded schedule. With J fixed candidates, bounded-loss design radius $\sqrt{\log(2J/\delta_d)/(2n_d)}$ gives a uniform feasibility test; conditional on design selection, Theorem 5 uses the new schedule split. Proposition 7 proves the combined statement. Prices and acceptable tolerances remain externally declared requirements; fees describe implementation costs rather than a free optimization knob. This is a specified procedure, not a claimed new joint-budget experiment.

Proposition 7 (Separated risk and work selection)**.** *Assume the finite fitted candidate family is independent of the iid design sample, losses lie in* $[0,1]$*, and the schedule sample is iid and independent of both fitting and design selection. With probability at least* $1-\delta_d-\delta_s$*, any selected feasible candidate has full-recall risk at most* $r_{\max}$ *and its empirical conjoined-guard schedule obeys the finite-family work bound at level* δ_s*. Under the pointwise return contract and identical prefix semantics, the resulting guarded policy has the same decision risk as that selected full-recall candidate.*

Proof. A union bound over the J design risks gives the stated simultaneous one-sided risk bound. Any data-dependent choice from the feasible subset therefore satisfies the risk requirement. Conditional on the fitting and design data, the selected candidate and regime are fixed and independent of the schedule sample, so the preceding work theorem applies with m schedules. A union bound over the two failure events proves the assertion. The final equality is the pointwise return-preservation corollary, integrated against the input/label distribution; it requires no additional estimation. □

Scaling and comparison. A sufficient fixed-utility state consists of current logits, best score and earliest return, largest past net reward and paid charges. It avoids storing all scalar records, but not hidden activations or endpoint/bit costs. Large class counts, routing, masking and learning useful bounded readouts may each matter in residual or transformer models; the present small-model results do not identify a universal first bottleneck. Amortizing invariant table validation across trusted calls is a further implementation opportunity; the present timings deliberately retain validation at every inspection. The exact contract is general, while the positive experiments remain one small public benchmark. The saved 128-row Fashion-MNIST verifier diagnostic remains a different cost regime: at hidden radius .1, IBP/split accept 69 and CROWN/optimized CROWN 97; the structural rule accepts 65. Weight-aware bounds have more information, are uncharged in this diagnostic, and are floating relaxations rather than outward-rounded deployment proofs. No verifier-family ranking follows.

9 Conclusion

Coupled output increments give an attainable continuation witness and a sharp certificate for fixed inspection order, finite observed state, specified confidence-resource utility and declared charges. The separate return guard preserves full-recall decisions, not unknown-label correctness. Selecting schedules for that guard, publishing the choices before replay, and compiling the exact history-aware arithmetic close distinct theorem-to-execution gaps. Precision and temperature require new contracts and records, while measured small-batch failures prevent work savings from becoming generic latency claims. The active results and all historical proofs and adverse evidence remain separately indexed and reproducible.

REPRODUCIBILITY STATEMENT

The release preserves all preceding proofs, fitting records, checkpoints, traces, empirical failures and writing sources. New code contains an independent Python integer reference, a checked C++ kernel, separate calibration and test commands, rational table construction, all guard-specific candidate sums, public pre-replay freeze, full per-condition outcomes, actual CPU timing pairs, hardware identifiers and deterministic table generation. The release manifest records source and input hashes; a detached read-only audit identifies the exact published commit. Appendix A is a one-page map of active claims, detailed results, archived material and validation entry points. The benchmark is public historical data, not an untouched confirmatory sample.

AI USE STATEMENT

Generative AI tools assisted with derivations, implementation, literature checking, manuscript editing and artifact validation. The authors remain responsible for the claims, experimental design, results and citations.

REFERENCES

Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019.

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pp. 527–536, 2017.

Elena Botoeva, Panagiotis Kouvaros, Jan Kronqvist, Alessio Lomuscio, and Ruth Misener. Efficient verification of ReLU-based neural networks via dependency analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3291–3299, 2020.

Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi: 10.1016/S0304-3975(01)00144-X.

Rudy Bunel, Alessandro De Palma, Alban Desmaison, Krishnamurthy Dvijotham, Pushmeet Kohli, Philip Torr, and M. Pawan Kumar. Lagrangian decomposition for neural network verification. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 124 of *Proceedings of Machine Learning Research*, pp. 370–379, 2020.

Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, pp. 238–252, 1977. doi: 10.1145/512950.512973.

Yizhak Yisrael Elboher, Avraham Raviv, Amihay Elboher, Zhouxing Shi, Omri Azencot, Hillel Kugler, and Guy Katz. Bridging efficiency and safety: Formal verification of neural networks with early exits. *arXiv preprint arXiv:2512.20755*, 2025.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelović, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations*, 2018.

Metod Jazbec, Patrick Forré, Stephan Mandt, Dan Zhang, and Eric Nalisnick. Early-exit neural networks with nested prediction sets. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pp. 1780–1796, 2024.

Francis A. Longstaff and Eduardo S. Schwartz. Valuing american options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001. doi: 10.1093/rfs/14.1.113.

Qian Qi. Optimal depth of neural networks. *arXiv preprint arXiv:2506.16862*, 2025. URL https://arxiv.org/abs/2506.16862.

Zhouxing Shi, Qirui Jin, Zico Kolter, Suman Jana, Cho-Jui Hsieh, and Huan Zhang. Neural network verification with branch-and-bound for general nonlinearities. *arXiv preprint arXiv:2405.21063*, 2024.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *23rd International Conference on Pattern Recognition*, pp. 2464–2469, 2016. doi: 10.1109/ICPR.2016.7900006.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, pp. 409–424, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *International Conference on Learning Representations*, 2021.

Yuanyuan Yang, Ruimin Zhang, Jamie Morgenstern, and Haifeng Xu. Hold that exit: Near optimal early-exit inference via recall. OpenReview, 2026a. URL https://openreview.net/forum?id=dr1gwWGlhS.

Yuanyuan Yang, Ruimin Zhang, Jamie Morgenstern, and Haifeng Xu. T-tamer: Provably taming trade-offs in ml serving. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=JFY9MZtWTu.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

A ACTIVE-CLAIM AND ARTIFACT MAP

Reading guide. The main text contains the active claims and principal results. Appendices B–C give new proofs, full outcomes and numerical semantics. Appendix D begins the explicit archival section; earlier developments there are retained for audit, not additional headline claims. Historical negative evidence is also summarized in the current main text.

Active claim or question	Manuscript location and authoritative artifact
Sharp charged stopping	Theorem 1; coupled support, full observed past and rejected checks. Historical complete proofs remain in Appendix G.
Full-recall decision loss	Theorem 3 and Corollary 4; exact finite ties in Appendix B.1. This is not unknown-label correctness.
Compiled exact kernel	Proposition 2; Appendix B.2; `r35_kernel.cpp`, `r35_core.py`, `R35_KERNEL_SELFTEST.json`.
Guard-specific selection	Theorem 5; Table 2; all 60 conditions in Table 5; `results/R35_TEST_RESULTS.json`.
Pre-replay chronology	Appendix C.1; `calibration/R35_CALIBRATION_FREEZE.json` and `R35_REPLAY_CHRONOLOGY.json`; all choices and candidate sums.
Precision / temperature	Table 3; Appendix B.4; nine proved tables in `calibration/tables/`; all 54 guard-specific summaries in Table 6.
Actual execution cost	Table 4; all raw pairs, setup and row decisions in `hardware/x86/` and `hardware/arm/`; Table 7 gives setup-inclusive costs.
Joint design without leakage	Proposition 7; separate fitting, labeled design and unlabeled schedule samples. A procedure/theorem, not an executed joint-budget search.
Retained training and verifiers	Table 1; complete prior body and appendices; original R34 inputs, optimization and hardware records are preserved in the companion baseline directory.

Entry points. `r35_study.py calibrate` regenerates calibration-only choices; `r35_study.py replay --freeze-sha <hash>` requires the pinned public object before test access. `r35_release.py build` creates the manuscript, checks and manifest. `r35_release.py audit --output <outside-path>` replays the published state without changing it. `r35_tables.py` regenerates current tables from saved outcomes; it performs no policy selection.

Exact release identity. The build-source commit, input hashes and every output byte are recorded in `R35_MANIFEST.json`. The publication commit itself is identified in `R35_EXACT_HEAD_AUDIT.json`, produced from its detached checkout and included beside this paper. It is external to the hashed PDF to avoid a self-referential commit hash. `R35_PRESERVATION.json` maps unchanged historical source files; the original previous source is `R34_main_archived.tex`.

B COMPILED GUARDED STOPPING: COMPLETE ARGUMENTS

This appendix contains the new results used in the current main text. The sharp coupled-completion theorem, threshold geometry, capacity and projection statements, and return-preservation proofs remain in the inherited technical appendices. The archived previous main text is explicitly separated in Appendix D.

B.1 FINITE ENDPOINT AND RETURN SEMANTICS

Fix integer scale S, stage budgets b_j, and a monotone table E on the validated gap domain, with $E[0] = M$ and $0 \le E[d] \le M$. Let $d(q) = \sum_k E[\max_l q_l - q_k]$ and $g(q) = M/d(q)$. Increasing a maximal coordinate and decreasing its competitors weakly decreases d and hence increases g. If an arbitrary completion has winning class k, its winning class denominator is $\sum_l E[q_k - q_l]$, with nonnegative gaps. Boosting k to its coordinatewise upper bound and lowering all competitors to their lower bounds weakly increases every such gap. The same class remains a possible winner, and confidence cannot decrease. Among endpoint winners, boosting an initially maximal coordinate minimizes the endpoint denominator: exchange a smaller boosted coordinate with an initially maximal one and compare each nonboosted gap. Thus the structural endpoint support is exact. Applying that fixed signed update at each remaining stage attains all future supports simultaneously, including when table plateaus create ties.

For returned-label preservation, let a be the current recalled label and d_* its record denominator. Scores tie according to earliest exit, and class logits tie by smallest class index. A harmful future return must have a label other than a and score strictly larger than M/d_*. If such a completion exists, boost its winning competing class at the maximum remaining radius and lower the others. That class has not lost a pairwise margin, the resulting winning label is not a, and the endpoint denominator is strictly below d_*. Conversely, any competing-class endpoint with these properties is reached by splitting the signed update across the remaining budgets. An earlier intermediate return cannot invalidate this witness: every score in that aligned path is bounded above by its endpoint support, and at least one non-a record strictly exceeds the old record. A competing winner's endpoint therefore supplies a violating completion. Testing all alternate-class endpoints is necessary and sufficient. A denominator equal to d_* is not harmful, because the observed record is earlier than every future exit.

The controller in the new experiments accepts only when the scalar predicate and, for the guarded mode, this return predicate both hold. Every rejected check remains charged. A final fallback returns the full observed recall record but is not automatically a successful scalar certificate. For the conjoined mode, pointwise decision-loss equality to full recall holds even when fallback scalar regret exceeds its tolerance. The comparator is the full-recall output of the *same* precision/temperature candidate, not a different floating-point or calibrated network.

B.2 PROOF OF THE COMPILED INTEGER-WIDTH PROPOSITION

The kernel accepts $K = 10$, $L = 3$, $S \in \{16, 32, 64\}$, $b = S/4$, and $q_0 \in [-8S, 8S]^{10}$. It validates all observed consecutive differences before returning any acceptance. Hence

$$|q_{t,k}| \le 8S + tb, \qquad |q_{t,k} \pm (3-t)b| \le 8S + 3b.$$

Every observed or boosted endpoint gap is at most $16S + 6b = 17.5S < 32S$, inside the table's $32S + 1$ entries. Because a maximal coordinate contributes $E[0] = M$, every denominator satisfies

$$M \le d \le 10M < 2^{52}, \qquad M = 2^{48}.$$

The implementation uses unsigned 64-bit sums for denominators; these sums are exact under the bound. For any rational rule u/v with $|u| \le 2^{40}$ and $0 < v \le 2^{40}$,

$$\frac{M}{d_1} - \frac{M}{d_2} \le \frac{u}{v} \quad \Longleftrightarrow \quad Mv(d_2 - d_1) \le ud_1d_2.$$

The left absolute magnitude is below 2^{140} and the right below 2^{144}. All intermediate products in the stated multiplication order obey those bounds as well. Signed checked 256-bit arithmetic therefore computes the same result as arbitrary-precision integer arithmetic, with ample headroom. The code

never multiplies these expressions in a narrower integer type: operands are explicitly promoted to the multiprecision type first.

Past record minima, earliest record updates, class maximizers, future endpoint construction, and competing-class return tests are the same finite operations as in the independent Python reference. By induction over the observed exits, the records are identical. Exact comparison then gives the same two flag bits at every legal prefix. Applying the same schedule and mask recursion gives identical executed depths, paid check counts and returned labels for every legal input trajectory, not only for the tested trajectories.

This is an arithmetic refinement statement under the ordinary preconditions of the C interface: live arrays with the declared shapes, a faithful compiler and library implementation, and checking the returned status. It is not a proof of memory safety for arbitrary pointers or of an operating system. The Python binding initializes flags to zero, rejects nonzero status and never treats a failed call as a certificate. The native code validates monotone table entries and their range, logit domains, incremental budgets, and rational operand ranges. It resets all flags on exceptions. Invalid buffer metadata is rejected; malformed rows are checked *before* the first acceptance so they cannot leave a partially accepted output batch. The complete invalid-input tests and the source hashes accompany the release.

The native function still scans the observed history (at most three exits here), validates the table each call, constructs future endpoints, and may check nine competing classes. It is not the one-comparison high-confidence first-exit shortcut. Compilation removes Python object-integer dispatch from these operations; it does not make normalization, validation, memory copies, history reconstruction, masking, or hidden-state inference free.

B.3 Independent rational table construction

For $T \in \{1/2, 1, 2\}$, define $x = (ST)^{-1} \le 1/8$. Alternating Taylor partial sums satisfy

$$l_{17} = \sum_{k=0}^{17} \frac{(-x)^k}{k!} \le e^{-x} \le u_{16} = \sum_{k=0}^{16} \frac{(-x)^k}{k!}.$$

All sums are exact rationals. Set $a = \lfloor M l_{17} \rfloor$, $b = \lceil M u_{16} \rceil$, $L_0 = U_0 = M$, and recursively

$$L_{d+1} = \lfloor L_d a / M \rfloor, \qquad U_{d+1} = \lceil U_d b / M \rceil.$$

Induction yields $L_d \le M e^{-d/(ST)} \le U_d$. The lower sequence is nonnegative and nonincreasing. With $w = \max_d (U_d - L_d)$, the sum of ten lookup errors is at most $10w$. Both exact and finite denominators are at least M, so their reciprocal score difference is at most $10w/M$. The frozen table metadata records every a, b, w and bound. At $S = 64, T = 1$, the lower table is byte-identical as an integer array to the original candidate's table; consequently the main-grid finite score is unchanged. Other tables are rebuilt and checked, not obtained by temperature-scaling an already computed confidence score.

B.4 Precision transformation and score transfer

Let $z_j = q_j^{(64)}/64$ be a saved integer trajectory. The coarser candidate rounds z_0 and each additive correction separately to the nearest multiple of $1/S$, with ties to the even integer; its trajectory is

$$\widetilde{z}_0 = S^{-1} \operatorname{round}(S z_0), \qquad \widetilde{z}_j = \widetilde{z}_0 + S^{-1} \sum_{l=1}^{j} \operatorname{round}(S(z_l - z_{l-1})).$$

Since $S/4$ is integral, rounding a correction in $[-1/4, 1/4]$ preserves the same deployment budget. Prefix rounding preserves the range $[-8, 8]$. The triangle inequality gives

$$\|z_j - \widetilde{z}_j\|_\infty \le \frac{j+1}{2S}.$$

At $S = 64$ the transformation is exactly the identity, so its actual error is zero. It is not a reconstruction of unrounded training logits, and the coarser trajectories are not newly trained networks.

For $g_T(z) = \max_k \mathrm{softmax}(z/T)_k$, the ℓ_1 norm of a class-probability gradient is $2p_k(1-p_k)/T \le 1/(2T)$. Integrating along a segment, and using that the maximum of functions with a common Lipschitz constant is equally Lipschitz, gives

$$|g_T(z_j) - g_T(\widetilde{z}_j)| \le \frac{j+1}{4ST}.$$

Include the two finite-table transfer errors when comparing different finite scores. For any fixed executed depth and common charges, both the largest comparator reward and the candidate reward change by at most the maximum per-exit score error; their regret changes by at most twice that error. This does not imply that a threshold-based stopping policy takes the same actions: a boundary crossing can change the schedule outcome discontinuously. Nor does a score-error bound preserve a returned label without an appropriate margin or return test. We therefore rebuild records, endpoint supports, return guards and training-selected schedules separately for every (S, T). Temperature changes the score utility and changes the effective radius to r/T; the original certificate cannot simply be attached to a rescaled reported score.

B.5 Guard-specific schedule selection

Fix a fitted candidate, precision, temperature, declared prices, tolerance, fee and guard mode before obtaining schedule-calibration observations. A schedule determines the inspected exits; every executed exit remains in recall. For either the scalar or conjoined guard, incremental work obeys $0 \le W_\sigma(X) \le H = 3(17664 + f)$ in the present architecture. For m fixed schedules and n iid inputs independent of candidate fitting, Hoeffding's inequality and a union bound imply simultaneously

$$|\widehat{\mathbb{E}}_P W_\sigma - \mathbb{E}_P W_\sigma| \le H\sqrt{\frac{\log(2m/\delta)}{2n}} =: e$$

with probability at least $1-\delta$. Comparing the empirical minimizer $\widehat{\sigma}$ with a population minimizer σ^* gives $\mathbb{E}_P W_{\widehat{\sigma}} - \mathbb{E}_P W_{\sigma^*} \le 2e$. If $\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)| \le \rho$, the layer-cake representation of a $[0, H]$ function bounds each expectation shift by $H\rho$. Applying this to the selected schedule and the Q-optimal comparator yields

$$\mathbb{E}_Q W_{\widehat{\sigma}} - \min_\sigma \mathbb{E}_Q W_\sigma \le 2e + 2H\rho.$$

The proof applies to the conjoined guard without modification, because the work function is bounded and measurable. It must, however, be applied to *that guard's* work function and empirical minimizer: the scalar-only selector need not minimize guarded work. The fixed-family assumptions do not become weaker because the return guarantee is pointwise.

The theorem does not certify the empirical test-oracle gap and does not turn five seeds sharing benchmark rows into independent datasets. It also does not cover adaptive enlargement of the schedule family on the same split. Repeated configurations in the current public-data reanalysis are reported as registered comparisons, not independent population confirmations.

B.6 Joint design without reusing the schedule split

A practical design separates requirements from quantities to be optimized. Prices represent an externally declared trade-off; tolerance is an externally acceptable shortfall; an inspection tariff represents a measured or otherwise declared implementation cost. Optimizing these numbers solely to improve reported savings would change the question rather than solve it. An application may predeclare several admissible regimes, but must report which regime its guarantee describes.

Here is a precise procedure for choosing a predictor within such a regime. Fit a finite family of J candidates (including every budget, precision, temperature and checkpoint that will be considered) on a fitting sample. Using an independent labeled design sample of size n_d, estimate each full-recall candidate's bounded decision risk $\widehat{R}_\theta$. For a specified failure level δ_d, put

$$e_d = \sqrt{\log(2J/\delta_d)/(2n_d)}.$$

Keep only candidates satisfying $\widehat{R}_\theta + e_d \le r_{\max}$, and choose among them by a predeclared design objective or cost estimate. If none qualify, report that fact rather than certifying a different risk

threshold after looking at the data. Freeze the selected candidate and its operating regime. On a *third*, independent, unlabeled schedule sample, select among the m fixed schedules using conjoined-guard work. Publish the complete selector before touching the evaluation sample.

Proposition 7 in the main text gives the separated risk/work statement and its proof.

This is a design procedure and theorem, not a claim that the present study ran a new joint budget/price optimization. Reusing a split is possible only with a valid bound for the entire considered family, such as a union bound over all Jm candidate/schedule pairs for bounded work, plus the necessary risk correction. The smaller $m = 8$ correction alone cannot justify joint selection. A shift bound for work does not control shifted task risk without additional risk assumptions; pointwise equality to full recall still holds only while the enforced inputwise contract and prefix semantics hold.

C COMPLETE NEW EVIDENCE AND ITS INTERPRETATION

The new analyses compare the same five frozen original hard bounded candidates. They do not add newly trained networks or overwrite earlier optimization runs. Each precision/temperature candidate has its own finite score and full-recall baseline. Scalar and scalar-plus-return policies are separately calibrated; all executed heads, not merely inspected heads, remain available to recall. Every table names its guard mode. All accepted scalar checks are valid, and all guarded candidate-policy returns agree with their own full-recall labels. The empty schedule returns full recall without inspections; this is not an accepted early certificate. Terminal fallback can have scalar regret above the tolerance and is reported separately in the per-condition records.

C.1 CHRONOLOGY, SOURCES AND SCOPE

The protocol was committed before the new calibration command. A failed first calibration workflow lacked the C++ header dependency and stopped before calibration; its failure is retained in the repository history. Installing the header library changed the workflow environment, not the registered scientific code. The successful calibration run published all 870 policies, all 6,960 candidate work sums, input/model/split hashes and nine proved tables in a separate commit before the new replay workflow was created. The replay and hardware jobs require that commit to be an ancestor, require byte equality to its freeze, and verify its SHA-256 before any evaluation command.

The immutable freeze SHA-256 is

```
435047544fce8398562b0ec74d29ea4ad33
33d59fafcbc547c7da41e43a93358
```

The freeze commit is `f7b8d48927d5474dfd975757cf8e87d7875df309`; the new replay source commit is `384993b2b7e254fb9891a2af29fdde7e2e9eb162`. The public commit timestamps place the freeze at 09:24:42 UTC and the replay workflow source at 09:29:14 UTC on September 26, 2026. Full run identities and archive hashes accompany the release. This proves the ordering of these public artifacts; it neither erases earlier knowledge of Fashion-MNIST nor backdates the previous schedule study's combined post-replay artifact. The latter chronology caveat remains in the preserved R34 appendix.

C++/Python flag agreement covers 13,050,000 calibration and 26,100,000 test prefix/configuration comparisons. Test replay adds 7,830 separate Fraction checks; deterministic synthetic tests add 186,624 flag comparisons, 5,832 Fraction checks and 27 invalid calls. The 6,960 candidate-policy settings produce 69,600,000 stored depth and check-count entries each. These are repeated configurations on five sets of 10,000 benchmark inputs, not millions of independent observations. The file records contain every negative outcome, not only the examples in the main table.

C.2 ALL MAIN GUARD-SPECIFIC SCHEDULES AND DIFFERENCES

Table 5: All 60 main configurations. S uses the scalar gate; R uses scalar plus return preservation, each independently selected on calibration data. Guarded label changes are zero in every row. Counts per 10,000; work differences have unadjusted paired five-seed t intervals (df 4).

Fee	Price	Ratio	Changes	Early S	Early R	Labels S	ΔW	95% interval
0	0.01	0	0	2841.8	2841.8	0.0	0.00	[0.00,0.00]
0	0.01	1/6	0	3309.2	3309.2	0.0	0.00	[0.00,0.00]
0	0.01	1/3	0	3932.0	3931.2	0.0	1.41	[-0.42,3.25]
0	0.01	1/2	0	4177.2	4175.6	0.0	2.83	[-0.84,6.50]
0	0.01	1	0	5764.2	5746.6	1.4	31.09	[17.43,44.75]
0	0.02	0	0	3571.2	3571.2	0.0	0.00	[0.00,0.00]
0	0.02	1/6	0	4092.8	4092.8	0.0	0.00	[0.00,0.00]
0	0.02	1/3	0	4709.2	4707.4	0.0	3.18	[-0.74,7.10]
0	0.02	1/2	0	4990.4	4987.0	0.0	6.01	[-2.71,14.72]
0	0.02	1	0	6623.6	6580.8	3.8	75.60	[51.71,99.50]
0	0.04	0	0	4387.8	4387.8	0.0	0.00	[0.00,0.00]

Fee	Price	Ratio	Changes	Early S	Early R	Labels S	ΔW	95% interval
0	0.04	1/6	0	4955.8	4955.8	0.0	0.00	[0.00,0.00]
0	0.04	1/3	0	5611.2	5606.2	1.2	8.83	[2.63,15.04]
0	0.04	1/2	0	5938.0	5930.0	1.6	14.13	[2.12,26.14]
0	0.04	1	0	7672.6	7582.0	9.2	160.04	[110.39,209.68]
0	0.08	0	0	5291.2	5291.2	0.0	0.00	[0.00,0.00]
0	0.08	1/6	0	5995.6	5995.6	0.0	0.00	[0.00,0.00]
0	0.08	1/3	0	6801.4	6788.6	2.8	22.61	[2.91,42.31]
0	0.08	1/2	0	7269.6	7249.4	4.2	35.68	[5.47,65.89]
0	0.08	1	0	9268.0	9057.4	28.4	517.91	[338.65,697.17]
4,096	0.01	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
4,096	0.01	1/6	0	2981.6	2981.6	0.0	0.00	[0.00,0.00]
4,096	0.01	1/3	0	3294.8	3294.8	0.0	0.00	[0.00,0.00]
4,096	0.01	1/2	0	3463.6	3463.6	0.0	0.00	[0.00,0.00]
4,096	0.01	1	0	5329.2	5315.6	1.0	24.02	[4.25,43.80]
4,096	0.02	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
4,096	0.02	1/6	0	3716.8	3716.8	0.0	0.00	[0.00,0.00]
4,096	0.02	1/3	0	4053.2	4053.2	0.0	0.00	[0.00,0.00]
4,096	0.02	1/2	0	4252.0	4252.0	0.0	0.00	[0.00,0.00]
4,096	0.02	1	0	6449.2	6411.8	3.0	66.06	[42.98,89.15]
4,096	0.04	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
4,096	0.04	1/6	0	4526.8	4526.8	0.0	0.00	[0.00,0.00]
4,096	0.04	1/3	0	4906.0	4906.0	0.0	0.00	[0.00,0.00]
4,096	0.04	1/2	0	5139.6	5139.6	0.0	0.00	[0.00,0.00]
4,096	0.04	1	0	7452.8	7376.0	7.2	135.66	[92.87,178.45]
4,096	0.08	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
4,096	0.08	1/6	0	5488.2	5488.2	0.0	0.00	[0.00,0.00]
4,096	0.08	1/3	0	5930.4	5930.4	0.0	0.00	[0.00,0.00]
4,096	0.08	1/2	0	6227.2	6227.2	0.0	0.00	[0.00,0.00]
4,096	0.08	1	2	8847.0	8719.4	22.0	269.35	[177.05,361.64]
16,384	0.01	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.01	1/6	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.01	1/3	0	610.8	610.8	0.0	0.00	[0.00,0.00]
16,384	0.01	1/2	0	2606.0	2606.0	0.0	0.00	[0.00,0.00]
16,384	0.01	1	0	3713.2	3713.2	0.0	0.00	[0.00,0.00]
16,384	0.02	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.02	1/6	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.02	1/3	0	3525.8	3525.8	0.0	0.00	[0.00,0.00]
16,384	0.02	1/2	0	3952.6	3952.6	0.0	0.00	[0.00,0.00]
16,384	0.02	1	0	4514.2	4514.2	0.0	0.00	[0.00,0.00]
16,384	0.04	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.04	1/6	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.04	1/3	0	4320.0	4320.0	0.0	0.00	[0.00,0.00]
16,384	0.04	1/2	0	4807.8	4807.8	0.0	0.00	[0.00,0.00]
16,384	0.04	1	0	5455.0	5455.0	0.0	0.00	[0.00,0.00]
16,384	0.08	0	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.08	1/6	0	0.0	0.0	0.0	0.00	[0.00,0.00]
16,384	0.08	1/3	0	5239.8	5239.8	0.0	0.00	[0.00,0.00]
16,384	0.08	1/2	0	5812.0	5812.0	0.0	0.00	[0.00,0.00]
16,384	0.08	1	0	6625.8	6625.8	0.0	0.00	[0.00,0.00]

A selected schedule can stay unchanged while guarded work increases: the return predicate may reject additional prefixes inside that schedule. Conversely, reselecting the schedule can recover some of this additional cost. The machine-readable paper summary separately compares guarded execution under the scalar-selected schedule with guarded execution under the guarded-selected schedule. Thus the main table's work difference is not confused with a pure schedule-reoptimization gain. No test best schedule replaces a calibration winner, and all paired intervals use five seed differences, four degrees of freedom and no multiplicity adjustment.

C.3 All precision and temperature settings

Table 6: All precision/temperature conditions, scalar (S) and conjoined-return (R) modes. Five selected schedules are listed in seed order, with a dash for no checks. Every input retains all executed heads for recall. No temperature or precision is selected using these test outcomes.

S	τ	Fee	Guard	Schedules	Early	Work	Labels changed
16	1/2	0	S	012/012/012/012/012	7471.4	186883.50	0.4
16	1/2	0	R	012/012/012/012/012	7467.4	186890.57	0.0
16	1/2	4,096	S	0/0/0/0/0	6863.2	192910.53	0.0
16	1/2	4,096	R	0/0/0/0/0	6863.2	192910.53	0.0
16	1/2	16,384	S	0/0/0/0/0	6414.2	207577.87	0.0
16	1/2	16,384	R	0/0/0/0/0	6414.2	207577.87	0.0
16	1	0	S	012/012/012/012/012	5612.4	197050.55	0.8
16	1	0	R	012/012/012/012/012	5607.4	197059.38	0.0
16	1	4,096	S	0/0/0/0/0	4908.2	203270.47	0.0
16	1	4,096	R	0/0/0/0/0	4908.2	203270.47	0.0
16	1	16,384	S	0/0/0/0/0	4319.6	218677.58	0.0
16	1	16,384	R	0/0/0/0/0	4319.6	218677.58	0.0
16	2	0	S	012/012/012/012/012	3392.8	210646.17	1.6
16	2	0	R	012/012/012/012/012	3379.0	210670.55	0.0
16	2	4,096	S	0/0/0/0/0	2092.2	218193.01	0.0
16	2	4,096	R	0/0/0/0/0	2092.2	218193.01	0.0
16	2	16,384	S	-/-/-/-/-	0.0	225184.00	0.0
16	2	16,384	R	-/-/-/-/-	0.0	225184.00	0.0
32	1/2	0	S	012/012/012/012/012	7463.2	186917.06	0.4
32	1/2	0	R	012/012/012/012/012	7460.0	186922.72	0.0
32	1/2	4,096	S	0/0/0/0/0	6857.4	192941.27	0.0
32	1/2	4,096	R	0/0/0/0/0	6857.4	192941.27	0.0
32	1/2	16,384	S	0/0/0/0/0	6422.8	207532.30	0.0
32	1/2	16,384	R	0/0/0/0/0	6422.8	207532.30	0.0
32	1	0	S	012/012/012/012/012	5613.0	197044.54	0.8
32	1	0	R	012/012/012/012/012	5608.8	197051.96	0.0
32	1	4,096	S	0/0/0/0/0	4907.6	203273.65	0.0
32	1	4,096	R	0/0/0/0/0	4907.6	203273.65	0.0
32	1	16,384	S	0/0/0/0/0	4320.4	218673.34	0.0
32	1	16,384	R	0/0/0/0/0	4320.4	218673.34	0.0
32	2	0	S	012/012/012/012/012	3392.2	210645.82	1.4
32	2	0	R	012/012/012/012/012	3378.8	210669.49	0.0
32	2	4,096	S	0/0/0/0/0	2092.2	218193.01	0.0
32	2	4,096	R	0/0/0/0/0	2092.2	218193.01	0.0
32	2	16,384	S	-/-/-/-/-	0.0	225184.00	0.0
32	2	16,384	R	-/-/-/-/-	0.0	225184.00	0.0
64	1/2	0	S	012/012/012/012/012	7459.0	186949.57	0.6
64	1/2	0	R	012/012/012/012/012	7455.8	186955.22	0.0
64	1/2	4,096	S	0/0/0/0/0	6855.0	192953.98	0.0
64	1/2	4,096	R	0/0/0/0/0	6855.0	192953.98	0.0
64	1/2	16,384	S	0/0/0/0/0	6428.8	207500.50	0.0
64	1/2	16,384	R	0/0/0/0/0	6428.8	207500.50	0.0
64	1	0	S	012/012/012/012/012	5611.2	197048.78	1.2
64	1	0	R	012/012/012/012/012	5606.2	197057.61	0.0
64	1	4,096	S	0/0/0/0/0	4906.0	203282.12	0.0
64	1	4,096	R	0/0/0/0/0	4906.0	203282.12	0.0
64	1	16,384	S	0/0/0/0/0	4320.0	218675.46	0.0
64	1	16,384	R	0/0/0/0/0	4320.0	218675.46	0.0
64	2	0	S	012/012/012/012/012	3392.6	210644.76	1.6
64	2	0	R	012/012/012/012/012	3379.0	210668.78	0.0
64	2	4,096	S	0/0/0/0/0	2092.2	218193.01	0.0
64	2	4,096	R	0/0/0/0/0	2092.2	218193.01	0.0
64	2	16,384	S	-/-/-/-/-	0.0	225184.00	0.0
64	2	16,384	R	-/-/-/-/-	0.0	225184.00	0.0

Every transformed full-recall label is compared with the original $S = 64, \tau = 1$ full-recall label in `precision_baselines`. These between-candidate changes are separate from within-candidate policy disagreements. In particular, positive temperature scaling does not change an individual head's logit argmax, but it can reorder which head has maximal confidence and is returned by recall. Coarsening logits can change both argmax ties and confidence records. No accuracy or coverage increase after changing temperature is called improved calibration: no temperature was fitted to a proper scoring objective in this study. All nine tables use independent exact rational construction and outward entry bounds, as proved in Appendix B.3.

C.4 Actual CPU execution and all setup conventions

The hardware source and mode order rule were committed before timing. It uses one model, seed 3101, and the reference fee/price/ratio. Both guard-specific selected schedules are $\{0\}$ for that setting. We time seven modes: full recall, compiled selected scalar, compiled selected guarded, compiled dense scalar, compiled dense guarded, Python dense scalar and Python dense guarded. The main table emphasizes the guarded paths; every raw pair and scalar-path measurement remains in the JSON. Each workload is warmed once per mode and then timed in nine independently shuffled mode orders. No empirical admission filter or fastest-configuration choice is applied. Rowwise depths, calls and returned labels accompany the timing records. These are scoped measurements, not tail-latency, energy, accelerator or production-serving evidence.

The new ARM host exposes `Neoverse-N2` in `lscpu --json`; `/proc/cpuinfo` lacks a commercial model-name line. We preserve both raw outputs and identify the core microarchitecture without guessing a system vendor or commercial chip. The x86 host exposes Intel Xeon 6973P-C. Both expose four logical CPUs; Torch and BLAS are set to one thread. The exact compiler, Torch, NumPy, Python, kernel binary, model and image hashes are recorded. The previous AMD and unidentified-ARM measurements remain archival evidence, not relabeled as the new hosts.

Table 7: Setup-inclusive repeated-workload accounting for the conjoined-return paths. Positive paired-median savings d require strictly more than setup/d repeated workloads. Table/rule setup is conservatively charged in full even though part is common normalization work. Fresh compilation is shown separately from ahead-of-time use. Common imports, model loading and data input are not included: these are not complete cold-start estimates.

CPU	Batch	Schedule	Setup charge	Setup ms	Saved ms	Break-even inputs
x86	1	dense	table/rules	1.239	-1.033	none
x86	1	dense	+ compilation	2864.544	-1.033	none
x86	1	selected	table/rules	1.239	2.886	128
x86	1	selected	+ compilation	2864.544	2.886	127,104
x86	32	dense	table/rules	1.239	-4.429	none
x86	32	dense	+ compilation	2864.544	-4.429	none
x86	32	selected	table/rules	1.239	-1.788	none
x86	32	selected	+ compilation	2864.544	-1.788	none
x86	1000	dense	table/rules	1.239	0.441	3,000
x86	1000	dense	+ compilation	2864.544	0.441	6,494,000
x86	1000	selected	table/rules	1.239	0.646	2,000
x86	1000	selected	+ compilation	2864.544	0.646	4,436,000
arm	1	dense	table/rules	1.559	9.587	128
arm	1	dense	+ compilation	3935.927	9.587	52,608
arm	1	selected	table/rules	1.559	17.774	128
arm	1	selected	+ compilation	3935.927	17.774	28,416
arm	32	dense	table/rules	1.559	-4.748	none
arm	32	dense	+ compilation	3935.927	-4.748	none
arm	32	selected	table/rules	1.559	-0.635	none
arm	32	selected	+ compilation	3935.927	-0.635	none
arm	1000	dense	table/rules	1.559	1.114	2,000
arm	1000	dense	+ compilation	3935.927	1.114	3,533,000
arm	1000	selected	table/rules	1.559	1.398	2,000
arm	1000	selected	+ compilation	3935.927	1.398	2,816,000

For k repetitions of a fixed measured workload, compare $T_{\text{full}}(k) = kt_f$ with $T_{\text{gate}}(k) = s+kt_g$ using the stated setup convention. The table uses the median paired saving $d = t_f - t_g$, not an assumption that the future deployment horizon is already large. Model construction/load was measured once per process and is common to the policy comparisons; framework import and image loading were not included. Fresh compilation, table/rule setup and steady-state execution are explicitly separated. Ahead-of-time compiled deployment and compiling on every launch are different scenarios. The derived break-even counts extrapolate a fixed workload's medians; they are not guaranteed future demand or latency bounds.

C.5 Inherited projection statement retained as an active reference

For independent hidden dynamics and unconstrained updates u_s, write $e_s = \|u_s - \text{clip}(u_s, -\beta_s, \beta_s)\|_\infty$. The retained trajectory bound is

$$\|z_j - \widetilde{z}_j\|_\infty \le E_j := \sum_{s=1}^{j} e_s + (j+1)/(2S). \tag{12}$$

A head margin above $2E_j$ preserves that head's label; confidence changes by at most $E_j/2$ and cross-entropy by $2E_j$ at unit temperature. These are headwise bounds, not a substitute for full-recall preservation. The complete historical proof and empirical projection results follow in the archive.

D ARCHIVED R34 MAIN-TEXT DEVELOPMENT

This is the complete preceding main-text development, preserved for continuity, not an additional set of active claims. Its original source is retained byte for byte as `R34_main_archived.tex`. Only section levels, the abstract wrapper, and internal labels in this readable copy are namespaced to prevent duplicate references. Current claims and results are indexed in Appendix A.

PRESERVED ABSTRACT

When can a neural network stop without executing the suffix that justifies the stop? For a fixed ordered inspection schedule, the full finite observed record, a confidence-minus-resource objective, and coupled bounded logit increments, we give an information-optimal certificate. One aligned completion attains every future confidence support. An exact integer implementation handles nonrecord prefixes and every unsuccessful inspection charge; a separate sharp guard preserves the full-recall returned label, not unknown label correctness. Five-seed Fashion-MNIST experiments connect the theorem to actual four-exit execution and repair severe hard-clamp saturation under the same deployment bound. We further complete a registered comparison of eight inspection schedules, three fees, four prices, and five tolerances. Schedules selected on reserved training inputs, without labels or test selection, recover 3,658 mean work credits relative to inspecting every exit at the reference fee and price. Free inspection instead favors the dense schedule. A finite-family bound separates calibration error from distribution shift. Rowwise verifier audits and x86/ARM measurements distinguish stronger bounds, resource accounting, and observed latency. All 2,400 schedule conditions, training freezes, proofs, and historical adverse results are retained.

D.1 INTRODUCTION

Adaptive depth needs both a prediction and a reason not to execute another layer. Learned halting and routing estimate useful task–compute trade-offs (Graves, 2016; Teerapittayanon et al., 2016; Huang et al., 2018; Wang et al., 2018). A certificate asks a different question: can any suffix consistent with the available information improve the specified objective enough to matter? Recall makes resource accounting essential. Executing three corrections and returning the prefix label still pays for three corrections.

We impose an architectural contract: each stage adds a coordinatewise bounded correction to persistent logits. Raising a currently maximal logit and lowering all competitors maximizes confidence at every future depth simultaneously. The resulting coupled witness proves both soundness and information-relative maximality. With the full finite observed record, fixed inspection order, confidence-resource utility, and declared charges, the first successful inspection is the minimum certifiable executed depth. This establishes an operational notion of optimal depth without conflating it with architecture search, task-optimal prediction, or unrestricted adaptive scheduling.

This distinction leaves three constructive problems. A current-confidence threshold loses recall information at nonrecord prefixes; we implement the history-aware certificate at every nonterminal exit. Almost fully saturated hard clipping impedes training; initialization and smooth bounded readouts repair much of that loss without weakening the deployed contract. Finally, extra inspections can cost more than the suffix work they save. We complete training-only selection among eight fixed schedules and prove a finite-family calibration bound. Thus depth-minimal certification *within* a schedule and expected-work selection *between* schedules have separate objectives, assumptions, and evidence.

The primary contribution is a sharp contract-to-controller construction, including finite numerical semantics and full-recall label preservation. Five-seed experiments, actual masked execution, charged schedule selection, matched verification, and two CPU instruction sets test that construction. Score-regret validity, return-label preservation, empirical classification accuracy, and measured latency remain distinct throughout. All earlier proofs, longer explanations, and unfavorable experimental findings are preserved in the appendices.

D.2 Related work

The depth-stopping formulation of Qi (2025) motivates the with-recall problem. Adaptive prediction and conditional computation (Bolukbasi et al., 2017; Graves, 2016; Huang et al., 2018; Wang et al., 2018) learn task–compute trade-offs. Regression and neural optimal stopping (Longstaff & Schwartz, 2001; Becker et al., 2019) estimate continuation values; recall-aware threshold models study loss trajectories (Yang et al., 2026b;a). Our guarantee instead quantifies all completions of an architectural increment contract, conditional on an observed prefix. Statistical early-exit prediction sets and compute-risk control (Jazbec et al., 2024; Wang et al., 2026) provide different sampling-based guarantees. A score-regret certificate is not a replacement for their population-risk statements.

Elboher et al. (2025) directly verify early-exit networks, including conditional exit guards and local input robustness. This is a close antecedent. A general verifier can encode our recall and regret predicate; our contribution is the sharp closed form obtained from the output contract, not a new generic solver or an input-robustness theorem. Interval bounds and abstract interpretation (Cousot & Cousot, 1977; Gowal et al., 2018), CROWN and optimized linear relaxations (Zhang et al., 2018; Xu et al., 2021), and split-domain verification (Wang et al., 2021) exploit more of the trained network. We compare the authors' auto_LiRPA implementations on the same Fashion-MNIST suffix and conditional domains. They have access to weights and hidden-state boxes that the structural rule deliberately ignores. The earlier MNIST comparison and historical grid, proof-library, and shortest-path developments remain in the appendices, without being relabeled as new experiments.

D.3 Sharp stopping with recall and inspection charges

A network has exits $0, \dots, L$, scores $g_j \in [0,1]$, and logits $z_j \in \mathbb{R}^K$. Write $C_0 = 0$, $C_t = \sum_{s=1}^{t} c_s$, $c_s \geq 0$, $B_t = \max_{j \leq t} g_j$, and $R_t = B_t - C_t$. The returned exit is the shallowest maximizer of the observed score; the executed depth is t. Let $Z_t \geq 0$ be the cumulative charge for all inspections through t, including the current check and earlier rejections. We compare the charged candidate $R_t - Z_t$ with the uncharged network-depth comparator $\max_{0 \leq j \leq L} R_j$. This demanding comparator does not silently give competing depths the controller's overhead. Its regret is

$$\begin{aligned} \Gamma_t^Z &= \max_{0 \leq j \leq L} R_j - R_t + Z_t = \max\{P_t, S_t\} + Z_t, \qquad P_t = \max_{j \leq t} R_j - R_t, \\ S_t &= \Big[\max_{j > t} (g_j - B_t - D_j) \Big]_+ . \end{aligned} \tag{13}$$

where $D_j = C_j - C_t$. Nonnegative transition costs justify the reduction: a future record is best charged when it appears, and an old record cannot create positive continuation improvement. The past term P_t is already observed; no estimate of the suffix can repair it.

The architectural contract is

$$z_j = z_{j-1} + \eta_j, \qquad \|\eta_j\|_\infty \leq \beta_j, \qquad r_j = \sum_{s=t+1}^{j} \beta_s. \tag{14}$$

Let $\mathcal{A}_t$ contain every completion satisfying this coupled contract, not just those reachable through particular trained weights. For max-softmax confidence, put $p = \max_k \operatorname{softmax}(z_t)_k$ and

$$Q_j(p) = \frac{e^{2r_j} p}{1 + (e^{2r_j} - 1)p}, \qquad U_t = \Big[\max_{j > t} \{ Q_j(p) - B_t - D_j \} \Big]_+ . \tag{15}$$

An empty future maximum contributes zero.

Theorem 8 (Sharp charged certificate)**.** *Under* (14)*,* $\sup_{\mathcal{A}_t} \Gamma_t^Z = \max\{P_t, U_t\} + Z_t$*. Thus stopping is sound at tolerance* λ *exactly when*

$$\max\{P_t, U_t\} + Z_t \leq \lambda. \tag{16}$$

No rule using only the same observed scores and logits, increment budgets, charges, and comparator can soundly accept a larger set of prefixes. On a fixed inspection schedule, the first successful test has the smallest certifiable executed depth along that observed trajectory.

Proof. For a fixed class, its softmax probability increases with its own logit and decreases with every competitor. Raising it by r_j and lowering all others by r_j gives the largest class probability

in the endpoint cube. Choosing a currently maximal class maximizes that expression over classes and yields $Q_j(p)$. Apply the same signed increment of magnitude β_s at every stage s. This one admissible sequence attains all Q_j simultaneously, so the maximum of the endpoint improvements in (15) is attained by a coupled completion. The past and inspection terms are fixed. If (16) fails, this completion is a witness against any rule with the stated information. Earliest acceptance follows on the fixed schedule; it does not compare different inspection schedules. □

This distinction between a certificate and a fallback is important. An input that fails every inspection executes the final stage and returns its best observed prediction. That fallback is *not* automatically certified: $P_L + Z_L$ can exceed λ even though there is no remaining suffix. Our implementations and tables distinguish certified early stops from compulsory final execution. Charging an inspection before acceptance also means that $Z_t > \lambda$ precludes any certificate, irrespective of confidence.

D.3.1 When the rule becomes a confidence threshold

At a record prefix $B_t = p$, each future constraint has the form $h_s(p) \le a$, where $s = e^{2r_j}$, $a = \lambda - Z_t + D_j$, and

$$h_s(p) = \frac{(s-1)p(1-p)}{1+(s-1)p}, \quad \max_{p\in[0,1]} h_s(p) = \frac{\sqrt{s}-1}{\sqrt{s}+1}, \quad p_\pm = \frac{1-a \pm \sqrt{(1-a)^2 - 4a/(s-1)}}{2}. \tag{17}$$

For a nonvacuous positive constraint, acceptance lies below p_- or above p_+. Intersect these sets over future exits and enforce the observed-past constraint. On $p \ge 1/2$, the future part is a single upper-threshold test; its cutoff can be compiled offline. Retaining only this branch is conservative when a lower branch exists. At nonrecord prefixes, replacing B_t by current p or discarding P_t changes the problem. Our multi-exit controller therefore retains the score record and past net rewards rather than applying a first-exit cutoff at every stage.

D.4 Finite implementation and task-level return preservation

Deployment uses $S = 64$ logit units, prefix integers $q_0 \in [-512, 512]^K$, and additive updates bounded by 16 units when $\beta = 0.25$. Let $M = 2^{48}$ and $E^-[d]$ be the committed monotone integer lower approximation of $Me^{-d/S}$, with $E^-[0] = M$. Define

$$d(q) = \sum_k E^-[\max_\ell q_\ell - q_k], \qquad \widehat{g}(q) = M/d(q). \tag{18}$$

Rational Taylor bounds verify the table independently; the uniform score error is at most $\epsilon_E = 730/M < 2.594 \cdot 10^{-12}$ on the declared domain. A regret statement transfers to real softmax with $2\epsilon_E$ added to its tolerance. The finite claim itself uses the rational score, not rounded floating-point softmax comparisons.

For a remaining integer radius b, add b to a currently maximal logit and subtract b from the others. Monotonicity of the table gives the exact finite-score endpoint support, attained simultaneously by aligned integer updates. A score difference is compared with rational $a = u/v$ through $Mv(d_2 - d_1) \le u d_1 d_2$, using arbitrary-precision integers. No tolerance is introduced to decide acceptance. Prefix normalization and table lookup are real operations; the "one comparison" first-exit implementation refers only to the additional cutoff decision after normalization and offline compilation.

At each inspected exit the general implementation maintains current logits, all available score denominators (or equivalent past-reward state), the minimum denominator and earliest return index, active input indices, and paid check count. It tests every observed-past constraint and each remaining endpoint support. Rejected inputs alone execute the next residual block. This direct general implementation does more work than the compiled first-exit branch; neither is assigned the other's timing. Invalid or out-of-domain arithmetic must fail closed rather than become a certificate. The released replay verifies all visited finite states; this is not a claim that arbitrary operating-system or NaN-producing executions are verified.

D.4.1 A SHARP GUARANTEE FOR THE RETURNED DECISION

Confidence regret and task loss are different quantities. A useful bridge is to preserve the decision made by the entire full-recall network, without requiring every intermediate head to keep the same label. Let a be the label currently recalled, let $v = \max_{k \neq a} \operatorname{softmax}(z_t)_k$, and put $r = \sum_{s>t} \beta_s$. The current prefix need not be a record.

Theorem 9 (Full-recall return preservation)**.** *With earliest-exit tie-breaking, every completion in $\mathcal{A}_t$ has full-recall returned label a if and only if*

$$\frac{e^{2r} v}{1 + (e^{2r} - 1)v} \leq B_t. \tag{19}$$

When this holds, stopping preserves $\ell(a, y)$ relative to full recall for every *possible true label y and every loss depending only on the returned decision. It does not establish that $a = y$ or preserve a probability-dependent loss such as cross-entropy.*

The proof, including the necessity witness, is in Appendix G.

The exact finite analogue checks alternate-class boosted endpoints: a harmful endpoint has a label other than a and a denominator strictly smaller than the observed record. Testing all alternate classes handles quantization and the deterministic class tie rule. Conjoining this guard with (16) yields both a charged score-regret certificate and full-recall decision-loss preservation. It does not certify the unknown label or remove the architectural approximation to the trained suffix. Appendix G gives the finite argument, state sufficiency, and exhaustive nonrecord-prefix checks.

Corollary 10 (Whole-policy decision-loss preservation)**.** *Assume early and full execution share the same observed-prefix semantics. A controller that conjoins the scalar and return tests, and otherwise falls back to full recall, returns the full-recall label on every valid input. Its expected decision loss therefore equals full recall under any joint input/label distribution for which the expectation exists, although its scalar regret guarantee remains selective to certified stops.*

D.5 LEARNING THE CONTRACT RATHER THAN SATURATING IT

The contract restricts output amendments, not their training algorithm. If a persistent feature permits arbitrary readouts into $[-B, B]^K$, every amendment $a(h)$ in that cube can be split as $\eta_j(h) = \beta_j a(h)/B$, where $B = \sum_j \beta_j > 0$; conversely, every admissible sum lies in the cube. This capacity statement does not prove trainability. A hard clamp has zero training derivative outside the budget, and the original hard $\beta = .25$ family clips approximately 99.8–100% of raw update coordinates. Even the .5 family clips roughly 96–98%. The constraint is active, not free.

We retain the same quantized deployment radius 16 and compare small-output initialization, $\beta \tanh(u/\beta)$ readouts, straight-through (STE) training, and projected unconstrained training. Small initialization uses training features only. The smooth derivative is $\operatorname{sech}^2(u/\beta)$, whereas STE changes only the backward rule. Hidden states remain independent of persistent logits. For unconstrained updates u_s, write $e_s = \|u_s - \operatorname{clip}(u_s, -\beta_s, \beta_s)\|_\infty$. Projection gives

$$\|z_j - \widetilde{z}_j\|_\infty \leq E_j := \sum_{s=1}^{j} e_s + (j+1)/(2S). \tag{20}$$

A head margin above $2E_j$ preserves that head's label; confidence changes by at most $E_j/2$ and cross-entropy by $2E_j$. These are not full-recall claims; proofs and complete earlier analyses remain in the appendices.

The five frozen convolutional prefixes, seeds 3101–3105, split, 128-dimensional features, optimizer family, and deployment precision are held fixed. Original four-epoch endpoints and all 45 new fixed checkpoints are retained. New fits have four- and twelve-epoch endpoints; continuation adds eight epochs with a fresh recorded optimizer state. This is a registered post-review extension on an already public test set, not untouched confirmation. No seed or checkpoint is selected by test accuracy.

The original bounded terminal accuracy is 85.380%, versus 86.070% for the paired unconstrained fit: a -0.690 ± 0.487 point paired difference (sample standard deviation). At twelve epochs, continued hard training reaches 85.416%, small hard initialization 86.472%, and small smooth initialization

Table 8: Optimization under the same $\beta = 0.25$ deployment contract, five paired seeds. Accuracy is in percent; saturation is the percentage of final-stage coordinates above 99% of the cap magnitude before rounding. Every registered endpoint is retained. The unconstrained row is a comparator, not a certified bounded model.

Readout/training	Epochs	Terminal	Recall	Saturation
Hard, continued	12	85.416	85.410	100.0
Small init., hard	4	86.170	85.866	84.1
Small init., hard	12	86.472	86.172	86.1
Small init., smooth	4	86.308	85.980	23.1
Small init., smooth	12	86.712	86.348	50.3
Straight-through	4	85.444	85.420	56.4
Straight-through	12	85.600	85.518	10.3
Projected + straight-through	12	85.576	85.538	37.9
Unbounded, continued	12	86.854	86.706	–
Unbounded then projected	12	86.276	85.894	61.0

Table 9: Executed depths for the general controller: mean counts per 10,000 test inputs across all five seeds, $\lambda/\mathrm{price} = 1/3$. Depths 0–2 are certified; depth 3 is compulsory, uncertified fallback. Network MACs and total charged credits per input are in thousands; full execution uses 225.184 thousand MACs. All inspections, including failures, cost 4,096 credits.

Model	Price	$T = 0$	$T = 1$	$T = 2$	$T = 3$	MACs	Credits
Hard (R31)	0.01	3294.8	79.2	23.0	6603.0	207.40	216.96
Hard (R31)	0.02	4053.2	99.2	33.0	5814.6	203.30	212.22
Hard (R31)	0.04	4906.0	117.8	28.6	4947.6	198.72	206.94
Hard (R31)	0.08	5930.4	126.2	30.8	3912.6	193.26	200.64
Smooth (R32)	0.01	3294.8	211.6	178.6	6315.0	206.66	216.16
Smooth (R32)	0.02	4053.2	221.2	180.8	5544.8	202.60	211.48
Smooth (R32)	0.04	4906.0	238.6	181.0	4674.4	198.02	206.19
Smooth (R32)	0.08	5930.4	255.0	173.0	3641.6	192.55	199.88

86.712%. The smooth final-stage near-cap occupancy is 50.331%, with mean derivative 0.1985; continued hard training has zero derivative at its last two stages. Raw smooth preactivation beyond the cap is not a hard-clipping event. Table 8 and the archived stagewise diagnostics retain both definitions.

Paired t intervals use five seeds and four degrees of freedom, not 50,000 independent training replications. Smooth minus continued hard is $+1.296$ points, 95% interval $[0.865, 1.727]$; smooth minus small hard is $+0.240$ $[-0.108, 0.588]$. The combined repair is supported, but a smooth-specific advantage over suitable initialization is not resolved. The unconstrained twelve-epoch model reaches 86.854%; smooth minus unconstrained is -0.142 $[-0.521, 0.237]$. Containing zero does not establish equivalence. Projection reaches 86.276%, while STE and projected STE reach 85.600% and 85.576%. All negative methods and unadjusted paired comparisons remain visible.

D.6 Execution, sensitivity, and measured resource benefit

We evaluate the original hard $\beta = 0.25$ family and the registered smooth 12-epoch family at all four total suffix prices $p_c \in \{.01, .02, .04, .08\}$. Each of the three transitions costs $p_c/3$. The prefix costs 172,192 MACs, each residual correction costs 17,664, and full execution costs 225,184. Each inspection is assigned 4,096 work credits, valued at $4096p_c/52992$ in the same objective; all failed inspections are charged. This declared work objective is distinct from a worst-case latency bound. The schedules are the compiled first-exit baseline and the complete $(0, 1, 2)$ schedule, with compulsory final fallback. A third mode adds the full-recall return guard to the complete schedule.

D.6.1 Actual intermediate-depth decisions

Table 9 shows that the general controller executes a real multi-depth policy. At price .04 and tolerance ratio $\lambda/p_c = 1/3$, the hard family stops 117.8 inputs at exit 1 and 28.6 at exit 2 per 10,000; the

smooth family stops 238.6 and 181.0. Initial-exit counts agree across families because the prefixes and deployment bounds are identical. This agreement is predicted by the information contract, not a fitted equality of policies. Among inputs that continue, the trained suffix changes the observed state and therefore later certified depths.

Every accepted early stop satisfies the registered charged score tolerance. At the default ratio, both families also preserve actual full-recall labels, and the additional return guard changes no actions. Nevertheless, terminal and recall accuracy are different: the smooth family's terminal accuracy is 86.712%, while full recall and certified stopping return at 86.348%. For the original hard family the corresponding recall value is 85.446%. The task claim is preservation relative to the specified recall baseline, not accuracy dominance over the raw terminal head.

With the same history and charges, structural supports also improve on trivial $g_j \leq 1$ headroom. At price .04, stopping is strictly earlier on 1,256.2 hard-family and 1,529.4 smooth-family inputs per 10,000. Every one of the 40 matched seed–model–price conditions satisfies rowwise depth dominance; full results remain in `R32_HEADROOM.json`.

The direct multi-exit controller reduces network work beyond the first-exit baseline, but extra inspections can more than offset this difference. For the hard family at price .04, average network work falls from 199,186.12 to 198,719.44 MACs, while charged work rises from 203,282.12 to 206,940.20 credits. Paired intervals for both differences are provided with all four prices. Both remain below full execution under the declared credit model, but first-exit compilation is cheaper. This does not contradict Theorem 8: optimality is conditional on a fixed schedule and information state, not a theorem that additional inspections are free or that every schedule improves expected resource cost.

Actual masked execution covers five seeds, two families, four prices, and two schedules: 80 executions. All executed-depth, returned-exit, and label decisions match frozen replay. Three of 20,913,040 quantized coordinates differ under changed floating-point batch shapes, without changing these decisions. The contract is checked on actual outputs, not on an assumption of bit-identical floating kernels.

D.6.2 Utility sensitivity and nonrecord prefixes

The original grid contains 600 seed–model–price–ratio–mode conditions, using ratios $\lambda/p_c \in \{0, 1/6, 1/3, 1/2, 1\}$. All are retained. Positive fees preclude certification at zero tolerance. At price .04 the hard scalar policy certifies 4,600.6, 5,052.4, 5,411.8, and 7,350.4 inputs per 10,000 as the ratio increases from $1/6$ to $1/3$, $1/2$, and 1. Nonrecord accepted prefixes reach 85.4 and 1,343.6 at the last two ratios, exercising both recall and past-regret state.

At ratio 1, scalar stopping changes 7.0 actual full-recall labels and turns 3.6 correct recalled predictions into errors; adding the return guard reduces coverage to 7,280.8 and eliminates those disagreements. The corresponding smooth-family counts are 13.4 changed labels and 9.0 harmful changes, removed by reducing coverage from 7,433.2 to 7,374.0. These observations do not replace the pointwise proof. At the reference ratio $1/3$, the hard multi-exit policy has 4,947.6 final fallbacks, of which 4,292.6 exceed the charged tolerance. They are not accepted certificates. Historical confidence heuristics with score violations and the full sensitivity tables remain in the appendices.

D.7 Selecting the inspection schedule

An inspection schedule specifies where to attempt certification, not which intermediate predictions to retain. All executed exits remain available to recall. For a fixed candidate and fee f, let $\sigma \subseteq \{0, 1, 2\}$ and

$$w_\sigma(x) = 17{,}664\, t_\sigma(x) + f n_\sigma(x), \qquad \widehat{\sigma} = \arg\min_{\sigma \in \Sigma} \frac{1}{n} \sum_{i=1}^{n} w_\sigma(x_i). \tag{21}$$

The common 172,192-MAC prefix is omitted only from the minimization. Every rejected check contributes to n_σ. Ties prefer fewer inspections then lexicographic order. Neither labels nor test utility enter this selection. At zero fee, adding inspections cannot delay a first valid stop. With positive fees, an earlier rejection can exhaust tolerance and invalidate a later stop; Appendix E gives a two-class constructive counterexample. Therefore schedule inclusion does not imply cost or depth dominance.

Table 10: Training-only schedule selection at price .04 and $\lambda/p_c = 1/3$. All five seeds select the shown schedule. Work includes the prefix and all check fees; accepted is per 10,000. Intervals elsewhere use paired seeds. The scalar policy does not automatically certify return preservation.

Fee	Schedule	Exit 0	Exits 1–2	Accepted	Work
0	$\{0, 1, 2\}$	5,022.2	589.0	5,611.2	197,048.8
4,096	$\{0\}$	4,906.0	0.0	4,906.0	203,282.1
16,384	$\{0\}$	4,320.0	0.0	4,320.0	218,675.5

Theorem 11 (Finite-family work selection)**.** *Fix a trained candidate and m schedules with $0 \le w_\sigma \le H$. If calibration inputs are iid from P and independent of candidate fitting, then with probability at least $1 - \delta$, for every Q with $\mathrm{TV}(P, Q) \le \rho$,*

$$\mathbb{E}_Q w_{\widehat{\sigma}} - \min_{\sigma \in \Sigma} \mathbb{E}_Q w_\sigma \le 2H \sqrt{\frac{\log(2m/\delta)}{2n}} + 2H\rho. \tag{22}$$

Each selected schedule retains the original pointwise accepted-stop certificate. This is not optimality over all observation-adaptive policies.

The proof is a finite union bound and a total-variation comparison. The historical reserved split supplies an exact empirical minimizer, not a claim that this particular public benchmark is a fresh iid replication. With $m = 8$, $n = 5,000$ and $\rho = 0$, the bound describes a separate conditional statistical guarantee; it is not a confidence interval on test-oracle gaps.

The completed registered study crosses eight schedules, three fees $\{0, 4096, 16384\}$, four prices and five ratios for every seed: 2,400 conditions and 24 million reused input–policy cases. The training freeze is verified against all five disjoint controller splits before applying selected schedules to test results. Independent code reproduces all 2,400 calibration work sums, all 24 million test depths and check counts, and 3,600 additional Fraction checks. All original results, including rejected checks and terminal regrets, are now included in the persistent release rather than only an Actions artifact.

At price .04 and ratio $1/3$, all seeds select every exit when checks are free, saving 1,521.58 MACs over first-exit inspection, paired 95% interval $[964.36, 2078.79]$. At fee 4,096 every seed instead selects exit 0 alone, saving 3,658.07 work credits versus dense inspection, interval $[3451.05, 3865.09]$. This changes the implemented schedule, not the theorem's objective. Across the 300 seed/configuration choices, the selected schedules are first-only (116), all exits (93), none (65), $\{0, 2\}$ (17), and $\{0, 1\}$ (9). No-check choices include conditions that cannot repay fees.

All 60 configurations are reported in Appendix E. Training-selected work matches the descriptive test-best mean in 55; the other gaps are retained, averaging 3.08 credits over the complete grid and reaching 104.70 at worst. Those test minima are diagnostic oracles, never selection inputs. The schedule study uses the scalar guard: equal mean task accuracy does not imply identical labels. At zero fee in the reference row, 1.2 labels per 10,000 change despite equal mean accuracy. The separate return guard is required for the pointwise decision-loss claim.

D.7.1 End-to-end runtime and setup-inclusive accounting

The retained measurements time complete image inference: prefix, score normalization, controller, masks, executed suffix, and return. The same frozen seed-3101 candidate runs on AMD EPYC 9V45 x86-64 and an ARM runner exposing only the processor identifier `aarch64`; a commercial ARM model is not identified. Both use one Torch/BLAS thread and nine randomized paired rounds per fixed workload, with idle and one-competing-worker regimes. The environment, training-only admission decisions, all 108 paired rounds and unsuccessful methods remain in the artifact.

At 1,000 images under idle load, full/first/multi times are 5.432/4.799/7.017 ms on x86 and 19.970/18.638/22.103 ms on ARM. The full-to-first ratios are about 1.13 and 1.07, not a hardware-independent speedup. Ridge controller times are 5.279 and 19.890 ms, including actual controller work. The general Python gate is slower despite fewer network MACs; the schedule experiment optimizes declared credits, not these measured wall-clock times. The complete calibrated path has its own confidence objective and does not automatically inherit the original certificate.

Table 11: End-to-end idle CPU time (milliseconds per fixed workload), including image prefix, normalization, policy, masking, suffix and return. Medians of nine randomized paired rounds; batch 1 processes 128 images, batch 32 processes 992, and batch 1,000 processes 1,000. Raw pairs, competing-worker results, training-only admission, and setup-inclusive curves are retained. Calib. denotes the full temperature-calibrated path, not a certificate.

CPU	Batch	Full	First	Multi	Ridge	Calib.
X86	1	15.77	11.96	33.47	37.78	17.79
X86	32	10.91	10.33	22.50	18.39	11.64
X86	1000	5.43	4.80	7.02	5.28	5.72
ARM	1	70.80	53.03	105.79	101.35	74.88
ARM	32	66.94	65.04	89.96	76.74	68.37
ARM	1000	19.97	18.64	22.10	19.89	20.52

Table 12: Matched Fashion-MNIST diagnostic: 128 label-free uniformly sampled fixed prefixes, hard $\beta = 0.25$ seed-3101 suffix, three conditional hidden-box radii. Structural acceptance is 65/128 at every radius. Standalone and intersected (Int.) acceptances are uncharged diagnostics, not charged deployment decisions. ΔQ is mean reduction of the structural cap. Times exclude the separate 0.027 s graph construction. All floating bounds include stagewise rounding allowances; they are not outward-rounded machine proofs.

Radius	Method	Standalone	Int.	ΔQ	Seconds
0	IBP	97	97	0.07310	0.00612
0	CROWN	97	97	0.07310	0.11415
0	CROWN-Optimized	97	97	0.07310	4.42484
0	IBP-split4	97	97	0.07310	0.01360
0.01	IBP	97	97	0.06335	0.00485
0.01	CROWN	97	97	0.07310	0.24901
0.01	CROWN-Optimized	97	97	0.07310	23.16868
0.01	IBP-split4	97	97	0.06356	0.01331
0.1	IBP	69	69	0.02438	0.00561
0.1	CROWN	97	97	0.07310	0.34234
0.1	CROWN-Optimized	97	97	0.07310	44.91505
0.1	IBP-split4	69	69	0.02452	0.01302

For incremental setup S_0 and repeated workload times F, G, total ratio and strict break-even are

$$\frac{kF}{S_0 + kG}, \qquad k_{\min} = \left\lfloor \frac{S_0}{F - G} \right\rfloor + 1 \quad (F > G); \qquad \text{none if } F \le G. \tag{23}$$

The measured incremental setup is 2.010 ms on x86 and 2.776 ms on ARM. Median paired gains recover it after four and three 1,000-image idle workloads, respectively. These totals exclude common model/framework loading and offline policy training; they are not complete cold-start measurements. The x86 competing-worker mean saving interval crosses zero despite a positive median. All paired intervals and earlier 13–15-workload setup results remain in the appendices. An empirical tariff is not a worst-case latency bound.

D.7.2 What equal verifier decisions do and do not imply

The matched diagnostic uses 128 label-free uniformly sampled Fashion-MNIST prefixes of the same hard seed-3101 suffix and conditional hidden-state boxes. The authors' auto_LiRPA implementation receives weights and hidden-box information that the structural rule ignores. The comparison transforms bounds into the same score-regret predicate, not a different classification predicate. Acceptance counts are *uncharged diagnostics*; verifier cost is not hidden inside the deployed integer certificate.

The rowwise audit distinguishes equal bounds from equal decisions. At radius 0 all methods have identical raw endpoint intervals and 97 accepts, versus 65 structural accepts. At radius .01, IBP and CROWN still accept the same 97 rows, but their maximum cap difference is nonzero. CROWN resolves all 3,840 raw endpoint coordinates to zero width after removing quantization padding,

whereas IBP resolves only 2,999. At radius .1, IBP and split IBP accept 69, while CROWN and optimized CROWN accept 97; IBP/CROWN disagree on 28 specific rows. CROWN and optimized CROWN retain respectively 3,828 and 3,831 flat raw coordinates. Hard-clipped outputs explain this local collapse; remaining bound improvements need not cross a decision threshold.

These are numerical observations at tolerance 10^{-12} for interval-width reporting, not exact arithmetic proofs from the verifier. Floating relaxations and explicit stagewise rounding allowances are not outward-rounded end-to-end machine certificates. The deployed scalar gate instead uses the proved integer table. All domains, method-specific arrays, pairwise decisions, graph construction and per-method times are retained; no broad ranking of verifier families follows.

D.7.3 Deployment state, budget choice, and shift

For a fixed utility, recall needs current logits, the best score and earliest return index, the largest past net reward, and cumulative charge. Updating the past maximum compresses a finite history without omitting a comparator exit; Appendix E proves sufficiency. Thus fixed-configuration state is $O(K)$, although the direct support check costs $O(K(L-t))$ and arbitrary-precision comparison cost also depends on bit length. Faster inspection, rather than memory of every denominator, is the main controller engineering target. Transformer hidden computation is not covered by an output-amendment capacity or timing claim.

The extension fixes $\beta = .25$ in its training protocol, after the preceding public results were known. It is not claimed as a globally untouched choice or an optimum. In a new application, budgets, precision, utility, and schedules should be selected and frozen on separate training/calibration data, including their task and resource costs. Dataset shift need not invalidate an enforced per-input increment bound, but can change task accuracy, coverage and expected work. Equation (22) isolates one conditional work-shift penalty; no shifted-data performance is inferred. Changing temperature or calibration changes scores, records and supports, which must be recomputed. The history of weaker grid, CIFAR-100, SVHN, MNIST, and learned-policy results remains intact rather than being substituted by these positive findings.

Cost calibration also has a measurable sampling price. At the reference fee, $H = 65,280$, $m = 8$, $n = 5,000$, and $\delta = .05$ give an excess-work bound of approximately 3,136 credits under independent calibration and zero shift. This conservative bound is not the observed oracle gap and is not used to choose a favorable price. Selecting budgets as well as schedules requires a family that includes every considered candidate and an evaluation split independent of candidate fitting; the same concentration statement cannot simply be reused after tuning on that split.

D.8 Conclusion

Coupled output increments turn an unknown suffix into an attainable extremal completion. The stopping certificate is information-optimal for the fixed inspection order, full finite observed record, confidence-resource utility, and declared charges. Its separate return guard preserves full-recall labels, not unknown-label correctness. Controlled training repairs severe saturation without relaxing the deployed bound. Completing the registered schedule study then shows when free checks help and paid checks should be omitted, while training-only selection and a finite-family bound keep test oracles outside the decision. Exact finite replay, actual masked execution, rowwise verifier comparisons, and scoped CPU measurements make these distinct claims auditable.

E Schedule selection, sufficient state, and source-level reconciliation

This appendix separates a theorem about an observed completion problem from selection among executable fixed inspection schedules. The latter is an expected-work problem. It is not resolved by calling the first accepted prefix information-optimal under a schedule that was held fixed.

E.1 A SUFFICIENT FINITE RECORD FOR A FIXED UTILITY

For every executed exit, compute its score even when its expensive support inspection is omitted: this is required by the full-recall semantics. Set $A_t = \max_{j \le t}(B_j - C_j)$. The updates

$$B_t = \max(B_{t-1}, g_t), \qquad A_t = \max(A_{t-1}, B_t - C_t), \qquad P_t = A_t - (B_t - C_t)$$

show that the whole prefix comparator is retained by two maxima, not a list of every historical logit. Keep also the earliest index achieving B_t, its label, current logits z_t, and cumulative inspection charge Z_t. The future supports depend only on z_t, the remaining budgets and charges, and B_t. Thus this state reconstructs the exact charged certificate and return guard at every inspected prefix. It neither drops an executed exit from the comparator nor assumes that the current prefix is a record.

For a finite score, store the best denominator and a denominator/cost pair attaining A_t. Comparisons can be cross-multiplied as arbitrary-precision integers. With a fixed rational utility configuration, the number of scalar records is constant; their bit length is not assumed constant. This proves $O(K)$ state in the number of logit coordinates, excluding model hidden activations, with direct $O(K(L-t))$ support evaluation. Recompiling the utility price retrospectively may change which old exit attains A_t; therefore this compression is not advertised as sufficient for arbitrary post-hoc utility changes. Keeping all old records supports those changes at additional cost. The supplied original replay retains full denominators; the recurrence is a proven implementation alternative, not a new timed optimized kernel.

E.2 FREE INSPECTION AND A PAID-INSPECTION COUNTEREXAMPLE

Suppose all inspection charges are zero and schedules $\sigma \subseteq \tau$ use identical score and recall semantics. At every common inspected depth, the same prefix has the same acceptance predicate. If τ does not accept before the first successful σ depth, it reaches that depth with the same prefix and accepts there. Therefore $t_\tau(x) \le t_\sigma(x)$, including compulsory terminal execution. Nonnegative transition work then gives the same ordering of network work. This assertion does not apply to actual wall-clock time when an ostensibly "free" inspection takes time.

Positive charges invalidate the argument. Let $K = 2$, $L = 2$, $z_0 = (0, 0)$, $\beta_1 = 1$, $\beta_2 = 0$, $z_1 = z_2 = (1, -1)$, and $c_1 = c_2 = .01$. Take an inspection charge .06 and tolerance .10. At exit 0 the best possible future confidence is $e/(e + e^{-1}) \simeq .880797$. The future gain minus the first transition cost, plus the check, exceeds the tolerance. Under $\tau = \{0, 1\}$ this check fails. At exit 1 the observed past regret and future positive improvement are zero, but the two paid checks cost .12, so it fails again and reaches exit 2. Under $\sigma = \{1\}$, exit 1 incurs only .06 and is certified. Thus adding a paid inspection strictly increases executed depth on an admissible trajectory. This is a constructive example of an avoidable charging failure, not a computational impossibility result.

E.3 PROOF OF THE FINITE-FAMILY SELECTION THEOREM

Condition on a trained candidate independent of the calibration sample. For any fixed schedule, $w_\sigma(X) \in [0, H]$. To make the concentration argument self-contained, the bounded-variable exponential moment satisfies $\log \mathbb{E} \exp(s(Y - \mathbb{E}Y)) \le s^2 H^2/8$: its second derivative is the variance under the exponentially tilted law, at most $H^2/4$, and its value and first derivative at zero vanish. Independence, exponential Markov inequality, and minimization over s therefore give

$$\Pr\{|\widehat{W}_\sigma - W_{P,\sigma}| > u\} \le 2 \exp(-2nu^2/H^2).$$

A union bound over m schedules gives a simultaneous deviation of at most $u = H\sqrt{\log(2m/\delta)/(2n)}$ with probability $1 - \delta$. Let σ_P minimize $W_{P,\sigma}$. On that event,

$$W_{P,\widehat{\sigma}} \le \widehat{W}_{\widehat{\sigma}} + u \le \widehat{W}_{\sigma_P} + u \le W_{P,\sigma_P} + 2u.$$

For any Q, a $[0, H]$-valued function has expectation difference at most $H\,\mathrm{TV}(P, Q)$ under the convention $\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)|$. Comparing both the selected schedule and the Q-minimizer yields $2u + 2H\rho$. The event does not depend on Q, so the assertion holds simultaneously for all such shifts. With $L = 3$, one valid range bound is $H = 3(17,664 + f)$ after subtracting the common prefix. Adding that fixed prefix back changes neither the empirical minimizer nor the excess-work comparison.

The pointwise scalar and returned-label certificates need no sampling assumption: selecting any one of the admissible schedules cannot turn a failed certificate into an accepted one. The statistical bound adds an expected-work comparison only under its separate sampling assumptions. The historical fixed data split is not promoted into independent-dataset evidence. A general observation-adaptive inspection policy would require its own information state, transition model and objective; enumeration here proves empirical optimality only in the eight registered fixed schedules.

E.4 Training-only selection and immutable execution products

The R33 protocol and readable programs precede workflow execution. The calibration program accesses 5,000 reserved training rows per seed, discards labels for its selection objective, and verifies separation from fitting rows. It records every candidate's integer work sum before the test replay program runs. The preserved freeze file contains all 300 selected schedules and source, model, score-table, split and calibration-trace hashes. Original logs and workflow/source identifiers fix the order of calibration and test steps.

The original combined workflow artifact was created after both steps, not a separate public commit before test replay. We do not describe that chronology as an independently timestamped pre-test publication. Its provenance rests on the preregistered source, workflow step order and frozen output, subsequently persisted together. The new independent replay recomputes all 2,400 calibration work sums from the saved calibration traces and reproduces all choices before joining them to the test results. It does not choose schedules by test loss, test work or oracle ranking.

The 24 million schedule/input cases repeatedly reuse five sets of 10,000 test rows. They are implementation-replay counts, not 24 million independent observations. The full original grid, including no-inspection plans, adverse fees, terminal fallback regrets and task-label changes, is included under `r33_evidence`. New results are under `r34_results`.

Table 13: All 60 registered configurations of the training-selected schedule. Five entries in the schedule column are seeds 3101–3105; A dash marks no inspection. Work is mean credits; accepted is per 10,000. Oracle gap compares to descriptive test-best fixed schedules and is never used for deployment.

Fee	Price	Ratio	Selected schedules	Work	Accepted	Oracle gap
0	0.01	0	01,012,01,01,012	210312.7	2841.8	0.4
0	0.01	1/6	012,012,012,01,012	208017.4	3309.2	0.4
0	0.01	1/3	012,012,012,012,012	205961.3	3932.0	0.0
0	0.01	1/2	012,012,012,012,012	204939.3	4177.2	0.0
0	0.01	1	012,012,012,012,012	200757.9	5764.2	0.0
0	0.02	0	01,012,012,01,012	206513.2	3571.2	0.0
0	0.02	1/6	012,012,012,012,012	203949.0	4092.8	0.0
0	0.02	1/3	012,012,012,012,012	201809.9	4709.2	0.0
0	0.02	1/2	012,012,012,012,012	200636.0	4990.4	0.0
0	0.02	1	012,012,012,012,012	196203.0	6623.6	0.0
0	0.04	0	01,012,012,01,012	202237.8	4387.8	0.0
0	0.04	1/6	012,012,012,012,012	199397.4	4955.8	0.0
0	0.04	1/3	012,012,012,012,012	197048.8	5611.2	0.0
0	0.04	1/2	012,012,012,012,012	195691.5	5938.0	0.0
0	0.04	1	012,012,012,012,012	190772.1	7672.6	0.0
0	0.08	0	012,012,012,01,012	197462.5	5291.2	0.0
0	0.08	1/6	012,012,012,012,012	193980.9	5995.6	0.0
0	0.08	1/3	012,012,012,012,012	191050.1	6801.4	0.0
0	0.08	1/2	012,012,012,012,012	189145.6	7269.6	0.0
0	0.08	1	012,012,012,012,012	182846.2	9268.0	0.0
4096	0.01	0	–,–,–,–,–	225184.0	0.0	0.0
4096	0.01	1/6	0,0,0,0,0	213479.9	2981.6	0.0
4096	0.01	1/3	0,0,0,0,0	211820.2	3294.8	0.0
4096	0.01	1/2	0,0,0,0,0	210925.7	3463.6	0.0

Fee	Price	Ratio	Selected schedules	Work	Accepted	Oracle gap
4096	0.01	1	02,0,02,02,02	208224.6	5329.2	0.0
4096	0.02	0	–,–,–,–,–	225184.0	0.0	0.0
4096	0.02	1/6	0,0,0,0,0	209583.9	3716.8	0.0
4096	0.02	1/3	0,0,0,0,0	207801.3	4053.2	0.0
4096	0.02	1/2	0,0,0,0,0	206747.8	4252.0	0.0
4096	0.02	1	02,02,02,02,02	203538.4	6449.2	0.0
4096	0.04	0	–,–,–,–,–	225184.0	0.0	0.0
4096	0.04	1/6	0,0,0,0,0	205291.6	4526.8	0.0
4096	0.04	1/3	0,0,0,0,0	203282.1	4906.0	0.0
4096	0.04	1/2	0,0,0,0,0	202044.2	5139.6	0.0
4096	0.04	1	02,02,02,02,02	197955.7	7452.8	0.0
4096	0.08	0	–,–,–,–,–	225184.0	0.0	0.0
4096	0.08	1/6	0,0,0,0,0	200196.9	5488.2	0.0
4096	0.08	1/3	0,0,0,0,0	197853.6	5930.4	0.0
4096	0.08	1/2	0,0,0,0,0	196280.8	6227.2	0.0
4096	0.08	1	012,02,012,02,02	190469.8	8847.0	39.1
16384	0.01	0	–,–,–,–,–	225184.0	0.0	0.0
16384	0.01	1/6	–,–,–,–,–	225184.0	0.0	0.0
16384	0.01	1/3	–,–,0,–,–	225224.0	610.8	40.0
16384	0.01	1/2	0,–,0,0,0	224481.5	2606.0	104.7
16384	0.01	1	0,0,0,0,0	221891.0	3713.2	0.0
16384	0.02	0	–,–,–,–,–	225184.0	0.0	0.0
16384	0.02	1/6	–,–,–,–,–	225184.0	0.0	0.0
16384	0.02	1/3	0,0,0,0,0	222884.1	3525.8	0.0
16384	0.02	1/2	0,0,0,0,0	220622.4	3952.6	0.0
16384	0.02	1	0,0,0,0,0	217646.4	4514.2	0.0
16384	0.04	0	–,–,–,–,–	225184.0	0.0	0.0
16384	0.04	1/6	–,–,–,–,–	225184.0	0.0	0.0
16384	0.04	1/3	0,0,0,0,0	218675.5	4320.0	0.0
16384	0.04	1/2	0,0,0,0,0	216090.5	4807.8	0.0
16384	0.04	1	0,0,0,0,0	212660.9	5455.0	0.0
16384	0.08	0	–,–,–,–,–	225184.0	0.0	0.0
16384	0.08	1/6	–,–,–,–,–	225184.0	0.0	0.0
16384	0.08	1/3	0,0,0,0,0	213801.3	5239.8	0.0
16384	0.08	1/2	0,0,0,0,0	210769.0	5812.0	0.0
16384	0.08	1	0,0,0,0,0	206456.6	6625.8	0.0

E.5 A CLOSER AUDIT OF VERIFIER EQUALITY

The rowwise audit uses the original 128 frozen input indices and all twelve method/radius arrays. Acceptance applies the same confidence-regret predicate with $p_c = .04$, $\lambda = p_c/3$ and no verifier fee. All method-specific standalone bounds and structurally intersected bounds are retained. The methods are not assigned different class-prediction objectives.

At zero hidden-box radius, interval propagation and linear relaxations both resolve the deterministic unquantized suffix, after which the same explicit rounding pad is added. At radius .01, CROWN and optimized CROWN resolve all 3,840 raw output coordinates to width below 10^{-12}; IBP resolves only 2,999. Their equal decisions do not mean identical envelopes: the IBP improvement does not change which side of the tolerance each of these 128 examples lies. At radius .1, the claim of identical aggregate decisions is false: IBP and split IBP accept 69 rows; CROWN and optimized CROWN accept 97. The new JSON gives the 28 disagreeing row positions, every method's Boolean vector, all pairwise maximum cap differences, and interval-width summaries.

A raw interval width is computed after subtracting the existing padding $(j/128)$ at each endpoint. Many remaining widths collapse because the bounded readout clamps are locally saturated. This explains local numerical behavior of the frozen suffix; it is not evidence that clipping is harmless for training or that the methods agree on general architectures. The diagnostic still uses float64 library

relaxations, not outward rounding. Its tolerances for reporting equal widths are not used in the exact integer stopping gate.

E.6 Budget choice, objective calibration, and broader architectures

The $\beta = .25$ extension budget was fixed in its protocol before the new training runs, after earlier public-test findings were available. The preceding study also retains .125 and .5; no claim is made that .25 was selected without any knowledge of that benchmark. For a new application, a training-only criterion can minimize declared work over a predeclared budget family subject to a separately stated held-out task-loss constraint. That criterion needs labels for its task constraint, unlike the R33 schedule-only criterion. It requires independent calibration or an explicit complexity correction and is not an executed new budget-search result in this release.

A distribution shift that preserves the implemented increment contract does not invalidate a pointwise completion certificate. It can change confidence calibration, acceptance frequency, accuracy and realized resource use. The new work-shift bound separates this issue from the certificate. Temperature scaling requires recomputation of current logits, records and supports with rescaled budgets; a fitted confidence transformation cannot be retrofitted into the old objective without checking its endpoint and recall semantics. Return-label preservation is an equality with the full-recall decision, not a guarantee that either decision is correct under the shifted distribution.

For residual and transformer-style systems, the output contract can be enforced by bounded additive readouts while allowing arbitrary hidden computation. Its utility is an empirical question: hidden computation can dominate total cost; a small output-amendment budget may constrain learning; and repeated exact support checks may cost more than the work they save. The sufficient-state recurrence removes the need to keep every historical scalar record for a fixed utility, but it does not establish a fast GPU kernel, transformer training result, energy saving, or tail-latency guarantee. All such distinctions are retained next to the positive CPU and small-model evidence.

F Preserved R32 main-text development

This is the complete preceding main-text exposition, including its tables. Numerical claims retain their original settings; the current main text and provenance reconciliation identify subsequent additions. Internal labels are namespaced only to avoid duplicate cross-references.

Preserved abstract. When can a neural network stop without computing the suffix that justifies the stop? We give an information-optimal answer for bounded additive logit corrections and a declared confidence-minus-resource objective with recall. A single aligned completion attains every future confidence support, making the certificate sharp for the coupled increment contract. We implement this result at every nonterminal exit, including nonrecord prefixes and the charges of unsuccessful inspections, with exact integer score semantics. A separate sharp criterion preserves the full-recall returned label for every possible true label. A five-seed convolutional Fashion-MNIST extension addresses the optimization cost of the contract: small-output smooth readouts improve terminal accuracy by 1.296 percentage points over continued hard-clipped training, with a paired 95% interval of [0.865,1.727], under the same deployed increment bound. Actual masked execution matches all registered stopping and return decisions. Intermediate stops reduce network work, although their additional inspections can cost more than a first-exit gate. Measurements on x86-64 and ARM64 include learned-controller overhead, competing load, and setup-inclusive break-even analysis. Matched Fashion-MNIST verification and preserved adverse results distinguish contract optimality, task behavior, numerical validity, and measured resource benefit.

F.1 Introduction

Adaptive depth requires both a useful prediction and a reason not to execute another layer. Learned halting and routing address the first requirement (Graves, 2016; Teerapittayanon et al., 2016; Huang et al., 2018; Wang et al., 2018). The second is an information problem: evaluating the skipped suffix to decide whether it was worth skipping can consume the proposed saving. Recall makes this distinction unavoidable. A controller that executes three corrections and then returns the prefix label has paid for all three corrections, not for the depth of its returned prediction.

We study this problem through an architectural contract. Each stage adds a coordinatewise bounded correction to a persistent logit vector. From the observed logits and remaining budgets alone, an extremal completion increases a currently maximal coordinate and decreases all competitors. It maximizes softmax confidence simultaneously at every future depth. Consequently, the worst possible loss from stopping has an exact expression, rather than an independent-box relaxation whose endpoints cannot coexist. Any sound rule with the same information, objective, and inspection charges must reject when this extremal completion violates the tolerance. The first successful inspection is therefore the minimum certifiable executed depth on a fixed schedule. This is an operational notion of optimal depth: neither searching architectures nor optimizing the inspection schedule is built into the claim.

Three questions determine whether the construction is useful. First, a current-confidence threshold is insufficient after a nonrecord prefix: both the recalled score and previously available net rewards matter. We implement the full history-dependent test at exits 0, 1, and 2, rather than using a binary first-exit surrogate for the general theorem. Second, almost completely saturated hard clipping can obstruct optimization even when the bounded output class is expressive. We compare initialization, smooth bounded readouts, straight-through training, and projection under an identical deployment contract. Third, fewer neural multiply–accumulates (MACs) need not mean lower total cost. Each failed check must be charged, and setup must be recovered over the actual workload horizon.

Our main contribution is the connection from a sharp information contract to an executable, history-aware controller. Its finite implementation separates three guarantees: confidence-regret validity, preservation of the final full-recall returned label, and a declared resource accounting objective. A registered five-seed extension supplies intermediate-depth execution, optimization ablations, utility sensitivity, paired uncertainty, matched verification, and two instruction-set architectures. It also records the costs of these extensions: the general gate is slower than its first-exit compilation, and the best observed task policy need not be the certified one. These measurements test the construction rather than redefining its objective after seeing the results.

F.2 Related work

The depth-stopping formulation of Qi (2025) motivates the with-recall problem. Adaptive prediction and conditional computation (Bolukbasi et al., 2017; Graves, 2016; Huang et al., 2018; Wang et al., 2018) learn task–compute trade-offs. Regression and neural optimal stopping (Longstaff & Schwartz, 2001; Becker et al., 2019) estimate continuation values; recall-aware threshold models study loss trajectories (Yang et al., 2026b;a). Our guarantee instead quantifies all completions of an architectural increment contract, conditional on an observed prefix. Statistical early-exit prediction sets and compute-risk control (Jazbec et al., 2024; Wang et al., 2026) provide different sampling-based guarantees. A score-regret certificate is not a replacement for their population-risk statements.

Elboher et al. (2025) directly verify early-exit networks, including conditional exit guards and local input robustness. This is a close antecedent. A general verifier can encode our recall and regret predicate; our contribution is the sharp closed form obtained from the output contract, not a new generic solver or an input-robustness theorem. Interval bounds and abstract interpretation (Cousot & Cousot, 1977; Gowal et al., 2018), CROWN and optimized linear relaxations (Zhang et al., 2018; Xu et al., 2021), and split-domain verification (Wang et al., 2021) exploit more of the trained network. We compare the authors' auto_LiRPA implementations on the same Fashion-MNIST suffix and conditional domains. They have access to weights and hidden-state boxes that the structural rule deliberately ignores. The earlier MNIST comparison and historical grid, proof-library, and shortest-path developments remain in the appendices, without being relabeled as new experiments.

F.3 Sharp stopping with recall and inspection charges

A network has exits $0, \dots, L$, scores $g_j \in [0, 1]$, and logits $z_j \in \mathbb{R}^K$. Write $C_0 = 0$, $C_t = \sum_{s=1}^{t} c_s$, $c_s \geq 0$, $B_t = \max_{j \leq t} g_j$, and $R_t = B_t - C_t$. The returned exit is the shallowest maximizer of the observed score; the executed depth is t. Let $Z_t \geq 0$ be the cumulative charge for all inspections through t, including the current check and earlier rejections. We compare the charged candidate $R_t - Z_t$ with the uncharged network-depth comparator $\max_{0 \leq j \leq L} R_j$. This demanding comparator

does not silently give competing depths the controller's overhead. Its regret is

$$\Gamma_t^Z = \max_{0\le j\le L} R_j - R_t + Z_t = \max\{P_t, S_t\} + Z_t, \qquad P_t = \max_{j\le t} R_j - R_t,$$
$$S_t = \Big[\max_{j>t}(g_j - B_t - D_j)\Big]_+. \tag{24}$$

where $D_j = C_j - C_t$. Nonnegative transition costs justify the reduction: a future record is best charged when it appears, and an old record cannot create positive continuation improvement. The past term P_t is already observed; no estimate of the suffix can repair it.

The architectural contract is

$$z_j = z_{j-1} + \eta_j, \qquad \|\eta_j\|_\infty \le \beta_j, \qquad r_j = \sum_{s=t+1}^{j} \beta_s. \tag{25}$$

Let $\mathcal{A}_t$ contain every completion satisfying this coupled contract, not just those reachable through particular trained weights. For max-softmax confidence, put $p = \max_k \mathrm{softmax}(z_t)_k$ and

$$Q_j(p) = \frac{e^{2r_j}p}{1 + (e^{2r_j} - 1)p}, \qquad U_t = \Big[\max_{j>t}\{Q_j(p) - B_t - D_j\}\Big]_+. \tag{26}$$

An empty future maximum contributes zero.

Theorem 12 (Sharp charged certificate)**.** *Under* (25)*,* $\sup_{\mathcal{A}_t} \Gamma_t^Z = \max\{P_t, U_t\} + Z_t$*. Thus stopping is sound at tolerance* λ *exactly when*

$$\max\{P_t, U_t\} + Z_t \le \lambda. \tag{27}$$

No rule using only the same observed scores and logits, increment budgets, charges, and comparator can soundly accept a larger set of prefixes. On a fixed inspection schedule, the first successful test has the smallest certifiable executed depth along that observed trajectory.

Proof. For a fixed class, its softmax probability increases with its own logit and decreases with every competitor. Raising it by r_j and lowering all others by r_j gives the largest class probability in the endpoint cube. Choosing a currently maximal class maximizes that expression over classes and yields $Q_j(p)$. Apply the same signed increment of magnitude β_s at every stage s. This one admissible sequence attains all Q_j simultaneously, so the maximum of the endpoint improvements in (26) is attained by a coupled completion. The past and inspection terms are fixed. If (27) fails, this completion is a witness against any rule with the stated information. Earliest acceptance follows on the fixed schedule; it does not compare different inspection schedules. □

This distinction between a certificate and a fallback is important. An input that fails every inspection executes the final stage and returns its best observed prediction. That fallback is *not* automatically certified: $P_L + Z_L$ can exceed λ even though there is no remaining suffix. Our implementations and tables distinguish certified early stops from compulsory final execution. Charging an inspection before acceptance also means that $Z_t > \lambda$ precludes any certificate, irrespective of confidence.

F.3.1 When the rule becomes a confidence threshold

At a record prefix $B_t = p$, each future constraint has the form $h_s(p) \le a$, where $s = e^{2r_j}$, $a = \lambda - Z_t + D_j$, and

$$h_s(p) = \frac{(s-1)p(1-p)}{1 + (s-1)p}, \quad \max_{p\in[0,1]} h_s(p) = \frac{\sqrt{s}-1}{\sqrt{s}+1}, \quad p_\pm = \frac{1 - a \pm \sqrt{(1-a)^2 - 4a/(s-1)}}{2}. \tag{28}$$

For a nonvacuous positive constraint, acceptance lies below p_- or above p_+. Intersect these sets over future exits and enforce the observed-past constraint. On $p \ge 1/2$, the future part is a single upper-threshold test; its cutoff can be compiled offline. Retaining only this branch is conservative when a lower branch exists. At nonrecord prefixes, replacing B_t by current p or discarding P_t changes the problem. Our multi-exit controller therefore retains the score record and past net rewards rather than applying a first-exit cutoff at every stage.

F.4 FINITE IMPLEMENTATION AND TASK-LEVEL RETURN PRESERVATION

Deployment uses $S = 64$ logit units, prefix integers $q_0 \in [-512, 512]^K$, and additive updates bounded by 16 units when $\beta = 0.25$. Let $M = 2^{48}$ and $E^-[d]$ be the committed monotone integer lower approximation of $Me^{-d/S}$, with $E^-[0] = M$. Define

$$d(q) = \sum_k E^-[\max_\ell q_\ell - q_k], \qquad \widehat{g}(q) = M/d(q). \tag{29}$$

Rational Taylor bounds verify the table independently; the uniform score error is at most $\epsilon_E = 730/M < 2.594 \cdot 10^{-12}$ on the declared domain. A regret statement transfers to real softmax with $2\epsilon_E$ added to its tolerance. The finite claim itself uses the rational score, not rounded floating-point softmax comparisons.

For a remaining integer radius b, add b to a currently maximal logit and subtract b from the others. Monotonicity of the table gives the exact finite-score endpoint support, attained simultaneously by aligned integer updates. A score difference is compared with rational $a = u/v$ through $Mv(d_2 - d_1) \le ud_1d_2$, using arbitrary-precision integers. No tolerance is introduced to decide acceptance. Prefix normalization and table lookup are real operations; the "one comparison" first-exit implementation refers only to the additional cutoff decision after normalization and offline compilation.

At each inspected exit the general implementation maintains current logits, all available score denominators (or equivalent past-reward state), the minimum denominator and earliest return index, active input indices, and paid check count. It tests every observed-past constraint and each remaining endpoint support. Rejected inputs alone execute the next residual block. This direct general implementation does more work than the compiled first-exit branch; neither is assigned the other's timing. Invalid or out-of-domain arithmetic must fail closed rather than become a certificate. The released replay verifies all visited finite states; this is not a claim that arbitrary operating-system or NaN-producing executions are verified.

F.4.1 A SHARP GUARANTEE FOR THE RETURNED DECISION

Confidence regret and task loss are different quantities. A useful bridge is to preserve the decision made by the entire full-recall network, without requiring every intermediate head to keep the same label. Let a be the label currently recalled, let $v = \max_{k \neq a} \mathrm{softmax}(z_t)_k$, and put $r = \sum_{s>t} \beta_s$. The current prefix need not be a record.

Theorem 13 (Full-recall return preservation)**.** *With earliest-exit tie-breaking, every completion in $\mathcal{A}_t$ has full-recall returned label a if and only if*

$$\frac{e^{2r}v}{1 + (e^{2r} - 1)v} \le B_t. \tag{30}$$

When this holds, stopping preserves $\ell(a, y)$ relative to full recall for every *possible true label y and every loss depending only on the returned decision. It does not establish that $a = y$ or preserve a probability-dependent loss such as cross-entropy.*

The proof, including the necessity witness, is in Appendix G.

The exact finite analogue checks alternate-class boosted endpoints: a harmful endpoint has a label other than a and a denominator strictly smaller than the observed record. Testing all alternate classes handles quantization and the deterministic class tie rule. Conjoining this guard with (27) yields both a charged score-regret certificate and full-recall decision-loss preservation. It does not certify the unknown label or remove the architectural approximation to the trained suffix. Appendix G gives the finite argument, state sufficiency, and exhaustive nonrecord-prefix checks.

Corollary 14 (Whole-policy decision-loss preservation)**.** *Assume early and full execution share the same observed-prefix semantics. A controller that conjoins the scalar and return tests, and otherwise falls back to full recall, returns the full-recall label on every valid input. Its expected decision loss therefore equals full recall under any joint input/label distribution for which the expectation exists, although its scalar regret guarantee remains selective to certified stops.*

F.5 LEARNING THE CONTRACT RATHER THAN SATURATING IT

A bounded correction architecture has a precise capacity restriction. If a persistent feature representation permits arbitrary readouts into $[-B, B]^K$, then any desired amendment $a(h)$ in that cube can be split as $\eta_j(h) = \beta_j a(h)/B$, where $B = \sum_j \beta_j > 0$. Conversely, every admissible sum lies in that cube. This characterizes the available amendments, not whether a particular optimizer reaches them. At a raw hard clamp $\eta = \mathrm{clip}(u, -\beta, \beta)$, saturated coordinates have zero training derivative. An expressive amendment class can therefore coexist with a nearly frozen readout.

We retain the same bounded deployed map but vary how it is learned. Small initialization rescales the update heads using training features only. The smooth alternative uses $\eta = \beta \tanh(u/\beta)$ before deployment quantization, so its derivative is $\mathrm{sech}^2(u/\beta)$ rather than an abrupt zero outside the cap. Straight-through variants use a clipped forward map with surrogate backward derivatives; their forward contract is unchanged. A second construction trains an unconstrained suffix and clips its readouts only for deployment, optionally followed by straight-through fine-tuning. All bounded variants use the same integer radius 16; none gets a looser certificate by being assigned a different β.

Projection is a genuine alternative, not merely an ablation. When hidden states do not depend on the persistent logits, let u_s be the unconstrained readout and set $e_s = \|u_s - \mathrm{clip}(u_s, -\beta_s, \beta_s)\|_\infty$. Through exit j, projection and prefix/update rounding give

$$\|z_j - \widetilde{z}_j\|_\infty \le E_j := \sum_{s=1}^{j} e_s + \frac{j+1}{2S}. \tag{31}$$

An unconstrained top-two logit margin greater than $2E_j$ preserves that head's label; confidence changes by at most $E_j/2$ and cross-entropy by at most $2E_j$. The bound follows from telescoping and the softmax and cross-entropy gradient norms. It is useful when the realized projection residuals are small, but is not substituted for measured accuracy or a full-recall certificate. General feedback from logits into hidden states would require an additional propagation bound. Full proofs and the older capacity and projection development are preserved in the appendices.

F.5.1 REGISTERED OPTIMIZATION COMPARISON

The experiment uses the five frozen convolutional prefixes from the preceding Fashion-MNIST study, seeds 3101–3105, the same split, 128-dimensional hidden representation, optimizer family, and deployment precision. Each suffix has three residual hidden blocks and additive ten-class readouts. The public test results of the preceding study were already known when the new protocol was committed. This is a registered *post-review extension*, not a new untouched confirmation dataset. All seeds, methods, and fixed checkpoints are retained; no run is selected for its test score.

New small-initialization and straight-through fits have fixed 4- and 12-epoch endpoints. Continued hard and unconstrained fits add eight epochs to the original four-epoch checkpoints; projected fine-tuning starts from the frozen unconstrained checkpoint. The new optimizer state in continuation is recorded, so "12 epochs" describes total fitting work, not an uninterrupted optimizer trajectory. Readout rescaling uses no test features. The release contains all 45 fitted checkpoint hashes, training losses, stagewise gradient activity, cap occupancy, calibration, and post-training projection of both four- and twelve-epoch fits. The entire original four-epoch comparison remains in the archived main text.

Table 14 identifies an optimization failure that additional hard-clipped training alone does not repair. Continued hard training reaches 85.416% terminal accuracy, compared with 85.380% in the preceding four-epoch model. Its last two stages are completely saturated and have zero clamp derivatives. Small hard initialization reaches 86.472%; small smooth initialization reaches 86.712%, with final-stage near-cap occupancy 50.331% and mean smooth derivative 0.1985 instead of zero. Saturation here means proximity to the deployed cap; a large raw smooth preactivation is not itself a clipping event. The complete stagewise distinction is retained in Appendix H.

For inference over initialization and minibatch randomness, the unit is a paired seed, not a test row. We report $\bar{d} \pm t_{.975,4} s_d/\sqrt{5}$, conditional on the common data split and test set. Small smooth minus continued hard gives $+1.296$ percentage points, 95% interval $[0.865, 1.727]$; small hard minus continued hard gives $+1.056$ $[0.501, 1.611]$. The smooth-minus-small-hard difference is $+0.240$

Table 14: Optimization under the same $\beta = 0.25$ deployment contract, five paired seeds. Accuracy is in percent; saturation is the percentage of final-stage coordinates above 99% of the cap magnitude before rounding. Every registered endpoint is retained. The unconstrained row is a comparator, not a certified bounded model.

Readout/training	Epochs	Terminal	Recall	Saturation
Hard, continued	12	85.416	85.410	100.0
Small init., hard	4	86.170	85.866	84.1
Small init., hard	12	86.472	86.172	86.1
Small init., smooth	4	86.308	85.980	23.1
Small init., smooth	12	86.712	86.348	50.3
Straight-through	4	85.444	85.420	56.4
Straight-through	12	85.600	85.518	10.3
Projected + straight-through	12	85.576	85.538	37.9
Unbounded, continued	12	86.854	86.706	–
Unbounded then projected	12	86.276	85.894	61.0

Table 15: Executed depths for the general controller: mean counts per 10,000 test inputs across all five seeds, $\lambda/\mathrm{price} = 1/3$. Depths 0–2 are certified; depth 3 is compulsory, uncertified fallback. Network MACs and total charged credits per input are in thousands; full execution uses 225.184 thousand MACs. All inspections, including failures, cost 4,096 credits.

Model	Price	$T = 0$	$T = 1$	$T = 2$	$T = 3$	MACs	Credits
Hard (R31)	0.01	3294.8	79.2	23.0	6603.0	207.40	216.96
Hard (R31)	0.02	4053.2	99.2	33.0	5814.6	203.30	212.22
Hard (R31)	0.04	4906.0	117.8	28.6	4947.6	198.72	206.94
Hard (R31)	0.08	5930.4	126.2	30.8	3912.6	193.26	200.64
Smooth (R32)	0.01	3294.8	211.6	178.6	6315.0	206.66	216.16
Smooth (R32)	0.02	4053.2	221.2	180.8	5544.8	202.60	211.48
Smooth (R32)	0.04	4906.0	238.6	181.0	4674.4	198.02	206.19
Smooth (R32)	0.08	5930.4	255.0	173.0	3641.6	192.55	199.88

$[-0.108, 0.588]$, so the study supports the combined repair but does not isolate a statistically decisive smooth-activation advantage over suitable hard initialization. These unadjusted descriptive intervals are not a multiple-comparison familywise procedure.

The unconstrained 12-epoch comparator reaches 86.854%; its paired difference from the smooth bounded model is 0.142 points, with the smooth-minus-unbounded interval $[-0.521, 0.237]$. An interval containing zero does not establish equivalence or noninferiority. Post-training projection reaches 86.276%, below smooth bounded training by 0.436 points $[0.089, 0.783]$ in this budget, but above the poorly optimized hard baseline. Straight-through training without the small-output repair remains at 85.600%; projected straight-through fine-tuning reaches 85.576%. Thus backward substitution alone is not the observed solution. The paired comparisons make initialization, training budget, and projection effects visible without attributing all accuracy loss to the mathematical contract.

F.6 Execution, sensitivity, and measured resource benefit

We evaluate the original hard $\beta = 0.25$ family and the registered smooth 12-epoch family at all four total suffix prices $p_c \in \{.01, .02, .04, .08\}$. Each of the three transitions costs $p_c/3$. The prefix costs 172,192 MACs, each residual correction costs 17,664, and full execution costs 225,184. Each inspection is assigned 4,096 work credits, valued at $4096p_c/52992$ in the same objective; all failed inspections are charged. This declared work objective is distinct from a worst-case latency bound. The schedules are the compiled first-exit baseline and the complete $(0, 1, 2)$ schedule, with compulsory final fallback. A third mode adds the full-recall return guard to the complete schedule.

F.6.1 Actual intermediate-depth decisions

Table 15 shows that the general controller executes a real multi-depth policy. At price .04 and tolerance ratio $\lambda/p_c = 1/3$, the hard family stops 117.8 inputs at exit 1 and 28.6 at exit 2 per 10,000;

the smooth family stops 238.6 and 181.0. Initial-exit counts agree across families because the prefixes and deployment bounds are identical. This agreement is predicted by the information contract, not a fitted equality of policies. Among inputs that continue, the trained suffix changes the observed state and therefore later certified depths.

Every accepted early stop satisfies the registered charged score tolerance. At the default ratio, both families also preserve actual full-recall labels, and the additional return guard changes no actions. Nevertheless, terminal and recall accuracy are different: the smooth family's terminal accuracy is 86.712%, while full recall and certified stopping return at 86.348%. For the original hard family the corresponding recall value is 85.446%. The task claim is preservation relative to the specified recall baseline, not accuracy dominance over the raw terminal head.

A further read-only matched comparison replaces the structural supports by trivial score headroom, using $g_j \leq 1$ with the same histories, prices, schedule, and charges. The trivial future bound is $[1 - B_t - \min_{j>t} D_j]_+$. At every inspected prefix the structural bound is no larger, and the actual structural stopping depth is no later on any of the 40 matched seed–model–price conditions. Beyond this baseline, the hard family certifies an additional 930.6–1,381.4 inputs per 10,000 across the four prices; the smooth family adds 1,060.8–1,514.4. At price .04, structural stopping is strictly earlier on 1,256.2 and 1,529.4 inputs, respectively. Thus the information in bounded amendments changes decisions beyond generic remaining headroom even after every inspection is charged. This additional frozen-trace comparison involves no fitting or new claim of preregistration; its full results and rowwise depth-dominance checks are retained in `R32_HEADROOM.json`.

The direct multi-exit controller reduces network work beyond the first-exit baseline, but extra inspections can more than offset this difference. For the hard family at price .04, average network work falls from 199,186.12 to 198,719.44 MACs, while charged work rises from 203,282.12 to 206,940.20 credits. Paired intervals for both differences are provided with all four prices. Both remain below full execution under the declared credit model, but first-exit compilation is cheaper. This does not contradict Theorem 12: optimality is conditional on a fixed schedule and information state, not a theorem that additional inspections are free or that every schedule improves expected resource cost.

We replay the actual residual blocks with an active-input mask for all five seeds, both model families, four prices, and both schedules: 80 executions. All executed depths, returned exits, and returned labels agree with the frozen-trace policies. Of 20,913,040 visited quantized logit coordinates, three differ from their original full-batch traces because changing a masked floating-point batch can change a rounded neural output. Those differences change no recorded decisions, and the finite increment contract remains valid for the actual outputs. We report this distinction rather than calling the neural tensors bit-identical. The certificate is computed on the actual observed state and does not require the floating kernel to reproduce a previous trace coordinate for coordinate.

F.6.2 Utility sensitivity and nonrecord prefixes

The registered grid crosses all four prices with ratios $\lambda/p_c \in \{0, 1/6, 1/3, 1/2, 1\}$ and all three policy modes. Its 600 seed–model–price–ratio–mode conditions reuse the same test rows; the six million recorded policy/input cases are not independent experimental replications. Appendix M gives every condition, including all failures to certify. At ratio zero the positive inspection fee prevents acceptance and every input falls back. This is the correct consequence of the chosen objective, not a numerical failure.

At price .04, hard-family scalar coverage rises from 4,600.6 to 5,052.4, 5,411.8, and 7,350.4 per 10,000 as the ratio increases from $1/6$ to $1/3$, $1/2$, and 1. Nonrecord accepted prefixes appear at the last two ratios: 85.4 and 1,343.6. The controller therefore exercises the recalled record and observed-past term beyond synthetic tests. At ratio 1 the scalar policy disagrees with the actual full-recall label on 7.0 inputs and changes 3.6 correct full-recall decisions into errors. Adding the return guard reduces coverage to 7,280.8 and eliminates all such disagreements. For the smooth family, coverage changes from 7,433.2 to 7,374.0, eliminating 13.4 label disagreements and 9.0 harmful changes. The guard can reject more prefixes than exhibit a change on the trained trajectory because it covers all admissible completions, not only that one trajectory.

Certified coverage must also be separated from fallback behavior. At price .04 and ratio $1/3$, the hard-family general controller has 4,947.6 compulsory final fallbacks, of which 4,292.6 exceed the

Table 16: End-to-end idle CPU time (milliseconds per fixed workload), including image prefix, normalization, policy, masking, suffix and return. Medians of nine randomized paired rounds; batch 1 processes 128 images, batch 32 processes 992, and batch 1,000 processes 1,000. Raw pairs, competing-worker results, training-only admission, and setup-inclusive curves are retained. Calib. denotes the full temperature-calibrated path, not a certificate.

CPU	Batch	Full	First	Multi	Ridge	Calib.
X86	1	15.77	11.96	33.47	37.78	17.79
X86	32	10.91	10.33	22.50	18.39	11.64
X86	1000	5.43	4.80	7.02	5.28	5.72
ARM	1	70.80	53.03	105.79	101.35	74.88
ARM	32	66.94	65.04	89.96	76.74	68.37
ARM	1000	19.97	18.64	22.10	19.89	20.52

charged tolerance. These are not accepted certificates or hidden violations of the theorem. The released records include every fallback and its regret; an application requiring a guarantee for every input must handle this uncertified outcome explicitly. A controller-fit threshold that accepts more inputs can incur observed score violations even when its task accuracy is similar. The preceding fixed-model study retained 362.4–933.8 such heuristic violations per 10,000 across the four prices; at .08, 102 heuristic-accepted inputs changed a correct recall result into an error. At the other prices, zero observed task harm did not imply a score-regret guarantee.

F.6.3 End-to-end runtime and setup-inclusive accounting

We measure complete image inference, not timing over saved logits. The same frozen seed-3101 hard candidate and policy definitions execute on an AMD EPYC 9V45 x86-64 runner and an ARM64 runner whose exposed processor identifier is `aarch64`; the latter observation does not identify a particular commercial CPU model. Both use four available CPUs, one Torch/BLAS thread, and the recorded software environment. Training-only pilot comparisons freeze admission before this run's test access. Each fixed workload then has nine randomized paired rounds under idle conditions and under one competing CPU worker. Model seed, processor architecture, workload size, and execution repetition are separate axes of evidence.

Table 16 includes the learned ridge continuation controller and the temperature-calibrated full path with their actual feature construction, normalization, dispatch, masks, suffix, and returned prediction. On the 1,000-image idle workload, first-exit compilation takes 4.799 ms against 5.432 ms for full recall on x86, and 18.638 ms against 19.970 ms on ARM. The direct multi-exit path takes 7.017 and 22.103 ms: fewer network MACs do not compensate for its Python history-aware checks. The ridge path takes 5.279 and 19.890 ms, much closer to full execution than its uncharged-MAC comparison alone suggests. All methods remain in the table, including those declined by the training pilot. The ARM ridge path was not admitted; its small observed test-time advantage is not retroactively used for selection.

For these fixed workloads, first-exit speedups range from about 1.06 to 1.32 on x86 and 1.03 to 1.34 on ARM across idle batch configurations. Under one competing worker, its 1,000-image times are 5.021 and 18.781 ms, against full-recall times 5.734 and 20.188 ms. Paired timing intervals quantify repeat-execution variability for this workload only, not variation over new models or CPU fleets. The full calibrated path changes the confidence objective and has no automatic inheritance of the original regret certificate. The measured image subset has 87.3% full-recall accuracy at batch 1,000; it is not substituted for the complete 10,000-row accuracy comparison.

Initialization is a fixed cost, not an arbitrary per-input subsidy. If setup cost is S_0, full and gated workload times are F and G, then after k repetitions the measured total-speed ratio is

$$\frac{kF}{S_0 + kG}, \qquad k_{\min} = \left\lfloor \frac{S_0}{F - G} \right\rfloor + 1 \quad \text{when } F > G; \quad \text{no break-even when } F \le G. \tag{32}$$

The strict inequality is intentional: equal total times are not a speedup. All 135 preceding timing pairs are reanalyzed without assuming 100,000 future inputs. For the three retained large-batch processes, median paired gains give break-even at 15, 13, and 13 workloads of 1,000 inputs; two pairs

Table 17: Paired full-minus-policy savings in milliseconds, 1,000-image workloads. Brackets give unadjusted 95% t intervals over nine randomized rounds, not over new images or processors. $k_{\min}$ uses the median paired first-gate gain and its incremental table/cutoff setup; it is an empirical horizon, not a worst-case bound.

CPU	Load	First: saved ms [CI]	Ridge: saved ms [CI]	$k_{\min}$
X86	Idle	0.631 [0.586,0.676]	0.101 [0.039,0.163]	4
X86	Worker	0.364 [-0.106,0.835]	-0.592 [-1.525,0.341]	4
ARM	Idle	1.300 [1.232,1.368]	0.097 [0.037,0.157]	3
ARM	Worker	1.420 [1.305,1.534]	0.101 [0.003,0.198]	2

in the second process never break even. Small-batch median gains are nonpositive. New x86/ARM curves use their measured incremental table-check/cutoff setup (2.010 and 2.776 ms); common framework/model loading and offline policy training are not charged as incremental gate setup. These conditional curves are not cold-process startup benchmarks and do not borrow the preceding process's initialization. A training-frozen tariff is empirical, not a worst-case execution-time certificate. No GPU, energy, or universal latency benefit is inferred.

Table 17 reports means and intervals for paired timing differences alongside the median times. On the x86 competing-worker workload, a positive median first-gate gain coexists with a mean-difference interval that crosses zero; the complete nine rounds, including slow observations, are retained. Its mean-versus-median difference is evidence against replacing an empirical tariff by a deterministic latency guarantee. The idle paired gains imply incremental-setup break-even after four 1,000-image workloads on x86 and three on ARM. An application serving fewer requests cannot infer a lifecycle benefit merely from a steady-state ratio exceeding one.

F.6.4 Matched suffix verification and retained evidence

The modern-verifier comparison now uses the same Fashion-MNIST seed-3101 hard bounded suffix as the main frozen-model comparison, not the preceding MNIST candidate. A label-free uniform sample of 128 test indices is frozen with seed 320926 before bounding. Prefix logits are fixed, and all methods bound conditional hidden-state cubes of radii 0, .01, and .1. Structural support uses the same prefix and suffix contract but ignores hidden-state and weight information. IBP, CROWN, 20-iteration optimized CROWN, and a four-cell IBP split use the authors' pinned auto_LiRPA implementation; the split is a diagnostic, not a complete branch-and-bound evaluation.

At price .04 and tolerance .04/3, the *uncharged diagnostic* structural test accepts 65/128 prefixes. IBP accepts 97, 97, and 69 as the radius increases; CROWN and optimized CROWN accept 97 at every radius. The four-way split matches IBP's acceptance counts. Weight information materially tightens the structural contract, especially for a saturated hard suffix. At radius .1, IBP, CROWN, and optimized CROWN take 0.0056, 0.3423, and 44.9151 seconds for the 128 prefixes, versus 0.0000584 seconds for the structural bound calculation; graph construction adds 0.0270 seconds. These are bounding costs, not end-to-end deployed speedups. Standalone and structurally intersected bounds, all arrays, and every radius are reported in Appendix J. Stagewise quantization allowances are included, but float64 relaxations are not outward-rounded implementation proofs and are not silently used to accept deployment decisions.

The release independently checks 38,400 prefix states with Python rational arithmetic and exhaustively checks 43,740 small nonrecord return completions. The preceding 800,000 uncharged and 800,000 charged scalar replays, compiler checks, and complete historical derivations remain available. The older MNIST example, weak high-dimensional grid bounds, CIFAR-100 geometry equal to trivial headroom, and SVHN transport coverage below its nominal target are preserved, not removed to improve the narrative. The new evidence establishes a trainable convolutional construction, genuine multi-depth execution, and measured CPU implementations under an explicit contract. It does not establish large-transformer accuracy, arbitrary neural reachability, or transport to new deployment distributions.

In the frozen .04-price policy comparison, the matched ridge terminal controller's accuracy difference from the original certificate is $+0.048$ percentage points, with paired-seed interval $[-0.247, 0.343]$; its MAC difference is -234.2 $[-900.7, 432.3]$. These intervals do not establish a task–compute

frontier advantage. All prices and calibrated-policy differences are retained under their original baseline definitions, not substituted for the new charged work model.

F.7 Conclusion

An output-increment contract turns the unknown suffix into a sharp completion problem. The resulting stopping test is information-optimal, recall-aware, and charge-aware; its finite implementation distinguishes certified early termination from final fallback. A second sharp criterion connects the contract to preservation of the full-recall decision for any true label. The five-seed optimization extension demonstrates that severe hard-clamp saturation is not an unavoidable cost of the same deployment bound. Actual multi-depth execution and matched hardware measurements then expose when extra inspections help network work and when their implementation cost outweighs that saving. This combination makes optimal certifiable depth an executable claim with explicit information, task, and resource semantics.

G General history-dependent and return-preserving certificates

This appendix gives the implementation-level arguments for the new general controller. The earlier complete real/finite proofs, capacity result, projection analysis, compiler construction, and all counterexamples follow unchanged in the historical appendices.

G.1 Sufficient state and executable recurrence

Define $H_t = \max_{j \le t}(B_j - C_j)$. The sufficient scalar history for the past obstruction is (B_t, H_t, C_t): $P_t = H_t - B_t + C_t$. On observing score g_{t+1}, update $B_{t+1} = \max(B_t, g_{t+1})$ and $H_{t+1} = \max(H_t, B_{t+1} - C_{t+1})$. The earliest maximizing exit is updated only on a strict new score record. Thus the past obstruction can be maintained incrementally without replaying all past rewards. The released reference implementation deliberately also retains the full short history of denominators and quantized logits for auditing; it evaluates the same inequalities directly. It is not claimed to be an optimized native kernel.

For an inspection schedule $\mathcal{I} \subseteq \{0, \dots, L-1\}$, the controller has the following precise semantics. Execute the prefix and initialize B, H, the recalled exit, and paid inspection cost $Z = 0$. At each $t \in \mathcal{I}$, add the declared inspection charge to Z *before* testing. Compute $P = H - B + C_t$ and every future support from the current logits. Stop and return the stored record if $\max(P, U) + Z \le \lambda$, optionally conjoined with the return-preservation guard. Otherwise execute the next correction for the still-active inputs, update history, and continue. At L, return the stored record with a distinct fallback flag. No inspection failure is erased from Z, and no already-executed stage is refunded when the returned exit is earlier.

Proposition 15 (Inspection-schedule qualification)**.** *Fix the future inspection schedule and the charges accrued at every inspected prefix. Before the first successful scalar test, no sound stopping policy with the same information and accrued charges can stop at one of those inspected prefixes. Adding inspections to the schedule does not imply a monotone reduction of charged expected cost or wall-clock time.*

Proof. At each failed inspected prefix, the aligned completion from Theorem 1 is an admissible witness with regret above the tolerance. A policy accepting there is unsound for that prefix. This pointwise fact yields earliest certifiable execution for the fixed schedule. Changing the schedule changes Z and the computation itself. For example, an additional failed inspection adds a positive charge while changing neither the eventual return nor its execution depth; this strictly raises cost. Hence the first statement implies no schedule-level dominance. □

G.2 Exact integer comparisons and domain

For $\widehat{g} = M/d$, the comparison $M/d_1 - M/d_2 \le u/v$ is equivalent to $Mv(d_2 - d_1) \le u d_1 d_2$, with $v, d_1, d_2 > 0$. Python arbitrary-precision integers implement this identity without a floating acceptance tolerance. For an observed history $q_{0:t}$, let $d_j = d(q_j)$ and $b_j = \min_{i \le j} d_i$. The past tests

are, for every $j \le t$,

$$M/b_j - M/b_t \le \lambda - Z_t - (C_t - C_j).$$

For each future j, compute its exact endpoint denominator d_j^* by raising a current maximizing class by $16(j-t)$ and lowering all other classes by that amount. The future tests are

$$M/d_j^* - M/b_t \le \lambda - Z_t + (C_j - C_t).$$

The explicit $Z_t \le \lambda$ test covers the zero part of the positive maximum. These are exactly the inequalities used by `check_prefix`; accepted policies are also checked against the actual full observed trajectory's charged regret as a separate diagnostic.

The integer endpoint bound follows directly from monotonicity of E^-. For a fixed winning class, increasing its gap over a competitor cannot increase the denominator. The boosted endpoint dominates every reachable gap for that class. Choosing a currently maximal class gives the least possible endpoint denominator among classes. Applying the same signed update at successive stages achieves these denominators on one coupled path. Strict monotonicity of the approximation is unnecessary; plateaus can create additional ties but cannot reverse the support inequality. The committed score table's rational construction and error envelope are independently replayed by the preceding validator. Integer overflow cannot occur in the arbitrary-precision comparisons; vectorized neural operations are separately restricted to the declared finite conversion domain.

G.3 Finite full-recall returned-label theorem

Let a denote the currently recalled label and b the minimum observed denominator. For each $k \ne a$, form the endpoint $q^{(k)}$ by adding the entire remaining integer radius to coordinate k and subtracting it from all others. The return guard rejects if any endpoint satisfies

$$\operatorname{argmax} q^{(k)} \ne a \quad \text{and} \quad d(q^{(k)}) < b.$$

The strict denominator comparison enforces earliest-exit recall ties, while `argmax`'s first-class rule enforces class ties. At $t = L$ there is no unseen suffix and the returned decision is already fixed; this does not remove the independent scalar past-regret obstruction.

Proposition 16 (Finite return guard)**.** *For additive coordinatewise integer bounds and the declared score table, the guard above accepts exactly those prefixes whose returned label is preserved by every admissible integer completion.*

Proof. A harmful endpoint supplies a coupled witness by distributing its aligned increments across remaining stages. If some earlier exit on that witness already has a still-higher score with a label other than a, the return is harmful already. No a-labelled future exit can form a new score record on the aligned path toward k: the gaps of a against k decrease and its gaps against all other coordinates are unchanged, so its confidence while maximal cannot exceed its present value, which is at most the observed record. Hence the endpoint witness truly changes recall, not merely an isolated future head.

Conversely, any changed full-recall return has some future winning class $k \ne a$ with score strictly above the old record. At the corresponding depth, boost that class by the entire remaining radius and lower all its competitors. Every gap from k weakly increases relative to the harmful completion. The endpoint is therefore still won by a class other than a (with the same deterministic tie convention), and its denominator is no larger than the harmful completion's denominator. It is strictly smaller than b, so the endpoint test detects it. This proves necessity and sufficiency, including nonrecord prefixes. □

For the real-softmax version, the largest competing probability is $F_r(v) = e^{2r}v/[1 + (e^{2r} - 1)v]$, giving Theorem 3. Decision-loss preservation is pointwise in y: $\ell(a, y) = \ell(a_{\text{full}}, y)$ because the two decisions are equal. No label distribution, calibration, or relationship between max-softmax confidence and correctness is required. Conversely, confidence-regret control alone need not preserve the returned decision, as shown by the registered high-tolerance conditions. This distinction is also why a cross-entropy claim cannot be inferred from the return guard.

G.4 Projection and representation details

Assume the hidden trajectory is the same before and after output projection. At each stage the raw and clipped updates differ by at most e_s in infinity norm; prefix rounding and each of j update roundings cost at most $1/(2S)$. Summing gives (12). A perturbation of size E_j moves a top-two logit gap by at most $2E_j$, proving the head-label statement. For a softmax component p_k, the gradient has ℓ_1 norm $2p_k(1-p_k) \le 1/2$, so each component and their maximum are $1/2$-Lipschitz for infinity-norm logit perturbations. The gradient of $-\log p_y$ is $p - e_y$ and has ℓ_1 norm at most 2. This proves the confidence and cross-entropy bounds. Neither bound controls which exit wins recall without an additional score-margin argument or the return-preservation test. The preceding projection appendix retains the more detailed realized-margin and calibration diagnostics.

G.5 Validation scope

The new independent rational replay samples 128 distinct rows per seed, four prices, five tolerance ratios and three inspected depths: 38,400 prefix comparisons. It uses Python `Fraction` expressions for P, U, and Z, separately from the cross-product implementation. The return test additionally enumerates all 27 choices at each of two three-class increment stages for 60 generated histories, including prior nonrecord states: $60 \cdot 27^2 = 43,740$ completions. These checks test implementation correspondence; the universal guarantee comes from the proofs, not from interpreting sampled histories as exhaustive neural verification. Every reported main-grid condition is retained in the row archive, including compulsory final outcomes.

G.6 Real returned-label witness and whole-policy implication

Proof. The left side bounds every competing class probability at every future exit. If it is at most B_t, no competing label can beat the earlier record; equality loses the earliest-exit tie. Conversely, align all increments toward a class attaining v. If its final probability exceeds B_t, it becomes a new record. Every other class probability decreases from its current value, which is at most B_t, so no competing old class can mask the witness. That completion changes the returned label. Equality of decision losses follows from equality of the returned decision for every y. □

The proof partitions inputs into certified stops, covered by Theorem 3, and full-execution fallbacks. The shared-prefix qualification matters for an implementation: floating batch-shape changes are separately audited below rather than assumed bit-invariant. This corollary converts the local return test into a complete-policy task statement without claiming that confidence is calibrated or full recall itself is optimal for the task.

H Optimization protocol, paired uncertainty, and diagnostic definitions

The new registered protocol and training script were already committed before the five training jobs executed. The publication continues those exact jobs; it does not retrospectively register a newly chosen seed or checkpoint. The inherited per-seed split consists of 50,000 fitting, 5,000 controller-fit, and 5,000 tuning training examples. The 10,000 public test examples are never used for gradient updates. Their prior use in preceding revisions is explicitly acknowledged; this study is not claimed to be a fresh holdout evaluation.

The prefix is frozen. Each suffix uses three residual hidden blocks with 128-dimensional state and ten-dimensional additive output corrections. Training minimizes mean cross-entropy across the three correction heads, using Adam at learning rate .001, minibatches of 256, and the registered per-seed data permutations. New initializations use the same seed offset across methods. Continued fits skip the four preceding permutations before starting their additional eight epochs, and reset Adam because optimizer state was not retained in the preceding checkpoints. This makes the reset visible rather than silently describing continuation as uninterrupted training. Fixed four- and twelve-epoch endpoints are reported; there is no validation- or test-score checkpoint selection.

Small initialization rescales each output head by $\min\{1, \beta/(4\max(q_{.95}(|u|), 10^{-12}))\}$, with its .95 quantile computed from fitting features. The hidden trajectory is recomputed before rescaling the

next readout. The hard and smooth small-initialization methods therefore share the same initial head scales. For example, seed 3101 uses approximately .01514, .01861, and .01434; all five complete scale vectors are retained in `TRAINING.json`. This is an initialization intervention, not a post-test alteration of the deployment cap.

Clipping and derivative statistics. The registered diagnostic distinguishes raw output magnitude above β, deployed real output magnitude above $.99\beta$ before quantization, mean derivative used in training, exact zero-derivative frequency, and raw-magnitude quantiles. Hard readouts use derivative $\mathbf{1}\{|u| < \beta\}$; smooth readouts use $1 - \tanh^2(u/\beta)$; straight-through readouts use the surrogate derivative one. Thus a straight-through derivative of one is not evidence that the mathematical hard clamp is differentiable there. Projecting an unconstrained model at inference changes its forward output but does not retroactively change its unconstrained training derivative. Per-epoch mean readout gradient norms are separate from these local derivative statistics and are fully retained.

At epoch 12, the continued hard model's mean near-cap proportions are 99.9548%, 100%, and 100%; corresponding zero-derivative proportions are 99.9544%, 100%, and 100%. For small hard initialization the near-cap proportions are 86.4076%, 84.5992%, and 86.0728%, with mean derivatives .137364, .155604, and .140824. For small smooth initialization they are 63.6616%, 54.7060%, and 50.3310%, with mean derivatives .139828, .178832, and .198499. Smooth finite arithmetic still has nonzero exact-saturation frequency, recorded as 1.0590%, .1634%, and .1054%; "smooth" is not represented as a guarantee against all finite-precision saturation. The joint intervention, training losses, and paired accuracy differences are stronger evidence of a repaired optimization path than an expressivity theorem alone.

Seed-level uncertainty. For each comparison, form five within-seed differences and use Student's t interval with four degrees of freedom. The five smooth-minus-continued-hard terminal-accuracy differences, in percentage points, are $1.23, 1.54, .90, 1.06, 1.75$. Their mean is 1.296 and sample standard deviation .347318. Small-hard-minus-continued-hard differences are $.53, 1.48, .92, .79, 1.56$. Every comparison's raw vector, sample deviation, and interval is in `R32_ANALYSES.json`. The test images are shared across seeds and methods. We do not treat the resulting 50,000 seed/image predictions as independent Bernoulli trials or infer new-dataset uncertainty. The six intervals are descriptive and unadjusted. In particular, overlap with zero for bounded-versus-unbounded accuracy does not prove equivalence.

The mean final-head and full-recall accuracies are both reported for every method in Table 1. Per-head negative log likelihood and expected calibration error are retained in every seed's `RESULTS.json`; neither is substituted for accuracy. All fixed readouts, training logs, projection results, and tests from the preceding revision remain unchanged in the adjacent historical package.

I Execution and systems evidence

Actual masked execution. The 80 matched executions start from frozen observed prefix features and execute the residual layers for active inputs only. They are not full image-pipeline latency measurements; those are reported separately. Every executed depth, earliest returned exit, and returned class is compared with its corresponding full-trace policy. The released execution record includes per-run coordinate counts and discrepancies. Three quantized coordinates differ among 20,913,040 checked coordinates under changed batch shapes; all 80 sets of decisions agree. No discrepancy is hidden by overwriting the expected trace. Finite-domain checks verify that the actually generated updates satisfy the same 16-unit radius.

Declared resource objective. The exact policy grid charges $p_c/3$ for each executed correction and $4096p_c/52992$ for every attempted check. This is a work-credit objective, not a claim that a Python check performs exactly 4,096 CPU instructions or has a corresponding worst-case nanosecond cost. The timed general controller is a transparent reference implementation whose overhead can exceed its network savings. The fixed-schedule optimality theorem remains pointwise and information-relative under the declared credit objective. Raw timing comparisons are empirical statements about different costs.

End-to-end policies. The CPU paths start with image tensors, execute the convolutional prefix, quantize and normalize scores, evaluate the policy, construct active masks, execute the required suffix and return a label. The first path uses the compiled high-confidence cutoff. The multi path uses the complete charged history-dependent gate. The learned ridge path computes the frozen prefix features, standardization, intercept-augmented regression, and threshold; its nonstopped inputs return the terminal head, matching its registered baseline semantics. The full path uses the maximum finite confidence across all heads, while the calibrated full path uses the frozen per-head temperatures. These distinct return objectives are labeled, not pooled as identical algorithms. The ridge and calibrated parameters were frozen in the preceding policy experiment; neither is refitted to the timed test subset.

Hardware and pairing. The x86 runner exposes an AMD EPYC 9V45 processor and four logical CPUs; the ARM runner exposes four CPUs and the architecture string `aarch64`, not an exact chip model. Both use Ubuntu 24.04 runner images, Python 3.13.15, Torch 2.10.0+cpu, and one numerical thread. The training-only pilot uses three randomized rounds on reserved controller-fit images and freezes a descriptive admission decision before test access in that run. The same fixed test workloads are then measured in nine paired rounds at batch sizes 1, 32, and 1,000, under idle and one-competing-worker conditions. The competing worker executes an integer recurrence until signaled to stop; it is not a broad production-load trace. All pair orders, times, admission decisions, labels and software identifiers are retained.

Setup boundary and break-even. The earlier three-process timing experiment supplies its recorded setup envelopes and 135 paired workloads. Equation (11) is applied to every pair and to the median paired gain; a nonpositive gain is reported as having no break-even, not removed. The new architecture study measures incremental verification/loading of the committed score table and cutoff compilation after common model initialization. These incremental costs are 2,009,870 ns on x86 and 2,775,613 ns on ARM. They do not include common Python/Torch startup, loading the neural checkpoints, offline model training, or the pilot experiment. The new total curves are conditional on that explicitly initialized common system. Applying the same setup charge to another policy in the diagnostic JSON is an equal-budget accounting comparison, not a measurement of that policy's own cold-start preparation. For a different service boundary, add its measured common or policy-specific setup rather than interpreting these values as complete cold-start latency.

All new timing differences are computed within randomized rounds. `R32_HARDWARE_ANALYSES.json` supplies 95% descriptive t intervals with eight degrees of freedom, plus setup-inclusive speed ratios at $k = 1, 2, 4, 8, 16, 32, 64, 100$ workload repetitions. Repeated rounds on the same images measure execution variation; they do not provide independent accuracy observations or hardware-population confidence intervals. The shared-prefix and raw-return definitions are maintained in every timed path, so learned-controller feature construction is not omitted from its cost while the certified controller is charged for normalization.

J SAME-CANDIDATE MODERN VERIFICATION

All methods use the same frozen Fashion-MNIST hard-clipped suffix, seed 3101, and the same 128 uniformly sampled test indices. The sampling generator is NumPy seed 320926, without replacement and without inspecting labels. The sample, model-freeze hash, script hash, method list and radii are written before any bound is computed. This ordering prevents within-run row selection, but does not convert the already-known public benchmark into a new holdout.

The suffix graph maps a 128-dimensional hidden state and the fixed prefix logits to all three future logit vectors. Conditional cubes perturb the hidden state while holding prefix logits fixed. These are conditional representation domains, not input-space adversarial balls and not claims that every hidden vector is reachable from an image. Structural support is valid on the same domains but does not use their extra restrictions. Comparing valid bounds on a common property and domain therefore does not mean the methods use equal information.

The implementation is the authors' `auto_LiRPA`, pinned at commit `5a098e8f9fb5786a428a024981d833d303921f2d`, version 0.7.2. CROWN-Optimized uses 20 iterations. IBP-split4 bisects the first two hidden coordinates and takes the maximum support

Table 18: Matched Fashion-MNIST diagnostic: 128 label-free uniformly sampled fixed prefixes, hard $\beta = 0.25$ seed-3101 suffix, three conditional hidden-box radii. Structural acceptance is 65/128 at every radius. Standalone and intersected (Int.) acceptances are uncharged diagnostics, not charged deployment decisions. ΔQ is mean reduction of the structural cap. Times exclude the separate 0.027 s graph construction. All floating bounds include stagewise rounding allowances; they are not outward-rounded machine proofs.

Radius	Method	Standalone	Int.	ΔQ	Seconds
0	IBP	97	97	0.07310	0.00612
0	CROWN	97	97	0.07310	0.11415
0	CROWN-Optimized	97	97	0.07310	4.42484
0	IBP-split4	97	97	0.07310	0.01360
0.01	IBP	97	97	0.06335	0.00485
0.01	CROWN	97	97	0.07310	0.24901
0.01	CROWN-Optimized	97	97	0.07310	23.16868
0.01	IBP-split4	97	97	0.06356	0.01331
0.1	IBP	69	69	0.02438	0.00561
0.1	CROWN	97	97	0.07310	0.34234
0.1	CROWN-Optimized	97	97	0.07310	44.91505
0.1	IBP-split4	69	69	0.02452	0.01302

over the four covering cells; this fixed split is not an optimized or complete branch-and-bound run. The structural bound uses the deployed integer table; neural relaxations use float64 weights/operations and enlarge stage-j logit intervals by $j/128$ for sequential update rounding. Every saved actual integer trace is checked against the resulting intervals. Floating computation of the relaxations is not outward rounded; these diagnostic results are not substituted for the exact deployment certificate.

The first verifier job stopped before bounding because this pinned library requires Python approximately 3.11, whereas the common execution workflow used 3.13. A separate recovery changed only the interpreter to Python 3.11.16, retaining the exact authenticated script, model, sample and domains. It completed all twelve method/radius conditions. The failure and recovery remain in the provenance record. The successful run used Torch 2.10.0+cpu; peak process resident memory was 2,651,664 KiB. Each nonsplit returned bound array pair occupied 61,440 bytes; split arrays occupied 245,760 bytes. Array storage is not represented as peak solver memory.

The table distinguishes standalone relaxation acceptance from intersection with the structural cap. Acceptance here is the uncharged scalar diagnostic at $p_c = .04$, $\lambda = .04/3$, not the deployed charged gate. In these observations standalone and intersected counts agree; the arrays retain all differences in the underlying caps. Radius zero can be nearly exact because the state is fixed, and CROWN remains especially tight on this almost saturated hard suffix. These results need not transfer to the new smoothly trained model or an arbitrary large neural graph.

K MATCHED TRIVIAL-HEADROOM AUDIT

The additional read-only script `r32_headroom.py` evaluates the complete inspection schedule with future support replaced by one. It uses the same observed-past constraints, paid checks, prices, seeds, and two model families at tolerance ratio $1/3$. At every visited trivial-headroom acceptance it independently asserts structural acceptance, and it verifies the rowwise inequality $T_{\text{structural}} \le T_{\text{headroom}}$. All 40 conditions pass. Both the increment-support and headroom policies retain their own failed-check charges; no future information is used to select an inspection. This analysis was added after the five registered fits and is not claimed to be an additional preregistration. Its complete stop histograms, counts, and charged work are retained in `R32_HEADROOM.json`.

L POLICY UNCERTAINTY AND IMPLEMENTATION ROLES

The two implementations serve different deployment roles. The history-aware reference checks the full theorem and exposes nonrecord behavior; the compiled first-exit path targets low controller overhead. A resource manager can select the latter using a training-only workload pilot while

retaining the former as an auditable policy. That choice must be frozen before test-time performance is inspected.

The read-only `r32_uncertainty.py` checks the five frozen controller hashes and computes paired seed intervals for the matched ridge terminal, cost-selected ridge terminal, and matched calibrated-recall policies at all four prices. At price .04 the matched ridge accuracy differences, in percentage points, are $.37, .01, -.07, -.25, .18$, giving mean $.048$ and interval $[-.246670, .342670]$. The paired network-MAC differences have mean -234.22464 and interval $[-900.745086, 432.295806]$. The original certificate comparison was uncharged; these historical work values are not substituted for the R32 charged controller. The twelve complete policy/price conditions and calibrated full-recall comparison are stored in `R32_POLICY_UNCERTAINTY.json`.

M Complete utility sensitivity

Counts below are means per 10,000 inputs across all five seeds. S is scalar certified coverage, SR is coverage after adding the full-recall return guard, NR is scalar nonrecord acceptance, D is scalar returned-label disagreement with actual full recall, and F is the number of compulsory scalar fallbacks whose charged regret exceeds the tolerance. All guarded returns agree with full recall. Repeated rows and prices reuse inputs; they are not independent replications.

Model	Price	λ/p	S	SR	NR	D	F
Hard	0.01	0	0.0	0.0	0.0	0.0	10000.0
Hard	0.01	1/6	3032.2	3032.2	0.0	0.0	6967.8
Hard	0.01	1/3	3397.0	3397.0	0.0	0.0	5463.4
Hard	0.01	1/2	3664.6	3664.6	49.2	0.0	5122.0
Hard	0.01	1	5557.4	5542.6	1440.2	1.0	1357.4
Hard	0.02	0	0.0	0.0	0.0	0.0	10000.0
Hard	0.02	1/6	3779.6	3779.6	0.0	0.0	6220.4
Hard	0.02	1/3	4185.4	4185.4	0.0	0.0	4911.4
Hard	0.02	1/2	4487.4	4487.4	61.6	0.0	4546.4
Hard	0.02	1	6374.0	6339.6	1399.0	2.8	1084.0
Hard	0.04	0	0.0	0.0	0.0	0.0	10000.0
Hard	0.04	1/6	4600.6	4600.6	0.0	0.0	5399.4
Hard	0.04	1/3	5052.4	5052.4	0.0	0.0	4292.6
Hard	0.04	1/2	5411.8	5411.8	85.4	0.0	3873.6
Hard	0.04	1	7350.4	7280.8	1343.6	7.0	783.4
Hard	0.08	0	0.0	0.0	0.0	0.0	10000.0
Hard	0.08	1/6	5558.6	5558.6	0.0	0.0	4441.4
Hard	0.08	1/3	6087.4	6087.4	0.0	0.0	3507.8
Hard	0.08	1/2	6618.8	6618.8	155.2	0.0	2938.2
Hard	0.08	1	8736.8	8565.2	1288.4	20.6	355.0
Smooth	0.01	0	0.0	0.0	0.0	0.0	10000.0
Smooth	0.01	1/6	3113.4	3113.4	0.0	0.0	6886.6
Smooth	0.01	1/3	3685.0	3685.0	0.0	0.0	2488.4
Smooth	0.01	1/2	4025.4	4025.4	34.4	0.0	2244.0
Smooth	0.01	1	5424.0	5414.0	795.4	1.2	630.6
Smooth	0.02	0	0.0	0.0	0.0	0.0	10000.0
Smooth	0.02	1/6	3857.6	3857.6	0.0	0.0	6142.4
Smooth	0.02	1/3	4455.2	4455.2	0.0	0.0	2411.0
Smooth	0.02	1/2	4838.4	4838.4	39.2	0.0	2105.8
Smooth	0.02	1	6330.4	6306.0	834.8	3.4	511.4
Smooth	0.04	0	0.0	0.0	0.0	0.0	10000.0
Smooth	0.04	1/6	4675.8	4675.8	0.0	0.0	5324.2
Smooth	0.04	1/3	5325.6	5325.6	0.0	0.0	2297.0
Smooth	0.04	1/2	5760.4	5760.4	58.8	0.0	1921.4
Smooth	0.04	1	7433.2	7374.0	905.0	13.4	343.4
Smooth	0.08	0	0.0	0.0	0.0	0.0	10000.0
Smooth	0.08	1/6	5631.2	5631.2	0.0	0.0	4368.8

Model	Price	λ/p	S	SR	NR	D	F
Smooth	0.08	1/3	6358.4	6358.4	0.0	0.0	2139.4
Smooth	0.08	1/2	6925.8	6925.8	112.2	0.0	1652.6
Smooth	0.08	1	8871.6	8705.0	956.8	56.6	117.4

N COMPLETE PRECEDING R31 MAIN DEVELOPMENT

The complete preceding main development is retained below. Its historical empirical statements describe the R31 record, not the new R32 results. The byte-identical original source is also supplied as `R31_main_archived.tex`.

ARCHIVED R31 ABSTRACT

When can a neural network stop without computing the suffix that justifies the stop? We answer this question for bounded additive logit corrections and a specified confidence-minus-resource objective with recall. An aligned extremal completion makes the continuation certificate exact for the coupled increment contract, yielding the largest sound acceptance set available from that information. We characterize when this set is a confidence threshold and compile its high-confidence branch into one integer comparison after score normalization. Capacity and projection bounds expose the cost of imposing the contract, while a separate return criterion permits later labels to change. A preregistered five-seed convolutional Fashion-MNIST study measures that cost: at bound 0.25, terminal accuracy is 0.69 percentage points below the paired unconstrained model. Nevertheless, certification adds 851–1,313 acceptances beyond trivial headroom per 10,000 inputs across four prices. A training-frozen CPU tariff includes gate overhead before acceptance; the admitted large-batch implementation attains measured speedups of 1.08–1.14, while declining the slower small-batch regimes. Same-model IBP and CROWN comparisons, exact integer replay, and preserved adverse experiments separate guarantee, accuracy, and systems benefit.

O INTRODUCTION

Adaptive neural depth has two distinct problems: learning a useful inference policy and establishing that a proposed stop cannot forgo too much value. Early-exit and conditional-computation networks address the former (Graves, 2016; Teerapittayanon et al., 2016; Huang et al., 2018; Wang et al., 2018). The latter is difficult because evaluating the supposedly skipped suffix can consume the very resource being saved. Recall adds an accounting constraint: a policy may return an earlier prediction after inspecting later exits, but must pay for every executed transition. A statement about the returned exit alone is therefore not a statement about the cost of inference.

We develop a contract-to-controller construction for this setting. Each correction stage updates a persistent logit vector by a known bounded amount. This architecture exposes exactly the information needed to bound unseen confidence, without a grid over hidden representations. The endpoint softmax calculation is elementary. Its useful consequence is stronger: a single admissible sequence of aligned increments attains all endpoint supports simultaneously. The resulting stopping region is thus information-optimal for the *coupled increment contract*, rather than merely an arbitrary interval relaxation. The qualification matters: exploiting trained weights can tighten it, and optimality here does not compare arbitrary architectures or objectives.

An apparent difficulty is that the certificate often resembles an ordinary confidence threshold. We make this relationship explicit rather than treating near-equal actions as evidence of a new policy frontier. At record prefixes, the exact confidence acceptance region is an intersection of two-branch quadratic sublevel sets. Its high-confidence branch is a threshold, which can be compiled offline with exact rational arithmetic. At nonrecord prefixes, the recall record is indispensable; for preservation of the final returned label, the runner-up probability also matters. These distinctions identify both when a confidence heuristic can reproduce the certified actions and when current confidence alone is insufficient.

The architectural restriction is not free. We characterize the amendment class available under a fixed total increment budget and give a deterministic projection certificate for an unconstrained

model. A controlled five-seed convolutional experiment then measures accuracy, calibration, clipping, quantization, stopping, and resource cost on Fashion-MNIST. Unlike the retained MNIST example, its fitted confidence controller frequently chooses different actions and incurs observed violations of the registered score-regret tolerance. Neither score validity nor a successful projection bound is substituted for task correctness. The central contribution is the executable connection from an explicit output contract to a sharp, finite, resource-priced stopping rule; the analysis and experiments test the assumptions and costs of that connection.

P Related work

The finite-horizon depth formulation in Qi (2025) motivates our with-recall objective. Learned halting, routing, and adaptive prediction (Graves, 2016; Bolukbasi et al., 2017; Huang et al., 2018; Wang et al., 2018) optimize task or surrogate loss under a compute budget. Learned continuation values connect to simulation-based and neural optimal stopping (Longstaff & Schwartz, 2001; Becker et al., 2019); recall-aware threshold theory studies modeled loss trajectories (Yang et al., 2026b;a). Our pointwise certificate instead controls the worst completion consistent with an observed prefix and a fixed contract. Statistical alternatives, including nested early-exit prediction sets and compute-risk control (Jazbec et al., 2024; Wang et al., 2026), provide different guarantees under sampling assumptions.

Elboher et al. (2025) directly study formal verification of early-exit networks. Their central property is local input robustness under conditional exit execution, including guards ensuring that preceding exits were not taken. Their solver-based treatment is therefore a close antecedent, not subsumed by our work. We instead quantify confidence-minus-cost regret against all depths with recall. A general verifier could encode this property by replacing the robustness predicate with regret and recall comparisons. Our specialized increment contract makes those queries closed form and, on a high-confidence branch, compilable to a single integer cutoff. Conversely, our certificate does not replace their input-robustness guarantees or establish their solver complexity for unrestricted networks. The return-aware test concerns which exit wins recall, rather than which first confidence threshold is crossed.

Interval and abstract-interpretation bounds are established verification tools (Cousot & Cousot, 1977; Gowal et al., 2018). CROWN and optimized linear relaxations tighten neural output bounds (Zhang et al., 2018; Xu et al., 2021); split constraints support complete branch-and-bound verification (Wang et al., 2021). We use the authors' auto_LiRPA implementation for a same-candidate comparison, not an in-house substitute under its name. These weight-aware methods can improve the contract certificate, but require executing a verifier on the suffix graph. Historical interval, proof-library, and shortest-path constructions are retained in the appendices; they are not independent claims of a new generic verification algorithm.

Q A sharp stopping certificate from an architectural contract

A fixed network has exits $0, \dots, L$, logits $z_j \in \mathbb{R}^K$, and declared scores $g_j \in [0,1]$. Set $C_0 = 0$ and $C_t = \sum_{s=1}^{t} c_s$ with $c_s \geq 0$. With recall, $B_t = \max_{j \leq t} g_j$ and $R_t = B_t - C_t$. The policy returns the label of the shallowest maximizing score, but pays C_t. For hindsight regret define

$$\mathfrak{G}_t = \max_{0 \leq j \leq L} R_j - R_t = \max\{\mathfrak{P}_t, \mathfrak{S}_t\}, \qquad \mathfrak{P}_t = \max_{j \leq t} R_j - R_t. \tag{33}$$

The observed $\mathfrak{P}_t$ cannot be repaired by a suffix bound. Nonnegative costs imply the with-recall reduction

$$\mathfrak{S}_t = \left[\max_{j>t}\{g_j - B_t - D_j\}\right]_+, \qquad D_j = C_j - C_t. \tag{34}$$

A newly observed record is best charged when it first appears, not at a later exit; an old record cannot create positive future improvement. The empty maximum at $t = L$ contributes zero. This also covers zero-cost transitions.

The architectural contract is

$$z_j = z_{j-1} + \eta_j, \qquad \|\eta_j\|_\infty \leq \beta_j, \qquad r_j = \sum_{s=t+1}^{j} \beta_s. \tag{35}$$

Clipped output corrections enforce it regardless of hidden dimension. Let $\mathcal{A}_t$ be *all* suffix sequences obeying this contract and the observed prefix. For real softmax confidence, put

$$p = \max_k \operatorname{softmax}(z_t)_k, \quad Q_j(p) = \frac{e^{2r_j}p}{1 + (e^{2r_j} - 1)p}, \quad U_t = \left[\max_{j>t}\{Q_j(p) - B_t - D_j\}\right]_+. \tag{36}$$

Theorem 17 (Information-optimal contract test)**.** *Under* (35)*, the exact worst-case regret over $\mathcal{A}_t$ is $\max\{\mathfrak{P}_t, U_t\}$. If a known cumulative verification charge $Z_t \geq 0$ is paid by the deployed policy but not the comparator, the maximal sound acceptance test using only this information is*

$$\max\{\mathfrak{P}_t, U_t\} + Z_t \leq \lambda. \tag{37}$$

With a fixed inspection and charging schedule, its first successful prefix has minimum executed depth among prefixes admitting such a certificate.

Proof idea. Increasing a prefix-largest logit by r_j and decreasing all competitors by r_j gives Q_j. Applying the corresponding signed increment at each stage attains these maxima simultaneously. Thus the same coupled completion witnesses any suffix violation; a past violation is already observed. Appendix V.1 proves the claim, including finite integer scores. This is not exact reachability for trained weights, nor optimality over inspection schedules, learned policies, or neural architectures.

Q.1 What is lost by imposing the contract?

Suppose a feature h remains available to each correction, and write $B = \sum_{s=1}^{m} \beta_s$ for the total amendment budget. The following statement separates output capacity from representation learning.

Proposition 18 (Amendment capacity)**.** *Allowing arbitrary readout functions of the persistent feature, the attainable amendments are exactly the $[-B, B]^K$-valued functions. The best uniform infinity-norm approximation error to a target amendment f is $\sup_h(\|f(h)\|_\infty - B)_+$. If a neural readout approximates a feasible amendment to error δ, clipping and proportional decomposition across stages preserve δ. Rounding the prefix and m updates at scale S adds at most $(m+1)/(2S)$ when $S\beta_s$ are integers.*

The constructive decomposition is $\eta_s = (\beta_s/B)a$. Coordinate clipping is realizable with two ReLUs per output. A required amplitude D therefore needs at least $\lceil D/\beta \rceil$ stages in the worst case under a common bound β. More stages restore amendment capacity but increase both work and the unseen confidence envelope. The proposition does not promise that a fixed-width network learns the target, or recovers information lost by its prefix. An identity feature channel supplies the persistent-feature condition.

Proposition 19 (Post-training projection)**.** *If hidden dynamics do not depend on the persistent logits, projecting raw updates u_s to the contract and rounding gives*

$$\|\widehat{z}_j - z_j\|_\infty \leq E_j := \sum_{s=1}^{j} \|u_s - \operatorname{clip}_{[-\beta_s,\beta_s]} u_s\|_\infty + \frac{j+1}{2S}. \tag{38}$$

A raw winning margin greater than $2E_j$ preserves the label. Confidence changes by at most $E_j/2$, and cross-entropy by at most $2E_j$.

Both compared prefixes use the same initial clipping. Without that convention its clipping error must be added. The hidden-state assumption holds for our MLP and convolutional/residual examples; logit-to-hidden feedback would require another stability argument. Proofs are in Appendix V.3.

R Threshold geometry, finite compilation, and return behavior

R.1 When certification is a confidence threshold

At a record prefix $B_t = p$, each future constraint in (36) is $h_s(p) \leq a$, with $s = e^{2r_j}$ and $a = \lambda - Z_t + D_j$.

Theorem 20 (Exact threshold geometry). *For $s > 1$, let $h_s(p) = (s-1)p(1-p)/(1+(s-1)p)$. Its maximum is $(\sqrt{s}-1)/(\sqrt{s}+1)$. For $0 \le a$ below this maximum, its sublevel set is $[0, p_-] \cup [p_+, 1]$, where*

$$p_\pm = \frac{1 - a \pm \sqrt{(1-a)^2 - 4a/(s-1)}}{2}. \tag{39}$$

Larger a imposes no restriction. Intersect over depths and with $p \in [1/K, 1]$. On $p \ge 1/2$ the result is a single upper-threshold set; this holds on the entire feasible range if every active lower root is below $1/K$. Negative available tolerance cannot certify a nonnegative regret.

For the retained MNIST contract with three 0.25-bounded corrections, four prices $0.01, 0.02, 0.04, 0.08$, and tolerance one-third of price, the uncharged real-score thresholds are $0.984086, 0.967975, 0.935105, 0.866253$. A tuning-fitted controller can approximate these values, explaining its previous near-equivalence. The theorem supplies their universal contract meaning, not a new name for threshold fitting. Trivial first-exit headroom requires $p \ge 1 - \lambda - c_1$. A lower structural cutoff creates a certified confidence interval beyond that bound. Its empirical mass is reported separately from total acceptance. At a nonrecord prefix, $Q_j(p) - B_t$ depends on history: $p = 0.6, r = 0.5, \lambda + D = 0.03$ is admissible when $B_t = 0.8$ and inadmissible when $B_t = 0.6$, with zero past regret in both cases. Appendix V.2 constructs feasible histories.

R.2 An exact finite score and a one-comparison online test

We deploy integer logits q at scale $S = 64$, clipped prefix logits in $[-8, 8]$, and integer bounded amendments. Let $M = 2^{48}$ and $E^-[d]$ be a monotone integer lower table for $M \exp(-d/64)$, with $E^-[0] = M$. Define

$$D(q) = \sum_k E^-[\max_\ell q_\ell - q_k], \qquad \widetilde{g}(q) = \frac{M}{D(q)}. \tag{40}$$

The retained rational table construction gives uniform confidence error $\epsilon_E = 730/M < 2.594 \times 10^{-12}$ and regret transfer at most $2\epsilon_E$. Integer endpoint boosts and cross-products implement the full finite-score contract test without floating-point decisions. The prefix and update clipping maps, tie rules, table proof, and nonfinite-input behavior are specified in Appendix AI.2. This contract verifies the score and gate, not the operating system or all network arithmetic.

The threshold theorem removes most online gate work. Offline, for each future integer radius R_j, form $\bar{s}_j = M/E^-[2R_j] \ge e^{2r_j}$. Binary-search an integer m such that $p_* = m/M - \epsilon_E \ge 1/2$ and

$$h_{\bar{s}_j}(p_*) \le \lambda - Z_t + D_j - 2\epsilon_E \quad \text{for all } j > t. \tag{41}$$

Every operation in this search uses rational arithmetic. At a first exit, the live gate is simply

$$D(q_0) \le \left\lfloor \frac{M^2}{m} \right\rfloor. \tag{42}$$

Monotonicity of h_s above $1/2$ proves soundness; if no m exists the gate rejects all inputs. This is a conservative compilation of the high-confidence branch, not an assertion of exact equivalence on all possible integer states. It has zero action differences from the full finite gate in all 800,000 registered comparisons across seeds, bounded variants, prices, and test rows. Score normalization is not free: the end-to-end measurements include it, and setup includes verifying the table and compiling the cutoff. The online incremental decision itself is one integer comparison.

A sufficient task-preservation region. Confidence regret and correctness are distinct, but the output contract can sometimes explain their observed agreement. If the initial confidence obeys $p > 1/(1 + e^{-2r})$, then $p/(1-p) > e^{2r}$, and hence every pairwise winning margin exceeds the largest possible adverse logit change $2r$. All future raw labels are then identical. For the 0.25 contract, $r = 0.75$ gives the sufficient threshold 0.817575, below all four scalar cutoffs above. Finite-score transfer is far smaller than this gap. Thus those scalar acceptance regions also preserve every raw label under the contract; their zero observed task continuation regret is not evidence that high confidence implies a correct label. A confidently wrong accepted prefix can remain wrong at every exit. Other bounds or prices need not place scalar acceptance inside this stronger label-preserving region.

R.3 Preserving the return is a different property

At the first exit let p_1 be the winning probability, p_2 the largest competitor probability, and r the total remaining radius. The coupled increment contract preserves the *full-recall returned label* exactly when

$$\frac{e^{2r}p_2}{1+(e^{2r}-1)p_2} \le p_1. \tag{43}$$

A future different label cannot win recall unless its confidence exceeds the already observed p_1. Conversely, aligned increments for the runner-up attain the left-hand side; once it exceeds p_1, that class wins. Equality preserves the earlier return. This is exact for the increment uncertainty model, not the trained neural reachable set. It can hold even when the raw future label changes, and it depends on p_2, unlike the scalar first-exit gate. Its integer analogue is a boosted-runner denominator comparison. The more general product-box theorem, including nonnested heads and strict/weak exit ties, remains in Appendix AJ. These behavioral tests are never pooled with scalar-regret acceptances.

S Resource charges before the decision

A registry fixes the resource unit, its conversion to utility, the test schedule, and the charge for every check, including rejected checks. Formula (37) subtracts that fee from the admissible tolerance *before* choosing the stop. With a valid upper charge bound it remains sound, though no longer maximal for the unknown exact charge. The theorem certifies the declared objective; a measured timing tariff does not become a worst-case hardware bound merely by being written in the registry.

We use two explicit objectives. A weighted-work tariff prices each network MAC at one credit and a complete first-exit gate at 4,096 credits. This is a declared utility accounting unit, not an instruction count or a latency measurement. A separate CPU tariff fits per-stage durations, dispatch costs, and amortized table/compilation setup on reserved training inputs, then freezes them before timing test inputs. Conversion is fixed so the full suffix costs 0.04 utility units and the tolerance is $0.04/3$. All gates consume their registered fee even if inference subsequently continues.

Proposition 21 (Net-resource admission)**.** *For N checks, A early accepts, avoided suffix cost at least s_ℓ per accept, full gate cost at most o_u per check, and setup at most F, total saving relative to always executing the suffix is at least $As_\ell - No_u - F$. Thus $A/N \ge (o_u + F/N)/s_\ell$ is sufficient for net saving. Nonadditive batching requires a batch-level cost model.*

The bound describes a robust admission condition when its envelopes are valid. Our CPU admission uses training-row paired timings instead: it selects the gated path only when the median pilot saving is positive, and freezes that selection. Held-out measurements then test, rather than assume, the prediction. A failed first-exit check followed by full inference is charged but is not automatically a λ-optimal stopping certificate: terminal past regret can be positive. We report early-admission guarantees and fallback behavior separately. Appendix V.4 proves these distinctions.

T Experiments

T.1 Prospective convolutional replication and architectural ablation

Fashion-MNIST (Xiao et al., 2017) supplies 60,000 training and 10,000 test images in ten classes. Before training we committed the complete five-seed family, split, architecture, optimizer, schedule, price family, and six comparisons. A common split uses 50,000 training, 5,000 controller-fit, and 5,000 tuning rows. Seeds 3101–3105 train a two-layer convolutional prefix with 8 and 16 channels, 3×3 stride-two kernels, and a 128-dimensional feature. Three residual hidden stages add clipped logit corrections. The prefix trains for four epochs and is then frozen; each suffix trains for four epochs with Adam, learning rate 10^{-3}, and batch size 256. Paired variants share prefix weights, suffix initialization, and minibatch order.

The trained bounds are $0.125, 0.25, 0.5$, plus an otherwise identical unconstrained suffix. Two fixed-weight comparisons remove rounding from the 0.25 model or project the unconstrained model to 0.25 without retraining. Within each seed, all candidate weights, thresholds, and hashes are frozen before the first request for the official test files. No seed, epoch, or variant is selected from test performance.

Table 20: Fashion-MNIST contract ablation: all five seeds, mean ± sample SD in percentage points. Raw terminal and full-recall accuracies are separate; ECE uses 15 bins. Float and projection rows reuse frozen weights, without retraining.

Variant	Terminal accuracy	Recall accuracy	Terminal ECE
$\beta = 0.125$	85.37 ± 0.61	85.24 ± 0.63	1.10 ± 0.26
$\beta = 0.25$	85.38 ± 0.63	85.45 ± 0.68	1.19 ± 0.31
$\beta = 0.5$	85.05 ± 0.69	85.42 ± 0.65	0.93 ± 0.24
Unconstrained	86.07 ± 0.49	85.99 ± 0.57	1.31 ± 0.27
$\beta = 0.25$, float	85.37 ± 0.62	85.45 ± 0.68	1.17 ± 0.27
Projected to 0.25	85.70 ± 0.60	85.54 ± 0.70	0.72 ± 0.19

Table 21: Fashion-MNIST, $\beta = 0.25$, averages across five seeds. Counts are per 10,000 test inputs. Extra is beyond the same-row trivial-headroom test. Charged decisions include a predeclared 4,096-credit gate fee before acceptance.

Price	Accept	Extra	MAC saved	Charged accept	Net credits saved
0.01	3376.6	850.8	7.95%	3294.8	5.93%
0.02	4145.0	951.0	9.75%	4053.2	7.72%
0.04	5022.2	1085.4	11.82%	4906.0	9.73%
0.08	6074.8	1312.8	14.30%	5930.4	12.14%

This is prospective relative to these candidates, not a globally untouched public benchmark or research question. Main tables use means and sample standard deviations across five training seeds, not five independent test datasets. Complete seed-wise results, all four prices, and conditional row-frequency intervals appear in Appendix W.

Table 20 quantifies the restriction rather than assuming it harmless. The paired terminal-accuracy difference between bound 0.25 and unconstrained training is -0.690 ± 0.487 percentage points. Removing rounding alone changes this by only 0.006 ± 0.015 points in favor of the integer model; that small descriptive difference is not evidence of a quantization benefit. Post-training projection costs 0.366 ± 0.321 points relative to the unconstrained terminal head. Its deterministic margin certificate preserves 2,076–3,628 terminal labels per seed; actual agreement is 9,529–9,716 of 10,000. All logit projection inequalities hold. The gap measures conservatism, not a failed inequality. Clipping is active, and per-stage coordinate clipping fractions and raw update-norm quantiles are reported rather than interpreting the global bound as a fitted typical norm.

T.2 Scalar actionability, policy comparisons, and task risk

The convolutional prefix costs 172,192 network MACs; each correction costs 17,664, and full inference costs 225,184. At four fixed total prices we use tolerance one-third of price and evaluate first-exit certification. The baseline trivial-headroom bound is $1 - g_0$ less the cheapest future cost; extra counts compare the two tests on the same rows and objective. Table 21 shows positive extra acceptance in every seed. At price 0.04, bounds $0.125, 0.25, 0.5$ add respectively 1,780.8, 1,085.4, and 553.8 accepts per 10,000 on average. Their network-MAC savings are 13.45%, 11.82%, and 10.57%. The smaller bound offers a tighter continuation envelope, while Table 20 records its accuracy cost. The projected model shares its prefix and contract with the 0.25 model, so their first-exit acceptance sets coincide despite different suffix accuracy.

A tuning-frozen confidence controller maximizes mean confidence-minus-cost among a fixed threshold grid. It is not fitted to the certificate's worst-case criterion. For bound 0.25, its decisions differ from the certificate on 1,243.8, 1,518.4, 2,225.4, and 3,182.8 rows across the four prices, all through additional heuristic accepts. Of those accepts, 362.4, 457.2, 693.2, and 933.8 have *observed* hindsight score regret above the registered tolerance. At price 0.08, 102.0 heuristic-accepted rows have an incorrect prefix and a correct future exit; this count is zero for the certified accepts. Thus the distinction changes actual decisions and their certified objective, not only terminology. It does not imply that worst-case certification always maximizes average accuracy or average utility.

Table 22: Fixed-model task baselines, Fashion-MNIST $\beta = 0.25$, five-seed means. Accuracy is mean $\pm$ sample SD in percentage points. Price is 0.04. Network MACs exclude controller overhead; matching is fixed on tuning rows and only approximate on test.

Policy	Accuracy (%)	Mean MACs
Fixed prefix	85.058 ± 0.700	172192
Raw terminal	85.380 ± 0.630	225184
Full recall	85.446 ± 0.684	225184
Temperature-calibrated recall	85.454 ± 0.673	225184
Certificate	85.446 ± 0.684	198570
Ridge, matched terminal	85.494 ± 0.603	198336
Ridge, matched calibrated recall	85.440 ± 0.655	198336
Ridge, task-cost terminal	85.276 ± 0.669	176553

The scalar theorem is not a correctness theorem. Every certified accepted row in the bounded Fashion-MNIST family empirically agrees with terminal and full-recall labels and has zero task continuation regret, but those are separate rowwise measurements. For the 0.25 model the resulting accuracy is 85.446% at each price, equal to its full-recall policy; its raw terminal accuracy is 85.380%. A separate tuning-fitted prefix-versus-terminal policy, fixed-depth heads, and expected compute-matched randomized recall/terminal mixtures are all retained. Task-aware selection can trade the guarantee for more aggressive compute savings, so a confidence-regret certificate is not presented as task-optimal architecture search. Calibration is also measured rather than assumed: ECE and negative log likelihood are reported at every head, with rational scores used for all integer-model policy comparisons and real softmax used for floating variants.

T.3 Learned terminal selection and calibrated recall

Full recall is the fallback for the declared confidence objective, not a presumption that the most confident exit is the most accurate one. We therefore add a fixed-model study that fits stronger task comparators on the reserved controller-fit split. One temperature per exit minimizes negative log likelihood on the 5,000 fitting rows, with temperatures restricted in advance to $[0.25, 4]$. A ridge controller predicts the terminal-minus-prefix correctness advantage from the 128 prefix features, ten class probabilities, confidence, margin, and entropy. Its regularization is fixed at 10, without test selection. The cost-based rule continues when predicted advantage exceeds the total price. For a compute-matched alternative, a threshold quantile on the separate tuning split targets the certificate's tuning acceptance frequency. Thus matching does not use test labels or force identical test-set compute.

Table 22 includes every fitted comparator. At price 0.04, the matched ridge/terminal rule averages 85.494% accuracy and 198,336 MACs, against 85.446% and 198,570 for certification. This is a small descriptive improvement in both quantities, not a statistically established population frontier; it also lacks the contract-regret guarantee. The more aggressive task-cost rule uses 176,553 MACs and reaches 85.276%. Temperature-calibrated full recall reaches 85.454%, close to uncalibrated recall, so simple miscalibration is not established as the sole cause of imperfect exit selection. Changing the recall score changes the policy and requires a new certificate; one cannot inherit the rational-score guarantee merely by attaching calibrated probabilities. Controller execution cost is excluded from these network-MAC comparisons and is not inferred from the separate compiled-gate CPU study.

The five model families were already evaluated when this additional controller study was specified. Each new controller is fitted only on training-derived splits and hashed before its test-trace access, but the study is explicitly a retrospective analysis of fixed models, not another untouched-dataset replication. All prices, fitted temperatures, controller coefficients, and per-seed task results are retained in Appendix Y.

T.4 Costed gates and complete CPU measurements

Charging the 4,096-credit gate *before* acceptance removes 81.8–144.4 accepts per 10,000, depending on price. Net credit savings in Table 21 deduct the gate fee for all inputs, including rejects; they

Table 23: Held-out complete-pipeline CPU results, ranges over three processes. Speed is median full-path time divided by median forced-gated time. Admission is fixed using training inputs, not these test timings.

Batch	Test inputs	Early accepts	Forced-gate speed	Admitted path
1	128	0	0.910–0.916	Full
32	992	427–458	0.986–0.989	Full
1,000	1,000	485–487	1.082–1.144	Gated

are smaller than network-MAC savings. The exact-rational compiler then permits a lower-overhead implementation, whose complete runtime we measure independently under a hardware-specific tariff.

For the preregistered network seed 3101, three independent CPU processes use 1,000 reserved controller-fit inputs to calibrate batch sizes 1, 32, and 1,000. Stage costs include score updates; dispatch fees use the largest pilot query duration, and setup includes table verification and cutoff compilation amortized over 100,000 inputs. The complete tariff and admission decision are hashed before test access in each process. Calibration uses nine interleaved full/gated timing pairs; evaluation uses fifteen independently randomized pairs on 128, 992, and 1,000 official-test inputs, respectively. These are process replications on AMD EPYC 7763 CPUs, not three hardware architectures.

The first two batch regimes are declined by the training-frozen admission rule; their forced-gate slowdowns remain visible in Table 23. At batch 1 the registered gate fee alone exceeds the tolerance. At batch 1,000 the admitted implementation is faster in all three processes, with paired-bootstrap 95% intervals $[1.070, 1.091]$, $[1.058, 1.332]$, and $[1.079, 1.097]$. These intervals summarize repeated timings, not hardware-wide worst-case guarantees. All admitted early stops pass the original exact integer inequalities with their frozen fees; no returned-label differences are observed. Startup is represented in the tariff by its stated amortization, not concealed inside a claim of zero setup cost. Accelerator speed or energy savings are not inferred from CPU or MAC measurements.

T.5 Weight-aware verification on the same candidate

We compare the frozen MNIST candidate against the authors' auto_LiRPA 0.7.2 implementation of IBP, CROWN, optimized CROWN (20 iterations), and four-child IBP splitting. The fixed first 128 retained inputs supply conditional 128-dimensional hidden-state boxes of radius 0, 0.01, and 0.1, holding prefix logits fixed. Each method bounds the *same* continuation-confidence property at price 0.04. We add explicit per-stage rounding allowances, intersect with the structural bound, and retain all output arrays and timings. These are float64 verifier diagnostics, not outward-rounded deployment proofs or a benchmark of input robustness.

The structural rule admits 87 of 128 inputs. At hidden radius 0.01, IBP, CROWN, optimized CROWN, and splitting admit 89, 93, 93, and 89. Their bound computation times are 3.32 ms, 204.79 ms, 24.30 s, and 7.47 ms; evaluating all structural caps takes 0.043 ms. At radius zero all four methods admit 93; at radius 0.1 they all admit 87, although optimized CROWN still tightens average caps. These results demonstrate a real tightness–cost trade-off, not dominance of the structural contract or competitiveness with complete branch-and-bound on unrestricted networks. Graph construction takes 19.14 ms; output-bound arrays occupy 60 KiB, or 240 KiB for splitting, while peak process memory is about 2.01 GiB. Appendix X reports all conditions.

T.6 Variability in usefulness and verification scope

Certificate validity is deterministic under the contract, but usefulness frequency varies across models and inputs. Table 24 reports variation across independent initialization/minibatch seeds on a common split and test set, not an artificial pool of 50,000 independent inputs. Per-model Wilson intervals, correction/spoil counts, calibration, and paired task comparisons are retained separately. Frequency intervals have a sampling interpretation; they are not additional formal certificates or uncertainty over new datasets.

The convolutional model family and six architectural comparisons were committed before training. Hardware tariffs were fitted on reserved training inputs; learned-policy, projection, return-witness,

Table 24: Usefulness variability across all five Fashion-MNIST models at $\beta = 0.25$: mean $\pm$ sample SD, in percentage points. Network-MAC savings exclude the gate fee, unlike the net-credit column of Table 21.

Price	Certified acceptance	Extra acceptance	MAC saving
0.01	33.77 ± 2.03	8.51 ± 0.56	7.95 ± 0.48
0.02	41.45 ± 2.04	9.51 ± 0.41	9.75 ± 0.48
0.04	50.22 ± 2.11	10.85 ± 0.58	11.82 ± 0.50
0.08	60.75 ± 2.12	13.13 ± 0.32	14.30 ± 0.50

and modern-verifier studies analyze fixed models retrospectively. These stages are not pooled as independent confirmation. Freezing a computation before its local test access prevents within-run tuning but does not erase prior research exposure.

Validation independently reconstructs the rational score table, verifies all five candidate freezes, and replays 800,000 uncharged and 800,000 charged scalar decisions. The compiler is additionally checked on 36,000 random charged configurations, and 43,740 small-state coupled completions exhaustively test the scalar extremizer and returned-label witness. These regression checks complement the proofs; they do not replace universal arguments with random testing. All finite convolutional traces satisfy their integer radius and score-domain constraints. The retained MLP's total nonfinite maps and the convolutional timing script's finite-output domain are distinguished in Appendix V.6 and its following numerical-domain note. Deploying outside the latter domain requires a total bounded conversion or fail-closed checks, whose cost belongs in the applicable tariff.

T.7 Retained evidence and behavioral diagnostics

The previously frozen 128-feature MNIST model retains its full 10,000-row results: 1,408–1,911 extra scalar accepts and 14.13–27.28% MAC savings, with 94.97% full-recall accuracy. Its more accurate raw terminal head is 95.40%, and the fitted confidence controller differs on only 1, 4, 1, and 2 rows. This remains an instance of nearly identical actions with different guarantee semantics, not evidence that the heuristic was outperformed. A new frozen-trace analysis finds 9,314 return-preserving contract certificates against 8,722 strict all-label-invariance certificates. All 592 additional behavioral accepts are accounted for separately from scalar stopping. Of 686 violating contract witnesses, 487 nevertheless have unchanged returns on the actual trained trajectory: contract uncertainty is materially larger than neural reachability. Observed score/task-regret correlations are only about 0.16.

The older two-dimensional verifier, every whole-cell proof object, and all historical derivations remain in the appendices. Its global grid-256 error bounds 8.65, 48.61, and 275.87 exceed the utility range, and its first exit outperforms full recall. The archived CIFAR-100 geometry equals trivial headroom rowwise and SVHN transport coverage is 93.65% at nominal 95%. These failures motivated the output-contract strategy; they are not recast as positive high-dimensional verification evidence. Fashion-MNIST adds a controlled convolutional replication, but does not establish accuracy or resource gains for large residual networks, transformers, or deployment shifts.

U Conclusion

Bounded logit corrections expose an information-optimal stopping certificate for a declared with-recall objective. Threshold geometry turns that certificate into an inexpensive finite controller, while capacity, projection, and predecision resource charges make its restrictions operational. The experiments show where certification adds decisions beyond trivial headroom, where a heuristic violates the stated score tolerance, and where complete runtime supports deployment. They also quantify architectural accuracy loss and verification conservatism. Optimal stopping under an explicit information and resource contract is the claim; task-optimal depth, unrestricted neural reachability, and universal hardware acceleration are different claims.

V INFORMATION-OPTIMAL STOPPING, THRESHOLD GEOMETRY, AND CONTRACT CAPACITY

This appendix proves the statements used in the main text. The uncertainty model and the comparator are part of each statement. In particular, information-optimality within an increment contract is not optimality over all weights, architectures, learned policies, or implementations.

V.1 THE SHARP CERTIFICATE FOR A COUPLED INCREMENT CONTRACT

Let $\mathcal{A}_t$ contain all future logit sequences satisfying the known prefix and $z_j - z_{j-1} \in [-\beta_j, \beta_j]^K$. For $j > t$, write $r_j = \sum_{s=t+1}^{j} \beta_s$, $D_j = C_j - C_t$, $p = \max_k \mathrm{softmax}(z_t)_k$, and

$$Q_j(p) = \frac{e^{2r_j} p}{1 + (e^{2r_j} - 1)p}, \qquad U_t = \max\{0, \max_{j>t}(Q_j(p) - B_t - D_j)\}.$$

Proof of Theorem 17. For any completion, a record from an already observed exit cannot create a positive future reward increment: its score is at most B_t, and subsequent costs are nonnegative. A new record at depth $i \le j$ is best charged at i, not at a later j. Consequently

$$\max_{j \ge t}(R_j - R_t) = \max\{0, \max_{j>t}(g_j - B_t - D_j)\}.$$

Taking a supremum commutes with the finite maximum. For every class k, its softmax is maximized over the radius-r_j coordinate box by increasing its own coordinate by r_j and decreasing all others by r_j. The resulting probability is $e^{2r_j} p_k / (1 + (e^{2r_j} - 1)p_k)$, increasing in p_k. Choose any current-largest class k_0. The coupled completion $\eta_{s,k_0} = \beta_s$ and $\eta_{s,k} = -\beta_s$ for $k \ne k_0$ attains these maxima *simultaneously at every future depth*. Thus there is no additional loss from replacing the coupled increment contract by its marginal boxes for this particular scalar objective. This observation does not make the boxes exact for a particular trained neural suffix, which can impose further constraints on the updates.

The prefix term $\mathfrak{P}_t$ is observed and fixed, so the exact worst-case hindsight regret is $\max\{\mathfrak{P}_t, U_t\}$. A known cumulative charge Z_t subtracts from the reward of the deployed policy but not from the stipulated uncharged comparator; the exact worst-case charged regret is therefore $\max\{\mathfrak{P}_t, U_t\} + Z_t$. If this exceeds ϵ, the aligned completion above or an already observed prefix exit witnesses the violation. Any rule that receives only this contract information and accepts this same prefix would also accept that violating completion. Conversely the displayed test bounds every completion. It is therefore the maximal sound acceptance set for the declared information and comparator. With a fixed sequence of inspections and charged tests, its first successful prefix is the least executed depth at which such a certificate exists. A procedure that exploits additional weight-dependent information or chooses a different test schedule is not included in this last comparison. □

The same argument works for the finite integer score of Appendix AI.2: replace radii by integer radii, select a prefix-largest integer coordinate, and use the monotone table denominators. All-aligned integer increments attain the structural maxima. This extends information-optimality to the declared finite score without importing a floating-point comparison into the gate.

V.2 COMPLETE CONFIDENCE-THRESHOLD CHARACTERIZATION

Proof of Theorem 20. For one future depth put $s = e^{2r} > 1$ and $a = \lambda + D$. At a record prefix, $B = p$, the admissibility inequality is $h_s(p) \le a$, where

$$h_s(p) = \frac{(s-1)p(1-p)}{1 + (s-1)p}.$$

Differentiation gives

$$h_s'(p) = \frac{(s-1)\{1 - 2p - (s-1)p^2\}}{\{1 + (s-1)p\}^2}.$$

There is exactly one stationary point in $(0,1)$, namely $p_* = (1+\sqrt{s})^{-1}$; it is the maximum, with $h_s(p_*) = (\sqrt{s}-1)/(\sqrt{s}+1) = \tanh(r/2)$. If a is at least this value, the depth imposes no restriction. Otherwise multiplication by the positive denominator gives an upward quadratic whose roots are

$$p_\pm(s,a) = \tfrac{1}{2}\left(1 - a \pm \sqrt{(1-a)^2 - 4a/(s-1)}\right).$$

The admissible set on $[0,1]$ is $[0,p_-] \cup [p_+,1]$, including the equality endpoints. The $s=1$ case has $h_s = 0$ and imposes no restriction for $a \geq 0$. Intersect these sets over future depths and with $p \in [1/K, 1]$. On $[1/2,1]$ all nonconstant h_s are decreasing, so the intersection is a single upper-threshold set. If every active $p_- < 1/K$, the same single-threshold statement holds on the entire feasible confidence range. More generally the full set intersection, not an assumed threshold, is the exact rule. □

For the three $\beta = 0.25$ transitions and prices $0.01, 0.02, 0.04, 0.08$ with λ one-third of the total price, all lower roots are below $1/10$. The exact real-score thresholds are, respectively, 0.9840861202, 0.9679753821, 0.9351054073, and 0.8662534473. Each is attained by the second future-depth constraint. A fitted confidence policy can estimate essentially these cutoffs without knowing the proof, explaining near-equal actions; equivalence is not evidence that the fitted rule enjoys a uniform guarantee outside its tuning rows. The score-table error changes any deficit by at most $2\epsilon_E$, so a finite-score/real-score action disagreement can occur only within $2\epsilon_E$ of a real-score constraint boundary. A fitted threshold can also differ because it is fitted on finitely many tuning examples; those sampling differences are not attributed to numerical error.

When current confidence is insufficient. At a nonrecord prefix, the bound is $Q_j(p) - B_t - D_j$, not $Q_j(p) - p - D_j$. For fixed B_t and $v = B_t + \lambda + D_j < 1$, the condition is

$$p \leq \frac{v}{s_j - (s_j - 1)v}.$$

Thus B_t is a necessary state variable, and a rule depending only on current confidence cannot represent the general acceptance set. For example, $p = 0.6$, $r = 0.5$, $\lambda + D = 0.03$ give $Q = 0.80305\ldots$. The prefix with $B = 0.8$ is admissible, whereas the prefix with $B = 0.6$ is not. These two histories are feasible under bounded updates: in the binary case, the change in log odds from 0.8 to 0.6 is less than one, and hence can be implemented by a radius-0.5 coordinate change. Set the intervening cost to zero to remove any past-regret difference. A positive tolerance 0.03 and zero future cost then give an explicit full-regret counterexample to current-confidence-only sufficiency. This example is an analytic distinction, not a claim that the first-exit MNIST actions differ materially.

V.3 Capacity and projection cost of bounded logit corrections

Let a prefix expose a persistent feature $h \in H$ and logits $z_0(h)$. The feature must remain available to each added readout; an identity channel is sufficient. Put $B = \sum_{s=1}^m \beta_s > 0$ and let $f(h)$ be a desired logit amendment. The capacity statement concerns the output amendment with this feature held available, not the cost of learning or extracting h.

Proof of Proposition 18. Every admissible amendment $a = \sum_s \eta_s$ obeys $\|a(h)\|_\infty \leq B$. The distance from $f(h)$ to the cube $[-B,B]^K$ in infinity norm is $(\|f(h)\|_\infty - B)_+$. Its coordinatewise clipped projection attains that distance. Conversely every cube-valued amendment a can be decomposed as $\eta_s = (\beta_s/B)a$, and each term obeys its stage bound. Thus the total amendment class, allowing arbitrary readout functions, is exactly the cube-valued class. Taking the supremum over H gives the stated minimax uniform approximation error.

For neural realization, suppose a readout $v(h)$ approximates a cube-valued $a(h)$ with error at most δ. Coordinate clipping onto $[-B,B]$ is nonexpansive and fixes a, so clipping v preserves this error. The identity $\mathrm{clip}_{[-B,B]}(x) = -B + \mathrm{ReLU}(x+B) - \mathrm{ReLU}(x-B)$ realizes it with two ReLU units per output coordinate. The proportional readouts above then implement the amendment. This is a constructive transfer from a given readout approximation, not an unsupported universal rate for a fixed-width network. With integer scale S, assume $S\beta_s$ is an integer and that the initial prefix and each update are rounded to nearest. Rounding adds at most $(m+1)/(2S)$ in infinity norm, including the prefix. Thus approximation error is at most $\delta + (m+1)/(2S)$ relative to the unrounded target. □

The result exposes a depth–budget trade-off rather than asserting that small bounds are free: with a fixed per-stage bound β, a desired amplitude D needs at least $\lceil D/\beta \rceil$ stages in the worst case, and those stages incur both network work and a larger continuation envelope. Increasing depth can restore output-amendment capacity while weakening early stopping. It cannot recover information already destroyed by the prefix representation.

Proof of Proposition 19. Assume the hidden trajectory does not depend on the persistent logit state, as in both experimental architectures. Then projecting the readouts leaves every hidden state, and hence every unprojected amendment $u_s(h_s)$, unchanged. Subtracting the two logit recursions and using the triangle inequality yields

$$\|\widehat{z}_j - z_j\|_\infty \le \sum_{s=1}^{j} \|u_s - \mathrm{clip}_{[-\beta_s,\beta_s]} u_s\|_\infty + \frac{j+1}{2S} =: E_j.$$

The prefix is compared after the same initial clipping; otherwise its clipping error must be added. If a winning-class logit margin exceeds $2E_j$, no competitor can overtake it. For any softmax coordinate p_k, the one-norm of its gradient is $2p_k(1-p_k) \le 1/2$. The mean-value bound gives $|p_k(\widehat{z}) - p_k(z)| \le E_j/2$. Taking maxima preserves this bound for confidence. Cross-entropy has gradient $p - e_y$ with one-norm at most two, so its change is at most $2E_j$. These are pointwise deterministic bounds; none equates confidence regret to classification risk. With logit-to-hidden feedback, the same-trajectory argument does not apply without an additional stability bound. □

V.4 Before-decision resource charges and admission

A deployment registry specifies one conversion α from a resource unit to utility cost. Let Z_t include all tests, rejected checks, score/table accesses, dispatch, and the amortized setup charged through the proposed stop. A test is scheduled only after its charge or valid upper bound has been included in the available tolerance. The special first-exit rule is $U_0 + \bar{Z}_0 \le \epsilon$, with the fee subtracted from the tolerance *before* performing the exact integer comparison. Using an upper bound rather than an exact charge retains soundness, but not maximality of the acceptance set. The declared weighted-work experiment charges 4,096 credits per complete first-exit test and one per network MAC. This is an operational utility tariff; it is explicitly not a calibrated count of CPU instructions, a latency measurement, or an energy estimate.

Proof of Proposition 21. For N first-exit checks let A be the number of accepted inputs. Relative to running every suffix, the saving is the sum of avoided suffix costs minus all N gate costs and setup. If every avoided suffix costs at least s_ℓ, every full check costs at most o_u, and setup costs at most F, then saving is at least $As_\ell - No_u - F$. It is nonnegative precisely under the displayed sufficient admission condition. With independent tuning inputs, a fixed policy, and an exact binomial lower confidence bound a_ℓ on population acceptance, replacing A/N by a_ℓ gives an expected-saving statement at that confidence level. It does not give a deterministic bound on a finite future batch. A measured timing maximum is not automatically a hardware-wide worst-case bound; empirical and certified envelopes are kept separate. Finally, batching can make suffix cost nonadditive. The result then applies to a registered batch as the unit, with measured or bounded costs for the whole batch, not an unjustified sum of per-input latencies. □

A failed first-exit check followed by full inference remains charged. Its regret need not satisfy the early-acceptance tolerance: the terminal depth itself need not maximize the confidence-minus-cost objective. The returned certificate records its actual $\max\{\mathfrak{P}_L, 0\} + Z_L$. Reporting only the successful branch as a certified early stop prevents the fallback from being silently counted as another ϵ-optimal stop.

V.5 Sound offline compilation to a single integer comparison

Let $M = 2^{48}$ and let $\epsilon_E = 730/M$ be the uniform table error. For an integer radius R, the lower exponential bracket gives $\bar{s} = M/E^-[2R] \ge e^{2R/64}$. For $p \ge 1/2$, $h_s(p)$ is nonincreasing in p, and for fixed $p \in [0,1]$ it is nondecreasing in s. Suppose the rational compiler selects m satisfying

(41). If $D(q) \le \lfloor M^2/m \rfloor$, then $\widetilde{g}(q) = M/D(q) \ge m/M$, and the real confidence satisfies $p \ge m/M - \epsilon_E = p_* \ge 1/2$. For each future stage,

$$Q_j(p) - p = h_{s_j}(p) \le h_{\bar{s}_j}(p_*) \le \lambda - Z + D_j - 2\epsilon_E.$$

Replacing both the current score and the future structural support by their finite-table counterparts changes the deficit by at most $2\epsilon_E$. Consequently the finite-score gate with charge Z is sound. The search uses rational cross-products and bisection over integers, not floating-point root finding. The accepted denominators fit in int64 under the ten-class contract; intermediate offline products use arbitrary precision. The inequality remains valid at its equality endpoint. An empty high-confidence region is compiled to a zero cutoff, which accepts no possible positive denominator.

The implementation deliberately does not recover a possible low-confidence branch. Nor does it assert that its conservative cutoff is the exact integer threshold on all possible states. The 800,000 recorded comparisons establish zero lost actions on the registered traces and all-safe containment, not a universal equivalence theorem. The construction also explains why compilation must be redone when stage prices, fee, logit scale, class count, or increment bounds change. Compiled configuration hashes identify these dependencies.

V.6 Exact returned-label invariance for nested increment boxes

Let the observed first-exit label be k_1, its probability p_1, and the largest other class probability p_2. A future different label k can win full recall only with confidence strictly greater than p_1, because the first exit wins score ties. At every later depth its probability is at most $Q_{r_j}(p_k) \le Q_r(p_2)$, where r is the total remaining radius. This proves sufficiency of (43), including equality.

For necessity choose a runner-up class k_2 and apply $+\beta_s$ to its coordinate and $-\beta_s$ to every other coordinate at every remaining stage. Its probability increases monotonically to $Q_r(p_2)$. All other class probabilities decrease: their relative logit differences remain fixed and their denominator acquires a larger runner-up term. If $Q_r(p_2) > p_1$, then k_2 is a winning class and ultimately has a score larger than the initial score or any score of an unchanged competitor. The returned label therefore changes. The witness is a single coupled increment trajectory, not an incompatible selection of headwise boxes. It can still be impossible for a particular trained suffix, which is why retrospective neural replay can reject the contract witness.

For the finite monotone score, maximize each competing class by the same integer endpoint boost and compare its denominator strictly with the prefix denominator, checking the fixed smallest-index class tie rule. Among competitors, a largest initial logit maximizes the boosted score by the same denominator-swapping argument used for the scalar cap. Earlier unchanged classes cannot improve their score under this boost. The witness and shallow score tie convention therefore transfer to the registered integer setting. A strict raw-label margin greater than $2r$ is a stronger sufficient condition; equality rows require the class tie rule and are not counted in the reported strict-margin baseline.

Unlike scalar first-exit certification, the returned-label criterion is not a function of p_1 alone. With $r = 0.6$, the distributions $(0.6, 0.39, 0.01)$ and $(0.6, 0.2, 0.2)$ have the same maximum confidence but different returned-label certificates: boosting the runner-up gives approximately 0.680 in the first case and 0.454 in the second. These are analytic feasible-logit examples, not invented experimental observations.

V.7 Numerical domain of the convolutional evidence

The frozen convolutional traces explicitly check finite logits and bounded integer amendments; the theorem is applied to those registered finite states. The retained R30 MLP additionally specifies total nonfinite clipping maps. The R31 PyTorch timing script is not a formally verified implementation for arbitrary NaN-producing executions, and its measured timings do not prove an operating-system or floating-point instruction contract. Any deployment outside the finite-output domain requires a fail-closed finite-value check or an explicitly total bounded conversion, with its cost included in that deployment's tariff. This distinction does not alter any reported finite trace but prevents an empirical arithmetic replay from being represented as whole-program verification.

W COMPLETE REPLICATION AND POLICY DIAGNOSTICS

No seed, variant, price, or checkpoint was selected after test inspection. Tables below summarize all registered conditions. The machine-readable rows also retain Wilson intervals conditional on each fixed model; these are not confidence intervals for variability across trained models.

W.1 EVERY TRAINING SEED

Seed	Variant	Terminal (%)	Recall (%)	Accept	Extra
3101	$\beta = 0.125$	85.48	85.04	5862	1771
3101	$\beta = 0.25$	85.82	85.51	5193	1102
3101	$\beta = 0.5$	84.47	85.38	4646	555
3101	Unconstrained	86.00	85.96	–	–
3101	$\beta = 0.25$, float	85.82	85.52	–	–
3101	Projected to 0.25	85.40	85.11	5193	1102
3102	$\beta = 0.125$	84.63	84.59	5523	1791
3102	$\beta = 0.25$	84.67	84.63	4824	1092
3102	$\beta = 0.5$	84.23	84.48	4304	572
3102	Unconstrained	85.71	85.27	–	–
3102	$\beta = 0.25$, float	84.68	84.65	–	–
3102	Projected to 0.25	85.14	84.89	4824	1092
3103	$\beta = 0.125$	85.90	85.70	5805	1706
3103	$\beta = 0.25$	85.74	85.92	5105	1006
3103	$\beta = 0.5$	85.23	85.65	4602	503
3103	Unconstrained	85.88	86.10	–	–
3103	$\beta = 0.25$, float	85.71	85.92	–	–
3103	Projected to 0.25	86.06	85.99	5105	1006
3104	$\beta = 0.125$	85.98	86.08	5951	1895
3104	$\beta = 0.25$	85.95	86.27	5220	1164
3104	$\beta = 0.5$	85.93	86.27	4650	594
3104	Unconstrained	86.92	86.84	–	–
3104	$\beta = 0.25$, float	85.94	86.26	–	–
3104	Projected to 0.25	86.58	86.53	5220	1164
3105	$\beta = 0.125$	84.86	84.81	5447	1741
3105	$\beta = 0.25$	84.72	84.90	4769	1063
3105	$\beta = 0.5$	85.37	85.33	4251	545
3105	Unconstrained	85.84	85.79	–	–
3105	$\beta = 0.25$, float	84.72	84.90	–	–
3105	Projected to 0.25	85.34	85.16	4769	1063

Acceptance columns use price 0.04 and tolerance $0.04/3$. A missing entry means that no finite bounded-increment certificate is claimed for that score/architecture variant.

W.2 ALL FOUR PRICES AND BOUNDED VARIANTS

Variant	Price	Accept	Extra	MAC (%)	Charged	Acc. (%)
$\beta = 0.125$	0.01	3890.2	1364.4	9.15	3806.6	85.244
$\beta = 0.125$	0.02	4724.8	1530.8	11.12	4636.2	85.244
$\beta = 0.125$	0.04	5717.6	1780.8	13.46	5624.2	85.244
$\beta = 0.125$	0.08	7110.4	2348.4	16.73	6959.8	85.244
$\beta = 0.25$	0.01	3376.6	850.8	7.95	3294.8	85.446
$\beta = 0.25$	0.02	4145.0	951.0	9.75	4053.2	85.446
$\beta = 0.25$	0.04	5022.2	1085.4	11.82	4906.0	85.446
$\beta = 0.25$	0.08	6074.8	1312.8	14.30	5930.4	85.446
$\beta = 0.5$	0.01	2960.4	434.6	6.97	2846.8	85.422

Variant	Price	Accept	Extra	MAC (%)	Charged	Acc. (%)
$\beta = 0.5$	0.02	3691.0	497.0	8.69	3549.4	85.422
$\beta = 0.5$	0.04	4490.6	553.8	10.57	4338.8	85.422
$\beta = 0.5$	0.08	5430.2	668.2	12.78	5249.0	85.422
Projected to 0.25	0.01	3376.6	850.8	7.95	3294.8	85.536
Projected to 0.25	0.02	4145.0	951.0	9.75	4053.2	85.536
Projected to 0.25	0.04	5022.2	1085.4	11.82	4906.0	85.536
Projected to 0.25	0.08	6074.8	1312.8	14.30	5930.4	85.536

W.3 Fitted controllers and task continuation regret

Price	Disagreement	Heuristic accepts	Score violations	Task regret	Certificate-only
0.01	1243.8	4620.4	362.4	0.0	0.0
0.02	1518.4	5663.4	457.2	0.0	0.0
0.04	2225.4	7247.6	693.2	0.0	0.0
0.08	3182.8	9257.6	933.8	102.0	0.0

Counts average five seeds and refer to first-exit decisions. Score violations exceed the registered hindsight-regret tolerance; task regret means an incorrect prefix with at least one correct later exit. Zero violations on observed rows would not itself prove universal validity. The data-fitted confidence policy maximizes tuning-set mean confidence-minus-cost, whereas the certificate controls every admitted completion; disagreement need not imply better average task accuracy.

W.4 Post-training projection certificate

Seed	Margin certified	Same terminal label	Mean error bound	Mean actual error
3101	3628	9591	2.545	1.814
3102	2076	9656	3.689	3.035
3103	2810	9716	3.235	2.404
3104	2124	9714	3.127	2.662
3105	3059	9529	2.911	1.948

These are exposed, retrospective checks of Proposition 19, not new model selection. Every projected logit vector satisfies its error bound. The margin test is sufficient, not necessary; its missed equal-label cases measure conservatism.

W.5 Update magnitude and clipping

Variant	Stage 1 clipped (%)	Stage 2 clipped (%)	Stage 3 clipped (%)
$\beta = 0.125$	99.87	99.98	100.00
$\beta = 0.25$	99.87	99.84	99.97
$\beta = 0.5$	96.43	96.73	98.18
$\beta = 0.25$, float	99.87	99.84	99.97
Projected to 0.25	75.17	58.57	50.95

These fractions count pre-clipping update coordinates outside the bound, not inputs violating the deployed contract. Per-stage pre-clipping infinity-norm quantiles are retained for every seed in `R31_ADDITIONAL_ANALYSES.json`; the hard cap is active rather than orders of magnitude above realized updates.

X Modern verifier and complete hardware records

X.1 Same-candidate output enclosures

The fixed diagnostic uses the first 128 retained R30 timing inputs and the frozen R30 weights, not a different network for each method. The hidden-region radii below hold prefix logits fixed. Rounding

allowances are $j/128$ at future stage j, followed by the retained finite-score transfer allowance. Actual integer center traces are enclosed in every case. Floating-point verifier bounds are diagnostic and are not substituted into the deployed integer certificate.

Method	Radius	Accept	Extra	Time (ms)	Mean cap decrease
IBP	0.00	93	6	4.394	0.012951
CROWN	0.00	93	6	106.454	0.012951
α-CROWN	0.00	93	6	3099.365	0.012951
IBP-split4	0.00	93	6	8.690	0.012951
IBP	0.01	89	2	3.321	0.004744
CROWN	0.01	93	6	204.793	0.010337
α-CROWN	0.01	93	6	24303.432	0.010369
IBP-split4	0.01	89	2	7.468	0.004763
IBP	0.10	87	0	3.294	0.000000
CROWN	0.10	87	0	191.280	0.000219
α-CROWN	0.10	87	0	26094.273	0.000721
IBP-split4	0.10	87	0	7.092	0.000000

The structural cap accepts 87 inputs and takes 0.043 ms for the full 128-input batch. This timing evaluates all three caps and is distinct from the compiled single-comparison implementation. Graph construction takes 19.14 ms. Output arrays use 61,440 bytes (245,760 for splitting), while peak total process RSS is 2,109,584 KiB; the small output arrays must not be presented as total verifier memory. The exact auto_LiRPA revision is `5a098e8f9fb5786a428a024981d833d303921f2d`. The library computes IBP, CROWN, and optimized CROWN, but this experiment does not run complete branch-and-bound.

X.2 Training-frozen hardware tariffs

All three processes use network seed 3101, one PyTorch thread, and AMD EPYC 7763 CPUs. Calibration takes the first 1,000 rows of the held-out controller-fit training split. The script freezes stage prices, dispatch/setup charges, integer cutoffs, and the whole-pipeline admission decision before requesting test inputs. Setup amortization is explicitly 100,000 inputs. Dispatch and setup envelopes are empirical pilot envelopes, not worst-case execution-time proofs. Input downloading and training are offline; the steady-state timing includes prefix computation, score normalization, gate/dispatch, executed suffixes, recall updates, and output assembly.

Process	Batch	Accept	Full (ms)	Forced gate (ms)	Speed [paired 95% interval]
31001	1	0	49.151	53.850	0.913 [0.907, 0.916]
31001	32	458	26.313	26.695	0.986 [0.982, 0.989]
31001	1000	485	10.232	9.460	1.082 [1.070, 1.091]
31002	1	0	50.300	55.284	0.910 [0.903, 0.921]
31002	32	448	26.637	27.018	0.986 [0.980, 0.992]
31002	1000	485	11.225	9.811	1.144 [1.058, 1.332]
31003	1	0	49.852	54.400	0.916 [0.914, 0.921]
31003	32	427	26.617	26.900	0.989 [0.979, 0.995]
31003	1000	487	10.305	9.485	1.086 [1.079, 1.097]

Every process admits the gated path only at batch 1,000; small-batch forced-gate results are retained despite being slower. The admitted small-batch policy is the full baseline. Bootstrap resamples the fifteen paired timing repetitions within each process, with 2,000 replicates. It does not resample training seeds or identify a population of CPU architectures. Every forced-gate accepted row passes all three exact integer comparisons with the corre-

sponding frozen tariff. There are zero returned-label disagreements in all nine conditions.

Process	Batch	Gate fee (utility)	Setup (ms)	Admission	Test inputs
31001	1	0.014191	11.53	Full	128
31001	32	0.009340	11.29	Full	992
31001	1000	0.005130	11.04	Gated	1000
31002	1	0.020643	11.36	Full	128
31002	32	0.010527	10.95	Full	992
31002	1000	0.005358	11.13	Gated	1000
31003	1	0.013987	11.82	Full	128
31003	32	0.012791	11.35	Full	992
31003	1000	0.004594	10.81	Gated	1000

Exact numerator/denominator pairs for every tariff and every raw timing pair are retained in the hardware JSON files. These allow rechecking the original decision without fitting a new cost model to evaluation times.

X.3 Retrospective return diagnostics

The nested-increment return test accepts 9,314 of the 10,000 frozen MNIST rows and the strict raw-label margin test accepts 8,722. All 592 extra return accepts preserve the actual full-recall return. Among 686 contract-bad rows, 487 have unchanged return on the fully replayed trained trajectory. These are retrospective witness false alarms relative to the realized network, not claims that an online exact joint verifier has been run for every region. The archived generic product-box theorem remains exact for its stated abstraction. Scalar accepted-score regret and the distinct task-regret indicators are saved separately for all four prices.

Y Additional fitted-controller and variability tables

This is a retrospective fixed-model comparison. Controllers use only the reserved 5,000 controller-fit rows and 5,000 tuning rows; each controller freeze precedes its saved-test-trace access. Temperature scaling separately minimizes each head's NLL on $[0.25, 4]$. The ridge penalty is fixed at 10, the intercept is unpenalized, and features are standardized using controller-fit rows. Features are all 128 prefix coordinates, ten prefix probabilities, confidence, margin, and entropy. The target is the signed difference between terminal and prefix correctness. No regularization or model family is selected on test.

Seed	Price	Policy	Accuracy (%)	MACs
3101	0.01	Certificate	85.510	206626.2
3101	0.01	Ridge matched terminal	85.820	206329.4
3101	0.01	Ridge task-cost terminal	85.860	194098.9
3101	0.01	Ridge matched calibrated	85.540	206329.4
3101	0.02	Certificate	85.510	202429.2
3101	0.02	Ridge matched terminal	85.860	202068.9
3101	0.02	Ridge task-cost terminal	85.840	186664.1
3101	0.02	Ridge matched calibrated	85.560	202068.9
3101	0.04	Certificate	85.510	197665.3
3101	0.04	Ridge matched terminal	85.880	197299.6
3101	0.04	Ridge task-cost terminal	85.520	178492.7
3101	0.04	Ridge matched calibrated	85.560	197299.6
3101	0.08	Certificate	85.510	192366.1
3101	0.08	Ridge matched terminal	85.860	191481.1
3101	0.08	Ridge task-cost terminal	85.160	173077.0
3101	0.08	Ridge matched calibrated	85.550	191481.1
3102	0.01	Certificate	84.630	208491.5
3102	0.01	Ridge matched terminal	84.710	208639.9
3102	0.01	Ridge task-cost terminal	84.560	190029.1
3102	0.01	Ridge matched calibrated	84.610	208639.9

Seed	Price	Policy	Accuracy (%)	MACs
3102	0.02	Certificate	84.630	204299.9
3102	0.02	Ridge matched terminal	84.610	204533.0
3102	0.02	Ridge task-cost terminal	84.420	183256.7
3102	0.02	Ridge matched calibrated	84.540	204533.0
3102	0.04	Certificate	84.630	199620.7
3102	0.04	Ridge matched terminal	84.640	199668.4
3102	0.04	Ridge task-cost terminal	84.230	175811.4
3102	0.04	Ridge matched calibrated	84.560	199668.4
3102	0.08	Certificate	84.630	193934.6
3102	0.08	Ridge matched terminal	84.570	193802.1
3102	0.08	Ridge task-cost terminal	84.230	172515.3
3102	0.08	Ridge matched calibrated	84.540	193802.1
3103	0.01	Certificate	85.920	206271.2
3103	0.01	Ridge matched terminal	85.740	205783.6
3103	0.01	Ridge task-cost terminal	85.860	192371.4
3103	0.01	Ridge matched calibrated	85.850	205783.6
3103	0.02	Certificate	85.920	202482.2
3103	0.02	Ridge matched terminal	85.890	201724.4
3103	0.02	Ridge task-cost terminal	85.820	184957.8
3103	0.02	Ridge matched calibrated	85.850	201724.4
3103	0.04	Certificate	85.920	198131.6
3103	0.04	Ridge matched terminal	85.850	197013.5
3103	0.04	Ridge task-cost terminal	85.650	176638.0
3103	0.04	Ridge matched calibrated	85.800	197013.5
3103	0.08	Certificate	85.920	192445.5
3103	0.08	Ridge matched terminal	85.820	191141.9
3103	0.08	Ridge task-cost terminal	85.390	172409.3
3103	0.08	Ridge matched calibrated	85.780	191141.9
3104	0.01	Certificate	86.270	206647.4
3104	0.01	Ridge matched terminal	86.030	207007.7
3104	0.01	Ridge task-cost terminal	86.030	190182.8
3104	0.01	Ridge matched calibrated	86.250	207007.7
3104	0.02	Certificate	86.270	202381.5
3104	0.02	Ridge matched terminal	86.000	202832.0
3104	0.02	Ridge task-cost terminal	86.000	182053.8
3104	0.02	Ridge matched calibrated	86.220	202832.0
3104	0.04	Certificate	86.270	197522.2
3104	0.04	Ridge matched terminal	86.020	197707.6
3104	0.04	Ridge task-cost terminal	85.940	174608.4
3104	0.04	Ridge matched calibrated	86.240	197707.6
3104	0.08	Certificate	86.270	191799.0
3104	0.08	Ridge matched terminal	86.010	192069.3
3104	0.08	Ridge task-cost terminal	85.980	172260.9
3104	0.08	Ridge matched calibrated	86.200	192069.3
3105	0.01	Certificate	84.900	208417.3
3105	0.01	Ridge matched terminal	85.170	208878.4
3105	0.01	Ridge task-cost terminal	85.060	191576.5
3105	0.01	Ridge matched calibrated	85.110	208878.4
3105	0.02	Certificate	84.900	204501.2
3105	0.02	Ridge matched terminal	85.130	204925.2
3105	0.02	Ridge task-cost terminal	85.070	184878.3
3105	0.02	Ridge matched calibrated	85.060	204925.2
3105	0.04	Certificate	84.900	199912.1
3105	0.04	Ridge matched terminal	85.080	199991.6
3105	0.04	Ridge task-cost terminal	85.040	177215.6
3105	0.04	Ridge matched calibrated	85.040	199991.6
3105	0.08	Certificate	84.900	194416.8

Seed	Price	Policy	Accuracy (%)	MACs
3105	0.08	Ridge matched terminal	85.050	195036.9
3105	0.08	Ridge task-cost terminal	84.640	172594.7
3105	0.08	Ridge matched calibrated	85.000	195036.9

Y.1 Variability in certificate usefulness

Conditional Wilson intervals for each trained model are saved in its results JSON. The following table instead measures sample SD across the five models. The same official test set is reused, so seed variability is not uncertainty across new datasets. MAC-saving SD is the deterministic rescaling of acceptance SD under the registered network-work count.

Price	Accept (%)	Extra (%)	MAC saved (%)	Charged accept (%)
0.01	33.766 ± 2.025	8.508 ± 0.561	7.946 ± 0.477	32.948 ± 2.069
0.02	41.450 ± 2.041	9.510 ± 0.406	9.754 ± 0.480	40.532 ± 2.077
0.04	50.222 ± 2.113	10.854 ± 0.577	11.819 ± 0.497	49.060 ± 2.104
0.08	60.748 ± 2.117	13.128 ± 0.319	14.296 ± 0.498	59.304 ± 2.210

Y.2 Correction, spoil, and calibration

Seed	Corrected	Spoiled	Prefix NLL	Terminal NLL	Terminal temperature
3101	172	120	0.4113	0.3989	0.9379
3102	148	108	0.4437	0.4366	0.9658
3103	146	98	0.4094	0.4059	0.9746
3104	91	66	0.3986	0.3978	0.9821
3105	143	114	0.4320	0.4261	0.9662

Correction/spoil counts compare full recall to the fixed prefix on 10,000 rows. Per-head NLL, ECE, and all four calibrated temperatures are retained for every seed. Calibration changes the selection score and therefore defines a different recall policy; it does not inherit the original finite-score stopping certificate without a new bound.

Y.3 Chronology and evidence stages

We distinguish three stages of evidence. The convolutional model family and six architectural comparisons were committed before training. Hardware tariffs were subsequently calibrated on unused controller-fit inputs, with each tariff and admission choice frozen before its test-timing run. Learned controller, projection, return-witness, and modern-verifier analyses reuse frozen models and are labeled as such. Freezing a new computation before opening its local test file prevents within-run tuning on that file, but does not erase prior research exposure. Neither the repeated CPU processes nor the five models are counted as new independent datasets.

Y.4 Interpretation of usefulness intervals

An exact stopping inequality and a useful stopping frequency have different uncertainty. The former holds deterministically for the declared contract; the latter changes with the trained model and input distribution. Table 24 reports model-to-model variation instead of pooling the five evaluations into an artificial sample of 50,000 independent inputs. The models use independent initialization and minibatch seeds but a common split and common test rows. Conditional Wilson intervals for each model's acceptance and extra-acceptance frequency are retained separately, as are correction/spoil counts, per-head calibration, and paired task comparisons. These intervals describe sampling variability under their sampling interpretation; they are not additional formal certificates or uncertainty over unseen datasets.

Z PRESERVED R30 MAIN-TEXT DEVELOPMENT

The following text preserves the preceding main-text development; the current revision adds the results above.

ARCHIVED ABSTRACT

Adaptive depth requires a certificate that is informative before the skipped computation executes. We study finite-horizon neural inference with recall, charging executed depth rather than the exit whose prediction is returned. Our primary object is scalar, costed continuation regret. For networks with bounded additive logit updates, we construct an exact rectangular support interface with online work linear in the number of future exits and classes, independent of hidden-state dimension. A finite integer implementation supplies an explicit numerical contract and a uniform error bound below 2.594×10^{-12} relative to real softmax confidence. Separately, an exact return-aware box theorem certifies behavioral equivalence even when future raw labels can change; this property is never counted as scalar-regret acceptance. A single predeclared 128-dimensional MNIST candidate is evaluated only after its model and controllers are frozen. At four registered small tolerances, the scalar gate adds 1,408–1,911 accepts beyond trivial headroom on 10,000 test rows, saves 14.13–27.28% of network MACs, and retains full-recall accuracy of 94.97%. Confidence stopping has essentially the same decisions; the new contribution is their explicit certificate rather than a policy-quality advantage over confidence. We also audit all 65,536 cells of the archived two-dimensional demonstrator with outward integer arithmetic, evaluate four nested resolutions, and compare optimized linear bounds and input splitting on the same regions. The evidence distinguishes certificate actionability, task utility, numerical soundness, and measured systems cost rather than treating one as evidence for the others.

AA INTRODUCTION

Early-exit networks allocate computation to individual inputs (Graves, 2016; Teerapittayanon et al., 2016; Huang et al., 2018; Wang et al., 2018). With recall, a controller may inspect a deep exit and still return an earlier prediction. The resource cost is nevertheless the cost of all executed layers. A useful stopping certificate must therefore bound the benefit of an unseen suffix without executing it, and its own cost must fit within the saved work. These requirements separate certification from learning a good stopping policy.

The central object of this paper is *scalar continuation regret*, not prediction equivalence. We retain the exact with-recall reduction to cost-adjusted marginal utility supports, then instantiate it by an architectural contract: each new exit adds a bounded logit increment. This contract yields a closed-form confidence cap from the observed logits. It requires neither a hidden-state grid nor an ideal query returning an exact nonlinear support. The resulting gate has explicit high-confidence acceptance regions and a finite integer realization whose guarantee does not depend on roundoff in unexecuted matrix multiplications. It is a sufficient certified stopping rule for a declared objective, not a claim that confidence equals task correctness or that all neural networks should obey the same depth law.

A separate return-aware theorem handles a question that scalar regret does not answer: can full with-recall inference ever return a different label? Existing all-label invariance is sufficient but not necessary. A future exit may change its label yet never win the recall comparison. We characterize this distinction exactly for a product of logit boxes, including both class ties and earliest-exit score ties, and construct an explicit violating completion whenever equivalence fails. The distinction prevents successful label preservation from being used to disguise an uninformative scalar certificate.

The experiments test these contracts separately. The new bounded-increment candidate uses a 128-dimensional prefix and a fixed training schedule, with model and policy bytes frozen before this execution opens the official MNIST test. The earlier digits model and split remain exposed development evidence. We do not discard its unfavorable depth profile: the first exit is more accurate than its full-recall classifier. Instead we measure what the verifier actually recovers, replay every deployed certified cell, evaluate its approximation constants, and compare stronger bounds on identical regions. The archived CIFAR-100 and SVHN results are preserved, including their negative findings.

The contributions are an executable bounded-increment scalar-regret certificate with a finite arithmetic contract; an exact return-aware abstraction theorem that permits label changes; and an evaluation separating actionability, task quality, verifier refinement, and complete runtime costs. Interval propagation, CROWN-style relaxations, decision-tree lower bounds, and shortest-path dynamic programming are established tools, not separate novelty claims.

AB Related work

The public preprint *Optimal Depth of Neural Networks* (Qi, 2025) develops the finite-horizon depth formulation that motivates this analysis. Adaptive-computation architectures learn routing or halting decisions (Graves, 2016; Huang et al., 2018; Wang et al., 2018; Bolukbasi et al., 2017); recall-aware policy theory optimizes thresholds or dynamic indices under modeled loss laws (Yang et al., 2026b;a). Learned continuation values connect this problem to simulation-based and neural optimal stopping (Longstaff & Schwartz, 2001; Becker et al., 2019). Our certificate is instead evaluated at an executed prefix for a fixed network. The retained split-conformal alternative calibrates a different, statistical regret target under exchangeability, related to nested early-exit sets and compute-risk control (Jazbec et al., 2024; Wang et al., 2026).

Abstract interpretation and interval propagation supply sound enclosures (Cousot & Cousot, 1977; Gowal et al., 2018). CROWN propagates linear activation relaxations backward (Zhang et al., 2018); optimized bounds and split constraints support complete branch-and-bound verification (Xu et al., 2021; Wang et al., 2021). Venus combines dependency analysis, symbolic interval propagation, and domain splitting (Botoeva et al., 2020). Lagrangian decomposition gives anytime valid bounds (Bunel et al., 2020), and GenBaB extends branch-and-bound to general nonlinear computation graphs (Shi et al., 2024). These methods can tighten a neural abstraction, but their generic verification algorithms are not introduced here. Our comparison implements optimized backward lower slopes and four-child input splitting; it does not impersonate the full published solvers. The stopping-specific results identify which abstract scores matter, how recall changes a return-equivalence witness, and when a bounded-logit contract makes online certification independent of the hidden representation dimension.

AC Resource-correct stopping

Let $H_0, \dots, H_L$ be a fixed network trajectory and let exit j have bounded declared utility $g_j(H_j) \in [a, b]$, with width $W = b - a$. Transition costs satisfy $c_j \geq 0$,

$$C_0 = 0, \qquad C_t = \sum_{j<t} c_j, \qquad B_t = \max_{j \leq t} g_j(H_j), \qquad R_t = B_t - C_t.$$

The shallowest maximizer of B_t is returned, but all transitions through t are charged. For a prefix-adapted stopping time T, define

$$\mathfrak{P}_T = \max_{j \leq T} R_j - R_T, \quad \mathfrak{S}_T = \Big[\max_{j \geq T} R_j - R_T \Big]_+, \quad \mathfrak{G}_T = \max_j R_j - R_T.$$

Then $\mathfrak{G}_T = \max\{\mathfrak{P}_T, \mathfrak{S}_T\}$. This identity is accounting, but it isolates the suffix quantity a live certificate must control.

Lemma 22 (With-recall pathwise reduction)**.** *For every t,*

$$\mathfrak{S}_t = \Big[\max_{j>t} \{ g_j(H_j) - B_t - (C_j - C_t) \} \Big]_+ .$$

Nonnegative, not strictly positive, transition increments are sufficient.

Let $\mathcal{F}_t$ contain the executed prefix and fixed deployment constants. A verifier returns a nonempty joint set $\mathfrak{R}_t \subseteq [a, b]^{L-t}$ of future utility vectors. We assume its support maps are $\mathcal{F}_t$-measurable; a measurable compact-valued correspondence is one sufficient condition. Write

$$q^\star_{j|t} = \sup_{v \in \mathfrak{R}_t} v_j, \qquad d_{j|t} = B_t + C_j - C_t, \qquad \mathcal{J}_t = \{t+1, \dots, L\}.$$

Theorem 23 (Exact support interface)**.** *For every nonempty $\mathfrak{R}_t$,*

$$\sup_{v\in\mathfrak{R}_t} \mathfrak{S}_t(v) = U_t(q^\star) := \Big[\max_{j\in\mathcal{J}_t}\{q^\star_{j|t} - d_{j|t}\}\Big]_+.$$

Thus cross-depth coupling is irrelevant only for this fixed-prefix, worst-case, scalar, with-recall finite maximum once the exact marginal supports are fixed.

The scope qualifier is essential. Coupling can matter for expectations, CVaR, multiple metrics, sequentially updated sets, confidence allocation, and the computational problem of obtaining the supports. For a deterministic network, unrestricted suffix execution collapses the semantic set to a singleton; Theorem 23 is therefore an interface lemma, not an unconditional lower bound on computation.

AD AN EXECUTABLE BOUNDED-INCREMENT CERTIFICATE

Suppose each future exit adds a logit increment η_j satisfying the runtime contract $\|\eta_j\|_\infty \le \beta_j$. Hidden layers and exit heads may change width; the common logit coordinates are class scores. For $j > t$, define $r_{j|t} = \sum_{s=t+1}^{j} \beta_s$ and observe the current largest softmax probability p_t. The best score over the already executed prefix is B_t.

Theorem 24 (Bounded-logit continuation certificate)**.** *For temperature one, put $s_j = \exp(2r_{j|t})$ and*

$$Q_{j|t} = \frac{s_j p_t}{1 + (s_j - 1)p_t}, \qquad U_t^{\mathrm{inc}} = \left[\max_{j>t}\{Q_{j|t} - B_t - (C_j - C_t)\}\right]_+. \tag{44}$$

Then $\mathfrak{S}_t \le U_t^{\mathrm{inc}}$. Each $Q_{j|t}$ is the exact largest confidence over the structural box $[z_t - r_{j|t}, z_t + r_{j|t}]$. At the first exit, accepting $U_0^{\mathrm{inc}} \le \lambda$ gives $\mathfrak{G}_0 \le \lambda$. At any later stopping time, $\mathfrak{G}_T \le \max\{\mathfrak{P}_T, U_T^{\mathrm{inc}}\}$.

The proof sets a candidate logit to its upper endpoint and all competitors to their lower endpoints, then uses the support interface. It does not assume an exact nonlinear reachability oracle. Appendix AI gives the complete proof, varying-temperature extension, and the exact quadratic acceptance region. For example, when $B_t = p_t = p$, a future depth with $a = \lambda + C_j - C_t$ is safe exactly when

$$(s_j - 1)p^2 - (s_j - 1)(1 - a)p + a \ge 0.$$

The simultaneous condition over future depths need not coincide with trivial headroom, nor does it require invariant future labels.

The implemented numerical contract. The network emits integer prefix logits $q_0 \in [-512, 512]^{10}$ at scale 64. Each of three correction heads emits an integer update in $[-16, 16]^{10}$. Clipping, rounding, and behavior on nonfinite inputs are explicitly defined. Hence q_j lies in $q_0 + [-16j, 16j]^{10}$ regardless of any numerical error in the floating-point hidden computation. A 2,049-entry monotone exponential lookup table is bounded using rational Taylor sums and integer outward recurrences. The declared score is a ratio of integers, its largest-class endpoint cap is evaluated by table lookups, and gate comparisons use exact cross-products. The score differs uniformly from real softmax confidence by less than 2.594×10^{-12}; regret transfer adds at most twice this error. All table brackets and range constraints are independently replayable (Appendix AI.2). This is an exact arithmetic contract for the specified program, not formal verification of Python or its operating system.

Certification uses $O((L - t)K)$ arithmetic and lookups, without a grid in prefix dimension. This avoids the compilation curse for this *bounded-increment architecture*; it does not eliminate high-dimensional neural verification in general. Offline work, table loading, online checks, and rejected checks are charged. With additional verification charge Z_T, regret against the original uncharged comparator is $\mathfrak{G}_T + Z_T$, not merely $\mathfrak{G}_T$.

AE RETURN-AWARE TASK PRESERVATION AND SHARED REFINEMENT

Prediction preservation is a separate property. For future boxes $[l_j, u_j]$, let L_j be the smallest achievable maximum softmax probability and Q_{jk} the largest score achievable when class k wins.

Total price	Accept	Trivial	Extra	MAC save	Accuracy	Max regret
0.01	4,122	2,211	1,911	14.13%	94.97%	0.00331
0.02	5,559	3,751	1,808	19.05%	94.97%	0.00663
0.04	6,871	5,209	1,662	23.55%	94.97%	0.01327
0.08	7,961	6,553	1,408	27.28%	94.97%	0.02631

Table 30: Primary scalar-regret gate on all 10,000 official test rows. The tolerance equals total price divided by three. Extra accepts are relative to the same-row trivial headroom gate, not the label gate. MAC saving excludes verification overhead. Maximum regret is over accepted first-exit rows.

Both have closed forms (Appendix AJ). If the current return is (y, B), a bad full-recall return exists in the product abstraction precisely when some feasible $k \neq y$ at head j satisfies

$$Q_{jk} > \max\{B, \max_{i<j} L_i\}, \qquad Q_{jk} \geq \max_{i>j} L_i. \tag{45}$$

The strict earlier inequality and weak later inequality encode earliest-exit ties. Theorem 25 proves necessity and sufficiency and constructs a completion. Unlike all-label invariance, this rule permits a changed future label that cannot win recall. A box witness is not automatically a reachable neural witness. Return equivalence is also not a guarantee that the returned label is correct, or that no future exit could be correct.

The retained compiled verifier provides a second way to obtain boxes. We now compare its independent-logit intervals with direct margins, optimized backward linear bounds, and input splitting. These are genuine multi-head proof actions. A finite state records which future-head obligations have been discharged; pre-registered head-specific and compound actions induce a finite DAG. Shortest-path evaluation selects an attained minimum under the declared costs. The library is constructed and charged before optimization, not treated as free future information. Our implementation records different action choices on actual neural regions, rather than only a synthetic graph. This frontier is exact for its registered action library, not a new shortest-path algorithm or a claim of globally optimal wall-clock scheduling.

AF Predeclared bounded-increment evaluation

One candidate uses a $784 \to 128$ ReLU prefix and a ten-class affine exit, followed by three width-128 ReLU correction blocks with clipped logit updates. The first 50,000 of a seeded permutation of MNIST training rows fit the network; 5,000 fit continuation-value regressions; 5,000 select confidence thresholds. The prefix trains for two fixed epochs, then is frozen while the correction blocks train for five epochs. No test-driven checkpoint selection occurs. Model, score table, and controllers are hashed before this execution requests the 10,000 official test rows. The public benchmark is not claimed globally unexposed to all prior research. The retained protocol and all candidate outcomes define the evidence scope.

The first exit attains 93.68% accuracy, full recall 94.97%, and the raw terminal head 95.40%. Full recall corrects 155 shallow errors and spoils 26 initially correct rows. Thus, unlike the archived digits example, deeper inference yields a net task gain. Across the four prices, the scalar gate adds 1,408–1,911 accepts over trivial headroom, saves 14.13–27.28% of full-path network MACs, and retains 94.97% accuracy. The exact accepted-regret check has zero violations at every price.

The tuning-frozen confidence rule differs from the certified gate on only 1, 4, 1, and 2 rows, respectively, and has the same task accuracy. The certificate therefore supplies an explicit validity guarantee for an essentially confidence-threshold decision, not an empirical victory over confidence stopping. The raw terminal head is more accurate than full recall, at higher computation than certified stopping.

Comparators and objectives. All four fixed computed depths use the same shallow-tie recall rule. Raw terminal-head accuracy is reported separately from full-recall accuracy. Adjacent-depth and prefix/full randomization match the certified policy's *expected network MACs* on the same rows. A confidence rule is frozen on tuning rows, and a backward fitted continuation-value policy is trained on controller-fit rows. First-valley ASTI charges the inspected transition even when it returns an earlier prediction. We report all policies and four total prices, with primary tolerance equal to one transition

Policy	Accuracy (%)	Network MACs (k)
Certified scalar	94.970	118.213
Confidence (frozen)	94.970	118.218
First-valley ASTI	94.230	128.882
Fitted continuation	94.970	119.898
Fixed prefix	93.680	101.632
Full recall	94.970	154.624
Adjacent random (recall)	94.178	118.213
Prefix/full random (recall)	94.084	118.213
Adjacent random (raw)	94.422	118.213
Prefix/terminal random (raw)	94.218	118.213
Raw terminal head	95.400	154.624

Table 31: Same-row comparison at total price .04. Random policies match expected network MACs, not total latency. Raw-head mixtures are post-hoc accounting on frozen outputs. The frozen confidence policy essentially coincides with the certified rule; raw terminal prediction remains more accurate but costs more computation. All prices and paired intervals are retained in the appendix.

Grid	Original box	Direct margin	4-child split	Optimized linear
32	0	10	95	231
64	32	112	218	377
128	144	243	358	409
256	305	366	398	418

Table 32: Same 450 exposed R29 rows and the same parent regions. Original box counts use the independently replayed integer deployment test. Direct margin keeps shared affine-head dependence; splitting covers all four children of each parent. The two stronger columns are float64 relaxation diagnostics, not replacement integer deployment proofs. The realized all-exit-stable denominator is 422.

price. Conditional label preservation counts are never added to scalar-regret gate counts. The full table, score regrets, paired accuracy intervals, and correction/spoil counts are in Appendix AK.

Systems accounting. The timing harness randomizes and interleaves full and gated workloads, measures complete prefix/score/check/dispatch/suffix paths, and reports batch sizes 1, 32, and the complete timing workload. Table loading and verification preprocessing are separate. A second execution environment replays the frozen candidate without retraining. Network-MAC equality is not total-cost equality; all observed slowdowns remain in the runtime table. No CPU result is used as a GPU measurement. At batch 1,000, the full/gated ratios are 1.13 and 1.16 on the two CPUs; batches 1 and 32 are slower on both. Verification setup plus one table load amortizes after about 5,287 inputs on the first CPU at the measured large-batch rate, but not at the small-batch rates. The full trace, repetitions, environments, and amortization calculation accompany the results.

AG Full-table replay, refinement, and verifier comparisons

The exposed two-dimensional digits candidate has 422/450 rows with identical realized labels at all four exits. The archived table certifies 305 of these, or 72.27%, not 305 out of an unspecified attainable set. An independent outward-integer pass reconstructs every cell at nested resolutions 32, 64, 128, and 256 and verifies all 42,742 originally certified deployment cells. Exact IEEE weight and edge values are converted to dyadic interval coefficients; overflow is proved absent before fixed-width matrix arithmetic, otherwise arbitrary-precision integers are used. The complete proof arrays are retained permanently. This proves enclosure for the declared exact-dyadic suffix, not correctness of arbitrary unmodeled BLAS implementations.

The approximation constants, products of induced infinity norms through each future head, are 74.63, 419.41, and 2380.08. At grid 256, the maximal half-width is 0.11591, so the theorem's error bounds are 8.65, 48.61, and 275.87: all exceed the utility range. Refinement empirically improves certification, but the global norm bound does not predict its useful scale. The complete half-

width, slack, memory, and compile-time curves are reported rather than replacing this unfavorable quantitative finding with an asymptotic claim. Scalar gates at the original three small prices still add zero acceptances over trivial headroom at every resolution.

On the *same occupied regions*, direct class-margin propagation and optimized lower slopes improve on the original independent-logit boxes. Table 32 reports the whole four-resolution comparison. The optimized method is an in-house CROWN-style relaxation with 30 slope updates, not the complete alpha-beta-CROWN solver; the splitting method checks all four children of the same parent region. Their floating-point diagnostics are separate from the integer deployment proofs. Actual head-specific and compound actions populate the registered proof library; its budget curves and all preprocessing costs are in Appendix AL.

The original first exit attains 92.44% accuracy and strictly dominates its 90.22%-accurate full-recall policy in accuracy and network MACs. This old example remains evidence about faithful reproduction, not a competitive adaptive-depth frontier. Likewise, the fresh archived CIFAR-100 geometry equals trivial headroom on every gate-relevant row, and q25 lowers task accuracy at all five prices; the SVHN transport coverage remains 93.65% at nominal 95%. All original tables and technical constructions are preserved in Appendix AM and the subsequent historical appendices.

AH Conclusion

Resource-correct scalar regret can be certified without future execution when an architecture supplies bounded logit increments. The resulting support formula has explicit actionability regions and an exact finite-score implementation that does not require low-dimensional prefix compilation. Return-aware box certificates answer a different question and now handle nontrivial changes of future labels. These contracts preserve the separation between mathematical validity, verifier tightness, task quality, and net runtime benefit. The new controlled evaluation and the exhaustive historical replay test each layer explicitly; neither a successful theorem check nor a label-preservation rate substitutes for the declared scalar and systems results.

AI Bounded-increment certification and exact deployed arithmetic

This appendix gives the assumptions and complete proofs for the new primary construction. The neural hidden computation may change dimension, contain arbitrary affine/ReLU operations, and use different exit heads. What is required is a runtime contract on the *logits*, not a coordinate grid in the hidden representation. This is an architectural restriction and is not asserted for the archived, unconstrained ResNet.

AI.1 Structural envelopes and nonuniform prices

Fix a prefix t and a common finite class set of size K. For simplicity of notation we use one score temperature. The current logit vector is z_t and future logits obey $z_j = z_t + \sum_{s=t+1}^{j} \eta_s$, with $\|\eta_s\|_\infty \le \beta_s$. Let $r_j = \sum_{s=t+1}^{j} \beta_s$ and $D_j = C_j - C_t \ge 0$. Set $p_t = \max_k \mathrm{softmax}(z_t)_k$ and keep the actual prefix best score $B_t \ge p_t$; these need not coincide away from the first exit. Define $s_j = \exp(2r_j)$ and

$$Q_j = \frac{s_j p_t}{1 + (s_j - 1)p_t}, \qquad U_t = \left[\max_{j>t} (Q_j - B_t - D_j) \right]_+ . \tag{46}$$

If different temperatures are used, compute each $p_t^{(j)}$ at the future head's temperature and replace r_j by r_j/τ_j; the past best B_t remains in the declared utility units.

Proof of the bounded-increment certificate. Every future logit lies in $[z_t - r_j, z_t + r_j]$. A class probability is nondecreasing in its own logit and nonincreasing in every competitor. For a candidate winning class k, its maximum over this box is achieved by setting its logit to $(z_t)_k + r_j$ and every other logit to $(z_t)_l - r_j$. It equals $s_j p_{t,k}/(1 + (s_j - 1)p_{t,k})$, an increasing function of $p_{t,k}$. Maximizing

therefore selects a prefix-largest class and gives Q_j. This support is exact for the rectangular logit abstraction, not necessarily for the neural suffix. The with-recall one-depth reduction gives $\mathfrak{S}_t \le U_t$ immediately. All bounds hold simultaneously over prefixes if the increment contracts hold for every executed transition. No distributional assumption is involved. □

The bound can be informative despite possible changes in future labels. With $B_t = p_t = p$, its improvement over p is

$$Q_j - p = \frac{(s_j - 1)p(1-p)}{1 + (s_j - 1)p}.$$

For $a_j = \lambda + D_j$, the exact structural target condition is

$$(s_j - 1)p^2 - (s_j - 1)(1 - a_j)p + a_j \ge 0 \quad \text{for every } j > t. \tag{47}$$

For $s_j = 1$ it always holds when $a_j \ge 0$. Otherwise, if the discriminant $(1 - a_j)^2 - 4a_j/(s_j - 1)$ is nonpositive, it again always holds. With a positive discriminant, the high-confidence branch is

$$p \ge \frac{1 - a_j + \sqrt{(1 - a_j)^2 - 4a_j/(s_j - 1)}}{2}.$$

These are sufficient neural stopping regions with no exact-support oracle. They do not imply that training will place a useful fraction of inputs there.

AI.2 Finite integer scores and outward exponential brackets

The executable candidate uses logit integers $q \in \mathbb{Z}^K$ at scale $h = 64$. Its prefix logits are clipped and rounded into $[-512, 512]$ and each of three updates into $[-16, 16]$. Finite total maps specify behavior even for nonfinite hidden arithmetic. Therefore every reachable future logit difference and every structural-box difference is at most 1216, within the registered range $0, \dots, 2048$. The clipping contract alone proves this range; it does not require a proof of the preceding floating-point matrix multiplication.

Let $S = 2^{48}$. The alternating rational Taylor sums through orders 17 and 16 bound $e^{-1/64}$ from below and above, respectively. Multiplying them by S and rounding outward gives integers b^- and b^+. Define

$$E_0^- = E_0^+ = S, \qquad E_{n+1}^- = \left\lfloor E_n^- b^- / S \right\rfloor, \qquad E_{n+1}^+ = \left\lceil E_n^+ b^+ / S \right\rceil.$$

Induction proves $E_n^-/S \le e^{-n/64} \le E_n^+/S$ for every registered n. The declared rational utility is

$$\widetilde{g}(q) = \frac{S}{\sum_{k=1}^K E^-_{\max_l q_l - q_k}}. \tag{48}$$

It is not an unspecified floating-point softmax. All denominators, record comparisons, and certification cross-products are integer operations.

For $K = 10$, the retained table has maximum bracket width 73. Both the true and approximate normalizing sums are at least S, so

$$0 \le \widetilde{g}(q) - \max_k \operatorname{softmax}(q/64)_k \le \epsilon_E := 10(73)/2^{48} < 2.594 \times 10^{-12}.$$

This is a uniform approximation guarantee over the registered domain. It is not estimated from the test sample. A scalar regret certificate for $\widetilde{g}$ consequently upper-bounds the analogous real-softmax regret by at most an additional $2\epsilon_E$, because each compared utility differs by at most ϵ_E. Costs and the executed stopping depth are unchanged. A change in the proxy-selected return exit can affect task accuracy; task outcomes are therefore evaluated for the actually deployed rational rule.

Exact structural cap without exponentiation online. For integer radius r and any prefix-largest index k_0, define

$$d_r = S + \sum_{k \ne k_0} E^-_{q_{k_0} - q_k + 2r}, \qquad \widetilde{Q}_r = S/d_r.$$

The lower exponential table is nonincreasing. For any other possible winning index k, swapping k and k_0 pairs their own and competitor terms, and every remaining difference for k_0 is no smaller. Its

denominator is thus no larger. Within a fixed candidate class, raising that coordinate and lowering all other coordinates maximizes every difference and hence the score. This proves exactness of $\widetilde{Q}_r$ over the finite integer logit box. At the first exit, with denominator d_0, total price $p = a/b$, and three equal stage costs, the test $\widetilde{Q}_{16j} - \widetilde{g}(q) \le (j+1)p/3$ is implemented as

$$3bS(d_0 - d_{16j}) \le (j+1)a\, d_{16j} d_0.$$

Arbitrary-precision integer products avoid overflow. Integer sums use a fixed int64 type only where a worst-case magnitude bound proves safety. The runtime work is $O(K(L-t))$ table accesses/arithmetic operations, independent of hidden width and without an m^d hidden-state grid. Bit complexity depends on the registered integer precision; unit-cost arithmetic is not a wall-clock model.

AI.3 Resource charge and two distinct return semantics

If verification and dispatch incur an additional utility-price charge Z_T, the implemented reward is $B_T - C_T - Z_T$, and regret against the original uncharged sequential hindsight comparator is exactly $\mathfrak{G}_T + Z_T$. Consequently

$$\mathfrak{G}_T + Z_T \le \max\{\mathfrak{P}_T, U_T\} + Z_T.$$

A proposed first-exit stop has $\mathfrak{P}_0 = 0$. Rejected checks followed by full execution are also charged. Network-only MAC matching excludes these costs and must not be described as matching total latency.

There is a separate prediction-preservation property: a bound on future logit margins can ensure that a full *proxy-selected return* has the same label as the prefix return. This is not equivalent to a scalar-confidence regret certificate, and neither property proves correctness of the returned label. The following theorem sharpens this separate property without requiring all future raw labels to be identical.

AJ Exact return-aware certificates for rectangular logits

Let the prefix return have label y and score B. Future logit boxes are $\mathcal{Z}_j = [l_j, u_j]$. The product across future heads is an abstraction: it need not equal the jointly reachable neural set. Class ties use the smallest class index; return-score ties use the earliest exit. For each head define

$$t_j = \max_k l_{jk}, \qquad L_j = \left[\sum_k \exp(\min\{u_{jk}, t_j\} - t_j)\right]^{-1},$$

and for each class feasible as an argmax define

$$Q_{jk} = \left[1 + \sum_{r \ne k} \exp(l_{jr} - u_{jk})\right]^{-1}.$$

Class k is feasible precisely when $u_{jk} > l_{jr}$ for all $r < k$ and $u_{jk} \ge l_{jr}$ for all $r > k$.

Theorem 25 (Return-aware box completeness)**.** *A product-box completion whose full with-recall return label differs from y exists if and only if some feasible class $k \ne y$ at some future head j satisfies*

$$Q_{jk} > \max\{B, \max_{i<j} L_i\}, \qquad Q_{jk} \ge \max_{i>j} L_i. \tag{49}$$

Thus failure of every such witness certifies preservation of the return label, including cases where a future raw label may change.

Proof. First, L_j is the minimum achievable maximum softmax probability in its box. For any logits let $M = \max_k z_k \ge t_j$. Then $\sum_k e^{z_k - M} \le \sum_k e^{\min(u_{jk}, M) - M}$, a nonincreasing function of M. Choosing $z_k = \min(u_{jk}, t_j)$ attains equality at $M = t_j$. Second, the endpoint $z_k = u_{jk}, z_r = l_{jr}$ for $r \ne k$ attains Q_{jk}; the stated feasibility inequalities give exactly the prescribed argmax ties.

If a bad head j wins a completion, its confidence strictly exceeds the prefix score and the confidence of all earlier heads, and weakly exceeds all later heads. Its confidence is at most Q_{jk} and every

competing head has confidence at least L_i, establishing necessity of (49). Conversely, choose the bad endpoint at head j and the confidence-minimizing logits at every other future head. Product independence makes these choices compatible within the abstraction. The strict earlier and weak later inequalities force the winning returned head to be j, proving sufficiency and constructing a witness. The theorem is exact for the abstraction; a witness need not correspond to an actual neural trajectory. □

A strict extension of all-label invariance. For a binary example, take prefix logits $(2, 0)$ and label 1, with future singleton logits $(0, 0.2)$. The future raw label changes to 2, but its maximum softmax probability is less than the prefix score. Recall returns label 1 for every completion. All-label invariance rejects this example, whereas Theorem 25 certifies it. A costed classification continuation theorem based on the *best truly correct exit* has a different comparator and cannot be inferred from return preservation. In particular, a low-confidence future prediction may still be correct. We do not replace one comparator by the other.

Finite-score counterpart. The same argument applies to finite integer logit boxes and the rational score in (48): replace exponentials of negative integer differences by E_n^- / S. Monotonicity, endpoint feasibility, and the explicit minimizer suffice; no exponential identity is used in the minimum-score proof. This version has an exact integer decision procedure including return ties. The old floating-point model's reported return-aware counts are numerical relaxation diagnostics, not silently upgraded to this integer semantics.

AK Frozen candidate, comparator ledger, and systems measurements

AK.1 Experimental chronology and network definition

The predeclared contract is committed before the new candidate is trained. A seed-300926 permutation divides the 60,000 MNIST training rows into 50,000 model-training, 5,000 controller-fit, and 5,000 tuning rows. The 784-to-128 ReLU prefix and affine head train for two epochs with Adam at learning rate .001 and batch size 256. Their parameters are then frozen. Three width-128 ReLU hidden blocks and affine correction heads train for five fixed epochs with the mean of three cross-entropies. Each correction is clipped to $[-.25, .25]$. No checkpoint is chosen by evaluation performance.

Deployment exports the weights, clips and rounds the prefix logits at scale 64, and rounds each clipped correction to an integer in $[-16, 16]$. The architecture is a bounded logit accumulator, not a claim that arbitrary unconstrained networks satisfy the same contract. The model, rational score table, fit-only value controllers, and tuning-only confidence thresholds are hashed before the official test files are requested in this run. Public benchmark exposure elsewhere is not ruled out. All four registered prices and all registered dimensional diagnostics are reported.

The test has 10,000 rows. Full recall chooses the shallowest largest rational score among computed exits. Its accuracy is distinct from the raw terminal head's accuracy. The code retains both names explicitly. The scalar gate uses only prefix logits and fixed constants. Full trajectories appear only in evaluation traces and in retrospective diagnostics, never in its decision.

AK.2 Same-row policy comparisons

The fitted-value comparator uses backward ridge regressions on the controller fit split. At each depth, its features are the observed hidden vector, current integer logits divided by 64, and best-so-far rational score. Standardization and coefficients are frozen. Its continuation target subtracts one transition price. The confidence threshold is fixed from the tuning denominator quantile matching the certified gate's tuning acceptance fraction. These choices do not use test labels.

Every fixed-depth policy in the next four tables uses with-recall selection. The additional raw-head rows distinguish terminal classification from recall. Raw-head randomized comparisons are a post-hoc accounting addition using unchanged model outputs, not another candidate selection. Randomized policies are evaluated in expectation, analytically mixing correctness and rewards; their expected

Policy	Acc. (%)	MACs (k)	Score regret	Δacc. CI (pp)
Certified scalar	94.970	132.781	0.00077	–
Confidence (frozen)	94.970	132.786	0.00077	[+0.00, +0.00]
First-valley ASTI	94.230	138.156	0.00371	[-0.95, -0.55]
Fitted continuation	94.970	143.158	0.00179	[+0.00, +0.00]
Fixed 0 (recall)	93.680	101.632	0.03347	[-1.55, -1.03]
Fixed 1 (recall)	94.210	119.296	0.02016	[-0.98, -0.56]
Fixed 2 (recall)	94.540	136.960	0.01080	[-0.59, -0.29]
Fixed 3 (recall)	94.970	154.624	0.00291	[+0.00, +0.00]
Adjacent mix (recall)	94.462	132.781	0.01301	[-0.66, -0.36]
Prefix/full mix (recall)	94.438	132.781	0.01551	[-0.64, -0.42]
Adjacent mix (raw)	94.905	132.781	–	[-0.16, +0.03]
Prefix/terminal mix (raw)	94.691	132.781	–	[-0.43, -0.13]

Table 33: Complete comparator ledger at total price 0.01, tolerance 0.00333. Raw-head-only comparators do not share the recall-score reward definition, hence their regret column is left unassigned.

Policy	Acc. (%)	MACs (k)	Score regret	Δacc. CI (pp)
Certified scalar	94.970	125.166	0.00144	–
Confidence (frozen)	94.970	125.187	0.00143	[+0.00, +0.00]
First-valley ASTI	94.230	133.332	0.00516	[-0.95, -0.55]
Fitted continuation	94.970	133.745	0.00303	[+0.00, +0.00]
Fixed 0 (recall)	93.680	101.632	0.02891	[-1.55, -1.03]
Fixed 1 (recall)	94.210	119.296	0.01892	[-0.98, -0.56]
Fixed 2 (recall)	94.540	136.960	0.01289	[-0.59, -0.29]
Fixed 3 (recall)	94.970	154.624	0.00834	[+0.00, +0.00]
Adjacent mix (recall)	94.320	125.166	0.01692	[-0.83, -0.48]
Prefix/full mix (recall)	94.253	125.166	0.01977	[-0.86, -0.57]
Adjacent mix (raw)	94.659	125.166	–	[-0.44, -0.19]
Prefix/terminal mix (raw)	94.444	125.166	–	[-0.69, -0.36]

Table 34: Complete comparator ledger at total price 0.02, tolerance 0.00667. Raw-head-only comparators do not share the recall-score reward definition, hence their regret column is left unassigned.

network MACs match the certified policy. They are not claimed to match table-lookup, dispatch, or total latency costs. Paired 95% intervals are descriptive row-bootstrap intervals for the accuracy difference (comparator minus certified), without a familywise or post-selection claim.

Raw fixed-head accuracies at depths 0–3 are 93.68%, 94.47%, 95.04%, 95.40%. The structural label-only gate accepts 8722 rows, but these are not added to the scalar acceptance counts.

AK.3 Prefix dimension and storage

The registered 4/8/16/32-dimensional digits candidates are development measurements, not replications of the MNIST test. They use one last-epoch candidate each, with 40 prefix epochs and 40 suffix epochs. The smallest candidate is inaccurate and all four scalar gates are inactive at the registered small prices; these outcomes are retained rather than tuned away. They establish finite storage and executable costs across widths, not useful small-budget certification at every width. The MNIST candidate has dimension 128 and a separate training/test protocol. This is an architectural alternative to an exponential grid, not evidence that the old grid itself scales.

AK.4 Randomized/interleaved timing and independent hardware replay

Both methods process the same retained timing inputs. Invocation order is randomized independently in each repetition and interleaved; batch one uses 7 measured rounds after warmup and the other MNIST batches use 21. Time is reported for the whole 1,000-row workload, not per example. The complete pipeline includes prefix evaluation, score normalization, gate comparisons, Boolean dispatch, any executed correction blocks, and recall selection. No GPU was measured. The first platform is the remote experiment runner; the second replays the exact frozen model and timing

Policy	Acc. (%)	MACs (k)	Score regret	Δacc. CI (pp)
Certified scalar	94.970	118.213	0.00260	–
Confidence (frozen)	94.970	118.218	0.00260	[+0.00, +0.00]
First-valley ASTI	94.230	128.882	0.00916	[-0.95, -0.55]
Fitted continuation	94.970	119.898	0.00313	[+0.00, +0.00]
Fixed 0 (recall)	93.680	101.632	0.02246	[-1.55, -1.03]
Fixed 1 (recall)	94.210	119.296	0.01914	[-0.98, -0.56]
Fixed 2 (recall)	94.540	136.960	0.01978	[-0.59, -0.29]
Fixed 3 (recall)	94.970	154.624	0.02190	[+0.00, +0.00]
Adjacent mix (recall)	94.178	118.213	0.01935	[-1.01, -0.59]
Prefix/full mix (recall)	94.084	118.213	0.02229	[-1.07, -0.71]
Adjacent mix (raw)	94.422	118.213	–	[-0.73, -0.38]
Prefix/terminal mix (raw)	94.218	118.213	–	[-0.94, -0.56]

Table 35: Complete comparator ledger at total price 0.04, tolerance 0.01333. Raw-head-only comparators do not share the recall-score reward definition, hence their regret column is left unassigned.

Policy	Acc. (%)	MACs (k)	Score regret	Δacc. CI (pp)
Certified scalar	94.970	112.437	0.00450	–
Confidence (frozen)	94.970	112.448	0.00450	[+0.00, +0.00]
First-valley ASTI	94.210	125.014	0.01871	[-0.97, -0.56]
Fitted continuation	94.960	110.494	0.00484	[-0.03, +0.00]
Fixed 0 (recall)	93.680	101.632	0.01424	[-1.55, -1.03]
Fixed 1 (recall)	94.210	119.296	0.02425	[-0.98, -0.56]
Fixed 2 (recall)	94.540	136.960	0.03823	[-0.59, -0.29]
Fixed 3 (recall)	94.970	154.624	0.05367	[+0.00, +0.00]
Adjacent mix (recall)	94.004	112.437	0.02037	[-1.19, -0.75]
Prefix/full mix (recall)	93.943	112.437	0.02228	[-1.23, -0.82]
Adjacent mix (raw)	94.163	112.437	–	[-1.00, -0.63]
Prefix/terminal mix (raw)	94.031	112.437	–	[-1.15, -0.73]

Table 36: Complete comparator ledger at total price 0.08, tolerance 0.02667. Raw-head-only comparators do not share the recall-score reward definition, hence their regret column is left unassigned.

inputs locally. Different cloud CPUs and software runtimes are not a controlled hardware causal study.

The first platform's verification setup is 2.446 ms and median repeated load/decompression is 0.182 ms. Including one such load raises large-batch amortization to about 5,287 inputs on that platform. Model training is common to full and gated inference and is separately logged.

Independent fresh process launches on the second platform use the same frozen model and inputs. Their large-batch ratios are 1.180, 1.173, 1.153. These are repeated processes, not additional training seeds. The filesystem cache is uncontrolled.

AL Exhaustive archived grid audit and actual proof-action library

AL.1 All-cell arithmetic replay

The old 256-by-256 edges and frozen dyadic weights define each cell exactly. For every affine layer, weight endpoints are obtained by exact rational floor and ceiling at scale 2^{24}. For an interval coefficient $w \in [w_-, w_- + e]$ and an interval input $x \in [x_-, x_+]$, the residual contribution is enclosed using $e \min(x_-, 0)$ and $e \max(x_+, 0)$. Signed affine products are evaluated with integer matrix arithmetic; a Python-integer absolute-sum bound proves no overflow before int64 is used, with arbitrary precision as fallback. Division and bias addition round outward. ReLU then clips both endpoints at zero. This covers entire cells, not just the observed trajectories. All 42,742 labels in the original 65,536-cell deployment table are independently verified by the resulting integer bounds. Closed cell boundaries are included.

Data	Width	First %	Recall %	Accept	Setup ms	Time ratio
DIGITS	4	14.00	17.56	0	2.204	0.398
DIGITS	8	67.11	72.89	0	2.192	0.423
DIGITS	16	85.56	89.56	0	2.295	0.447
DIGITS	32	86.22	89.78	0	2.225	0.505
MNIST	128	93.68	94.97	6871	2.446	1.130

Table 37: All registered widths at total price .04. Setup is verification-score-table construction/serialization, not shared model training. The online lower table is 16,392 raw bytes and the two-sided arithmetic proof table is 32,784 bytes for every width. Time ratio is full/gated for the complete registered timing batch (450 digits or 1,000 MNIST rows); below one means slower.

Platform	Batch	Full ms	Gated ms	Ratio	Amortize N
Remote 9V45	1	85.047	127.702	0.666	–
Remote 9V45	32	7.217	8.532	0.846	–
Remote 9V45	1000	4.307	3.810	1.130	4,921
Replay 9V74	1	123.248	196.864	0.626	–
Replay 9V74	32	11.995	14.273	0.840	–
Replay 9V74	1000	8.307	7.139	1.164	2,093

Table 38: Complete MNIST pipeline at total price .04. Amortization divides the frozen verification preprocessing time by positive per-input time saved; no finite break-even is reported for slowdowns. Repeated table loading is reported separately and is not an OS-cache-flushed cold benchmark.

Nested child boxes are intersected with their ancestor boxes. No evaluation row selects the resolution sequence. The preserved old data are nevertheless exposed: this is a diagnostic reanalysis. Exact-real suffix semantics and floating deployment semantics are distinguished. The new bounded-increment program has its own integer range contract and does not inherit an unproved BLAS error assumption from this historical reconstruction.

AL.2 Backward linear and splitting implementation

For each future head and each of nine competitors to the observed prefix label, direct affine margins retain dependence between head logits. The optimized baseline propagates these linear objectives backward through ReLU relaxations. On $l < 0 < u$, an upper line has slope $u/(u-l)$ and intercept $-lu/(u-l)$; a lower line has a slope in $[0, 1]$. Lower slopes are independently optimized for 30 Adam updates at learning rate .1, retaining the best valid bound per objective. This is a CROWN-style implementation, not the full alpha-beta-CROWN solver or a VNN-COMP comparison. Float64 baseline margins must exceed 10^{-9} to count; their numbers are descriptive under the analytical relaxation, not the independent integer deployment proof.

The splitting baseline quarters each original two-dimensional parent cell, propagates all children, and takes the worst margin over those children. Every child is included, so it certifies the same parent region rather than silently replacing the problem by one outcome-selected subcell. The initial IBP margins are retained whenever they are tighter.

The registered library contains the shared direct-margin pass, one optimized action for each future head, and one compound four-child action. A three-bit state records discharged head obligations. All action effects are computed before planning. Costs are the measured whole-population action times divided by 450, a declared work normalization rather than a promise of independently additive hardware latency. The library construction cost is charged separately; planning cannot claim that those answers were obtained for free.

The original 450-row label pipeline is also retimed without retraining in three fresh local processes, with randomized/interleaved full/gated order. Large-batch ratios are 1.724, 1.606, 1.612. Small-batch measurements and all raw timings are retained, without replacing the original R29 timing evidence.

Grid	$\delta_{\max}$	Compile s	Proof MB	Deploy kB	Stable recall %
32	0.927253	0.092	0.494	2.576	0.00
64	0.463627	0.158	1.974	9.232	7.58
128	0.231813	0.620	7.897	34.832	34.12
256	0.115907	2.560	31.588	135.184	72.27

Table 39: Four-resolution full-grid replay. Stable recall divides accepted rows by the 422 realized all-exit-stable rows. Raw storage uses decimal MB/kB; compressed proof files are retained in the repository.

Grid	Head 1 bound	Head 2 bound	Head 3 bound	Return-aware
32	69.197	388.897	2206.934	2
64	34.599	194.449	1103.467	64
128	17.299	97.224	551.733	206
256	8.650	48.612	275.867	356

Table 40: Worst-case norm bounds and a separate return-aware box diagnostic. All quantitative norm bounds exceed the utility range; they do not explain useful empirical tightness. Return-aware values use the old float64 score semantics and a conservative comparison guard, not the new integer-score deployment.

AL.3 What is not established

The new architecture and the stronger occupied-cell baselines are not a replay of the original R23 ResNet/CIFAR-100 network. No unsupported bounded-logit contract is inferred for that network. No complete alpha-beta-CROWN/Venus benchmark, GPU measurement, cache-flushed cold-start experiment, multi-seed training study, or globally untouched-public-dataset claim is made. The preserved negative CIFAR-100 and SVHN evidence is unchanged. These distinctions specify the experiments actually performed rather than turning missing work into a positive result.

AM Preserved methods and historical audit details

The following retained methods and empirical record preserve the technical content of the preceding revision. The new primary bounded-increment experiment is reported separately.

AM.1 Compiled prefix verification

We now construct a neural verifier rather than assuming an exact-support query. Freeze a finite partition of a bounded prefix-state region into axis-aligned cells. For each cell A, an offline pass propagates an interval through the actual suffix, reusing each intermediate enclosure for every downstream exit. An input outside the region uses the unchanged fallback. No label or future activation is used to locate the deployment prefix in its cell.

For a layer $h' = \mathrm{ReLU}(Mh + v)$ and an enclosure $h \in [l, u]$, write $M^+ = \max(M, 0)$ and $M^- = \min(M, 0)$. The shared pass uses

$$l' = \mathrm{ReLU}(M^+ l + M^- u + v), \qquad u' = \mathrm{ReLU}(M^+ u + M^- l + v). \tag{50}$$

An affine head is propagated without the final ReLU. This accommodates changing dimensions and unrelated heads. Let $[\underline{z}_{j,A}, \bar{z}_{j,A}]$ be the resulting logit box. Store

$$Q_j(A) = \max_k \left[1 + \sum_{r \neq k} \exp\{(\underline{z}_{j,A,r} - \bar{z}_{j,A,k})/\tau_j\} \right]^{-1}. \tag{51}$$

This is the *exact support of the rectangular logit abstraction*, and a possibly loose upper bound on neural reachable utility. It is not an exact support of the nonlinear reachable set. The distinction is algorithmic: all quantities in (50)–(51) are computed and stored, and deployment pays for lookup and reduction, not for an ideal oracle.

Grid	Shared IBP s	Linear heads s	Split s	Library accepts	Plan ms
32	0.0125	1.5087	0.0414	231	2.496
64	0.0104	1.5234	0.0214	377	2.627
128	0.0060	1.3552	0.0211	409	2.596
256	0.0052	1.2306	0.0223	418	2.588

Table 41: Actual occupied-region proof library. The one shared pass and all candidate action computations are paid to construct the library. A shortest path selects among registered summaries; a positive path count is not a net online speedup claim. Exact choices and budget curves are retained in JSON.

Grid	Menu budget	Incremental population s	Certified rows
32	0	0.00000	10
32	1	0.04135	95
32	2	0.60843	132
32	3	1.50872	231
32	4	1.55007	231
64	0	0.00000	112
64	1	0.02138	218
64	2	0.55997	235
64	3	1.52337	377
64	4	1.54476	377
128	0	0.00000	243
128	1	0.02113	358
128	2	0.57204	368
128	3	1.35522	409
128	4	1.37635	409
256	0	0.00000	366
256	1	0.02229	398
256	2	0.55708	402
256	3	1.23063	418
256	4	1.25291	418

Table 42: Attained registered neural proof-library cost–certificate frontier. Budgets are 0, the compound splitting cost, the largest singleton cost, the sum of singleton costs, and the sum of all costs. They are not necessarily sorted. Initial direct-margin construction is charged separately and all action effects have already been computed. No unknown prospective action is free.

Theorem 26 (Constructive approximation and stopping)**.** *Consider a finite ReLU suffix with affine heads, positive temperatures, and a cell A containing H_t, of infinity-norm half-width at most δ. Let M_j^{lip} be the product of induced infinity norms from the prefix through head j. In exact arithmetic the compiled caps are sound and*

$$0 \le U_t(Q(A)) - \mathfrak{S}_t \le \max_{j>t} \frac{M_j^{\mathrm{lip}}\delta}{\tau_j}. \tag{52}$$

For nested cells, intersecting every new logit box with its ancestors makes caps monotone. Refinement therefore certifies any strict-margin target $\mathfrak{S}_t \le \lambda - \eta$ once the right-hand side is at most η. No finite-resolution guarantee is asserted at a zero-margin boundary.

The rate is constructive but may be conservative. A uniform grid with m intervals per coordinate uses m^{d_t} cells; high-dimensional prefixes can make preprocessing prohibitive. The experiment tests a learned two-dimensional prefix rather than hiding this dependence. Preprocessing, proof storage, and online memory are reported separately.

A label-free gate with exact task preservation. Store a label k_A only if, at *every* future exit, $\underline{z}_{j,A,k_A} > \max_{r\neq k_A} \bar{z}_{j,A,r}$; otherwise store an abstention. At deployment accept only when k_A equals the observed prefix prediction. The label table alone is sufficient online; softmax and future heads are not executed on accepted inputs.

CPU process	Seed	Batch 1	Batch 32	Batch 450	Load ms
Local	940	0.917	0.840	1.724	3.170
Local	941	0.904	0.862	1.606	2.826
Local	942	0.912	0.840	1.612	3.053
Remote	940	0.823	0.781	1.628	4.177
Remote	941	0.815	0.773	1.598	4.164
Remote	942	0.808	0.769	1.532	4.160

Table 43: Read-only replay of the original R29 label pipeline. Each ratio is full/compiled time for the same 450-row workload with randomized/interleaved order. Each process opens model and table afresh; the operating-system cache is not flushed. These measurements supplement rather than overwrite R29 timings.

Theorem 27 (Certified optimal continuation depth)**.** *At the first available exit, the label gate implies that all future predictions equal the observed prediction. For every unknown true class and nonnegative transition costs, stopping immediately has zero task continuation regret and is the shallowest task-optimal depth along the declared sequential network path. The gate preserves the returned prediction of full with-recall inference row by row, regardless of which stored exit that inference returns.*

This is a conditional, certified optimal-depth result, not a statistically optimal architecture or learned-policy claim. At later exits the result controls continuation, not an already incurred prefix regret. Verification charges must still be paid, as specified next.

AM.2 Shared work, attainability, and neural witnesses

For deployment volume N, charge the amortized preprocessing cost P/N, online abstraction/lookup cost $A(x)$, any refinement cost $V(x)$, and controller and dispatch cost $D(x)$. In one common resource unit, net saving relative to fallback is

$$C_{T_{\mathrm{fb}}}(x) - C_T(x) - A(x) - V(x) - D(x) - P/N. \tag{53}$$

All attempted verifications, including rejected gates, enter this ledger. If their total utility-price charge is Z_T, the global regret against an uncharged sequential hindsight comparator is exactly $\mathfrak{G}_T + Z_T$, hence at most $\max\{\mathfrak{P}_T, U_T(Q)\} + Z_T$. MAC counts and seconds are not interchanged.

A shared interval pass is one compound action returning a vector of caps. More generally, consider a *finite registered acyclic proof graph*: a state records an enclosure and which consecutive depth obligations have been discharged; an edge supplies a sound multi-depth summary and charges its incremental work once. All edge effects and costs must already be in the registered library, and its construction and planning are charged. A path cannot silently reuse work absent from its state.

Theorem 28 (Attained shared-proof frontier)**.** *For fixed λ, retain edges whose newly discharged caps satisfy $Q_{e,j} - d_j \le \lambda$. Among complete registered proof paths, the minimum cost is attained and is obtained by the shortest-path recurrence*

$$F(v) = \min_{e=(v,w)\ \text{retained}} \{k_e + F(w)\}, \qquad F(v_{\mathrm{term}}) = 0.$$

An absent path has value $+\infty$. Evaluation takes $O(|V| + |E|)$ after cap checks. The smallest certificate at a finite budget is attained whenever a budget-feasible complete path exists.

The graph model includes the compiled table as one all-future summary edge, and ordered shared propagation as a chain. It is not a polynomial algorithm for arbitrary overlapping cache schedules: enclosures and cache states must be expanded explicitly, possibly exponentially. Dynamic programming itself is standard; its role is to remove independent-coordinate accounting from the implemented verifier. The separable baseline, retained in the appendix, now uses finite nonempty menus. General menus give only an infimal cost, not budget attainability.

Actual neural endpoint witnesses. For completeness, the exact-coordinate baseline can be realized by a width-two ReLU family. Start at $H_0 = (1, 0)$ and let

$$H_j = \mathrm{ReLU}\left(\begin{pmatrix} 1 & 0 \\ \theta_j & 0 \end{pmatrix} H_{j-1}\right), \qquad z_j = ((\log 4) H_{j,2}, 0), \quad \theta_j \in \{0, 1\}.$$

Then $H_j = (1, \theta_j)$ and max-softmax utility is $1/2$ or $4/5$ independently at each depth. In a parameter-probe model charging each read of θ_j, omitting any cost-relevant parameter leaves an explicit safe all-zero network and a violating one-bit network with the same read transcript. This supplies neural witnesses, but the resulting lower bound is the familiar OR certificate complexity (Buhrman & de Wolf, 2002), not a new unrestricted circuit lower bound. The implemented table does not claim to operate in that product oracle.

AM.3 Residual-box comparator

For the theorem-aligned audit, future states remain in one coordinate system:

$$H_{j+1} = \mathrm{ReLU}(H_j + r_j(H_j)), \qquad \|r_j(h)\| \le \ell_j \|h\| + \rho_j,$$

with shared affine head $z(h) = Wh + b$, K classes, and frozen depth-specific temperatures τ_j. From the observed H_t, the standard residual recursion produces radii $R_{j|t}$ satisfying $\|H_j - H_t\| \le R_{j|t}$. With $z_t = WH_t + b$, define

$$a^-_{j,k} = \exp\left(\frac{(z_t)_k - \|w_k\| R_{j|t}}{\tau_j}\right), \qquad a^+_{j,k} = \exp\left(\frac{(z_t)_k + \|w_k\| R_{j|t}}{\tau_j}\right).$$

A sound confidence cap is

$$Q_{j|t} = \max_k \frac{a^+_{j,k}}{a^+_{j,k} + \sum_{r \ne k} a^-_{j,r}}.$$

Proposition 29 (Linear-in-class implementation)**.** *Given z_t, row norms, residual constants, costs, and temperatures, all future box caps and $U_t(Q)$ can be streamed in $O((L-t)K)$ scalar operations and $O(K)$ working memory. Storing all caps uses $O(K + L - t)$ memory. The pairwise-margin refinement requires $O((L-t)K^2)$ work in the direct implementation.*

At each depth, compute $S^-_j = \sum_k a^-_{j,k}$ once and evaluate class k with denominator $a^+_{j,k} + S^-_j - a^-_{j,k}$. This avoids a hidden K^2 loop. Stage-changing spaces and depth-specific heads remain covered by Theorem 23 whenever a prefix-measurable verified set $\mathcal{H}_{j|t}$ and support $\sup_{h \in \mathcal{H}_{j|t}} g_j(h)$ are available; only this concrete radius construction requires shared coordinates/head.

AM.4 Policies and statistical alternative

The first-valley proposal continues while the observed best-so-far increment exceeds its transition price and stops after the first failed inspection. Its prefix regret is the failed-inspection amount $\iota_T = [c_{T-1} - (B_T - B_{T-1})]_+ \le c_{T-1}$. If a prefix-only certificate satisfies $\widehat{U}_T \le \lambda$, then

$$\mathfrak{G}_T \le \max\{\iota_T, \lambda\}.$$

A fallback guarantee can be composed by another maximum. These facts are policy-accounting consequences; they do not assert that the current verifier will accept.

The architecture-general baseline freezes a stopping rule, prefix feature map, bounded target, predictor, and level before split-conformal calibration. The one-sided residual order statistic gives marginal coverage for the frozen tuple. It is not policy-uniform after tuning, and repository chronology checks cannot establish exchangeability. The direct task-regret target uses labels only during fit/calibration, never as deployment features. Label-free deterministic task stability is also possible when verified future logits force every future predicted label into the already observed label set; The compiled label gate in Theorem 27 implements this condition on the new demonstrator; its frequency is not inferred for the ResNet audit.

AM.5 Frozen audit and break-even accounting

The confirmatory chain uses a fixed ImageNet-1K ResNet-34 backbone, one shared CIFAR-100 affine head, disjoint model-training/policy-fit/policy-tune/conformal splits, and a once-opened official test. The deepest audited shared-head exit has 58.20% accuracy; this is a theorem-aligned test bed, not a competitive early-exit architecture claim.

Price	CCI	Per-price	Line DP	CCI–PP 95% CI	CERT accepts
.01	.00720	.00415	.00395	[.00277,.00332]	0
.02	.00973	.00762	.00733	[.00181,.00241]	0
.04	.01470	.01401	.01355	[.00042,.00096]	0
.08	.02457	.02433	.02365	[.00004,.00044]	0
.16	.04338	.04041	.03931	[.00222,.00372]	0

Table 44: Frozen official CIFAR-100 audit. Regret is mean proxy Bellman regret (lower is better). Intervals are pointwise paired intervals. The predeclared $\lambda = c_{\max}$ gate accepts no override.

Price	$\lambda_{25}/c_{\max}$	Accept	MAC save	Break-even MAC	Δproxy regret	Δtask acc.
.01	69.8	23.47%	5.26%	12.97M	+.00264	-2.03 pp
.02	37.9	22.55%	5.02%	12.39M	+.00205	-2.28 pp
.04	22.5	20.88%	4.55%	11.22M	+.00136	-2.70 pp
.08	14.3	19.35%	4.20%	10.37M	-.00070	-2.85 pp
.16	9.9	12.32%	2.62%	6.46M	-.00184	-2.18 pp

Table 45: Frozen q25 gate relative to the same-population zero-accept $c_{\max}$ fallback. "Break-even MAC" is the maximum combined verifier/controller overhead compatible with a net MAC saving under the audited 246.765568M terminal-path convention; it is not a latency measurement.

The fresh unused-split audit is more diagnostic. On every gate-relevant row, the residual-box certificate equals trivial $[0, 1]$ utility headroom. Thus q25 acceptance is caused by a relaxed tolerance, not better geometry.

The complete twenty-point tolerance family remains archived. We do not manufacture a fixed-depth or confidence baseline by mixing populations: the R23 artifact lacks the per-example outcomes required to reconstruct a same-row, exactly compute-matched comparator. Appendix AR therefore specifies the frozen matching rule, tie-breaking, uncertainty interval, and required row ledger for the next execution. Present q25 results are evidence about safety/actionability and verifier saturation, not an algorithmic-quality victory.

Historical evidence is stated at its actual resolution. Rounded aggregate archives imply strict box gain above 10^{-6} on at least $5/297$ rows; they do not recover which rows. The R24 SVHN transport audit covers 93.65% (7,492/8,000) at nominal 95%, with continuous-residual split-conformal lower-tail probability 6.09×10^{-4}. A later same-source, outcome-fresh but post-hoc-designed diagnostic covers 94.84%. We retain both rather than replacing the adverse official result. Repository CI checks artifact hashes, arithmetic, theorem markers, and compilation; it does not prove exchangeability, semantic assumptions, or hardware latency.

AM.6 Executed compiled-verifier demonstrator

We trained a four-exit digits network with widths $64 \to 32 \to 2 \to 64 \to 32 \to 16$ and separate ten-class heads after widths $2, 64, 32, 16$. The model-training, validation and evaluation sizes are $1077, 270, 450$. The 256^2 grid bounds use training/validation prefixes only. This is *nonconfirmatory development evidence*: grid-64 prototype outcomes were inspected, and neither the dataset nor the final evaluation split is claimed untouched. It is not a replacement for the fresh ResNet audit.

The executed label gate stops on 305/450 examples and preserves all full-recall predictions (accuracy 90.22%). The network-only MAC reduction is 43.45%; the online label table occupies 135,184 bytes. Table 46 reports same-row alternatives, including randomized truncation analytically averaged over its routing coin and matched to the hybrid's mean network MACs. The random mixtures use compute, not correctness, to choose their mixing weight. Their uncertainty intervals, every fixed depth, ASTI, and active-depth counts are in Appendix AS.

The matched randomized baselines have higher task accuracy; even the first exit is stronger than full recall here. Thus certificate-induced preservation of a given network is demonstrated, not superiority of its accuracy–compute frontier. The proxy-cap gate at $\lambda = c$ adds 0, 0, 0 acceptances over trivial headroom at prices 0.01, 0.02, 0.04, respectively. The much larger label-stability population uses a different target. These are different utility contracts.

Method	Mean depth	Network MACs	Accuracy (%)
First exit	0.000	2132.0	92.44
Full recall	3.000	5940.0	90.22
Compiled label gate	0.967	3359.0	90.22
Validation-frozen confidence	1.047	3460.6	90.44
Matched adjacent random	1.194	3359.0	91.25
Matched prefix/full random	0.967	3359.0	91.73

Table 46: Same 450 rows; all returns use recall. Randomized rows are exact conditional expectations over the routing coin, not a fabricated random run. Matching is in network MACs, excluding method-specific dispatch overhead.

Batch	Full path (ms)	Compiled path (ms)	Speed ratio
1	20.606	23.393	0.88
32	1.466	1.647	0.89
450	0.730	0.420	1.74

Table 47: Median time for the complete 450-example workload on one CPU thread, including prefix, heads, lookup, dispatch and executed suffix. Ratio is full/compiled; below one is a slowdown. All tested batch sizes are shown.

Verification preprocessing, including proof serialization, costs 2.329 seconds; the shared interval compilation component costs 1.136 seconds. At batch 450, the measured timing difference amortizes all verification preprocessing after approximately 3.38 million inputs. At any batch showing a slowdown there is no positive amortization point. These CPU measurements neither establish the old ResNet verifier's cost nor turn its 6.46–12.97M-MAC ceilings into a demonstrated speedup.

AN CONSTRUCTIVE NEURAL VERIFICATION: PROOFS AND COST CONTRACT

AN.1 SOUNDNESS AND THE APPROXIMATION RATE

Induct on layers in (50). Positive coefficients multiply lower endpoints for the lower affine bound and negative coefficients multiply upper endpoints; the reverse gives the upper bound. Coordinatewise ReLU is monotone. Consequently every state reached from any $h \in A$ lies in the propagated interval. The same argument applies to each affine head, even with changing representation dimensions. Max-softmax for a fixed target class increases in that class's logit and decreases in all competing logits. Its supremum over a rectangular logit set is therefore the endpoint expression (51); the finite maximum over target classes commutes with this supremum. No claim of joint attainability by the actual network is used.

To bound the relaxation, let the initial box have coordinate half-width at most δ. An affine interval has half-width at most $|M|r$, and clipping by ReLU cannot increase its coordinate width. Thus the head-j logit box has infinity half-width at most $M_j^{\mathrm{lip}}\delta$. Any two vectors in this box differ by at most $2M_j^{\mathrm{lip}}\delta$ in infinity norm. For a softmax coordinate p_k with temperature τ_j,

$$\|\nabla p_k\|_1 = 2p_k(1-p_k)/\tau_j \le 1/(2\tau_j).$$

The maximum of finitely many functions with this Lipschitz constant has the same constant. Comparing the endpoint maximizer with the realized logit vector yields

$$0 \le Q_j(A) - g_j(H_j) \le M_j^{\mathrm{lip}}\delta/\tau_j.$$

Taking cost-adjusted finite maxima, followed by the positive part, proves Theorem 26. Intersection with ancestor enclosures preserves soundness and nesting, hence monotonicity of the support. A strictly safe instance is certified after finitely many refinements whose cell diameters converge to zero. This is relative completeness under a strict margin, not a claim that any fixed library certifies every safe input.

The construction computes increasingly tight upper supports using explicit neural matrix operations; it never requires a nonlinear exact-support oracle. The offline work per cell is bounded by a constant

number of affine passes through the suffix plus head boxing. If layer widths are d_j and head sizes are K_j, its arithmetic is

$$O\left(\sum_{j>t} d_j d_{j-1} + \sum_{j>t} K_j d_j + \sum_{j>t} K_j^2\right)$$

for the direct stable endpoint implementation. The K_j^2 term can be reduced by stable positive reductions; we count the actual direct implementation in this experiment rather than transferring the old asymptotic MAC assertion. For m^{d_t} cells, multiply this preprocessing bound by m^{d_t}. A Cartesian lookup uses d_t binary searches in stored grid edges, followed by one label read and one comparison; its arithmetic is $O(d_t \log m)$. A scalar-regret lookup additionally reads the future caps and takes a finite maximum. Neither method evaluates the future activations of the input.

AN.2 Label stability, recall, and the exact optimal-depth qualifier

On an accepted cell, every future exit predicts k_A, and the observed prefix exit predicts k_A. For any unknown label y, all their binary task utilities are the same number $\mathbf{1}\{k_A = y\}$. Starting at the first available exit, no future best-so-far utility can increase, whereas cumulative transition cost cannot decrease. Hence immediate stopping maximizes the resource-correct task reward. It is the shallowest maximizer even if some transition costs are zero. Full with-recall inference can return any of its stored exits without changing the predicted label. Rejected or out-of-region inputs run that identical fallback, proving rowwise prediction preservation for the complete hybrid policy.

At a later stopping proposal, this argument gives $\mathfrak{S}_t^u = 0$ only. It does not erase a prefix regret, certify the unknown true class, or imply that the fallback is statistically optimal. In particular, a shallow unverified baseline may have better average test accuracy than the full network. The demonstrator reports exactly that outcome rather than changing its target.

Verification costs are additional consumed computation. Writing $\widetilde{R}_T = B_T - C_T - Z_T$ and using the same uncharged comparator $R^\star = \max_j (B_j - C_j)$ gives

$$R^\star - \widetilde{R}_T = \mathfrak{G}_T + Z_T.$$

This proves the overhead correction in the main text. If preprocessing costs P and is amortized over N deployments, P/N joins Z_T under the same unit conversion. For wall-clock comparisons, one must measure the complete implemented paths directly; no conversion of an exponential, memory read, Python dispatch, or synchronization into one MAC is presumed.

AN.3 Finite shared-proof graphs and an attained frontier

A registered node has a depth index, an enclosure, and sufficient cache state so that future edge costs depend only on that node. An edge (v, w) discharges a nonempty consecutive block of outstanding exit obligations using a stored sound vector of caps. Its end enclosure must contain all states permitted by the edge's start enclosure. The sum of incremental edge costs must equal the cost of that registered program: shared preparation is charged on the first edge that performs it, not again on later edges. Construction of the library, cap evaluation needed to register it, and planning are separate charged costs. An as-yet-unexecuted verifier's unknown answer cannot be used as a free edge label. The theorem concerns available registered summaries, not clairvoyant planning over unknown future verifier outcomes.

A path certifies the target precisely when every discharged obligation obeys its threshold. Removing other edges is therefore both necessary and sufficient within this ordered proof system. Reverse topological induction proves the Bellman recurrence: each complete path chooses a first edge and then a complete suffix path. A finite acyclic graph has finitely many complete paths; whenever any path is feasible, the minimizing path exists. Computing all node values takes linear graph work after testing all edge caps. To obtain the budget frontier, enumerate the finite set consisting of zero and all positive cost-adjusted edge caps. At a fixed budget, the smallest feasible candidate is attained. This is not a claim that arbitrary shared-memory verification has a small state graph; histories with different enclosures or cache contents must not be merged without a proof of equivalent continuation options.

For an interval pass, each layer's input enclosure is the previous layer's output, so a chain state carries that enclosure once. A precompiled cell's all-future result forms one compound edge. The executable

checks contrast these shared edges with separate depth reads and brute-force every complete path of an explicit finite example. Those small checks are regression tests; the neural interval computation and measured pipeline are separate evidence.

AN.4 Repair of the separable frontier

Every menu in the attained separable theorem is finite and nonempty, with finite nonnegative action costs; a zero-cost trivial cap is permitted. Each feasible coordinate has an actual minimum, and finitely many joint choices ensure attainment at every budget with a feasible choice. General noncompact menus retain only

$$\inf_{(r_j):\, u_j(r_j)-d_j\le\lambda} \sum_j k_j(r_j) = \sum_j \inf_{r_j:\, u_j(r_j)-d_j\le\lambda} k_j(r_j).$$

For $k(r_n) = 1 + 1/n$ with all caps feasible, this infimum is one but there is no action at budget one. An alternative set of sufficient assumptions is a compact menu, lower-semicontinuous costs, and closed cap sublevel sets, together with nonempty feasibility. We do not apply an attained budget inverse without these assumptions. The earlier proof's assertion to the contrary is corrected in the retained appendix as well, not merely qualified elsewhere.

AN.5 Explicit neural indistinguishability witnesses

The width-two family in the main text uses a shared head, fixed prefix, and nonnegative activations. Matrix multiplication gives $H_j = (1, \theta_j)$ regardless of previous choices. Its exact utility vector ranges independently over $\{1/2, 4/5\}^L$; allowing $\theta_j \in [0, 1]$ fills the product interval by continuity and monotonicity of the logistic function. With $B_0 = 1/2$, $c_j = .02$, $L = 3$, and $\lambda = .1$, the all-zero network has zero continuation regret. Changing the first, second, or third unread parameter to one yields regret .28, .26, or .24, respectively, while every other parameter answer and the observed prefix remain unchanged.

The access model reveals θ_j only on a charged parameter read; arbitrary algebra after the reads is free. Any deterministic sound procedure that *terminates and certifies every safe instance* must read every potentially violating cell. A procedure allowed to abstain forever has no such obligation. This is a neural realization of the classical weighted OR argument, not an assertion that the weights of a fully loaded ResNet remain information-theoretically unknown. The retained exact-coordinate theorem now states completeness explicitly. The compiled interval algorithm has a different, fully charged preprocessing/lookup interface.

AN.6 Numerical implementation contract

Training uses float32; deployment exports the stored weights to float64 and uses a declared NumPy affine implementation. For a dot product of length d, the interval routine inflates both endpoints using $2\gamma_{2d+4}$ times a nonnegative absolute-product bound, plus outward `nextafter`, where $\gamma_n = n\epsilon/(1 - n\epsilon)$. This is conditional on the standard floating-point dot-product error model, finite intermediates, and no destructive underflow; it is not a formally verified BLAS library. The executable implementation tests that the actual logged trajectories lie in the resulting intervals. The primary label gate uses strict interval comparisons and never uses the exponential-based confidence caps. The latter are numerical audit quantities with a small added tolerance, not a claim of correctly rounded `libm` certification on arbitrary hardware. Deployment grid membership uses stored edge arrays and rejects the rightmost/outside boundary rather than clipping an out-of-region input into a certified cell.

AO Exact-coordinate and separable baselines (corrected)

AO.1 Weighted support-query oracle

Fix t. The exact support vector $q^\star$ is computationally unresolved inside a declared product family $\mathcal{Q}_t = \prod_{j\in\mathcal{J}_t}[\ell_j, u_j]$. Querying coordinate j costs $\kappa_j \ge 0$ and returns $q^\star_j$ exactly. Queries may be adaptive. After queried set S and answers r_S, a zero-error certificate must upper-bound $U_t(q)$ for every $q \in \mathcal{Q}_t$ agreeing with the transcript.

Theorem 30 (Smallest transcript-valid certificate)**.** *After any adaptive transcript (S, r_S), the unique pointwise smallest zero-error certificate is*

$$\widehat{U}_t(S, r_S) = \left[\max\left\{\max_{j\in S}(r_j - d_j),\ \max_{j\notin S}(u_j - d_j)\right\}\right]_+,$$

with the maximum over an empty set interpreted as $-\infty$.

Soundness follows by coordinatewise upper bounds. Minimality is a genuine indistinguishability statement: for every unqueried coordinate, a completion at its upper endpoint has exactly the same observed query transcript.

Corollary 31 (Exact target-certification complexity)**.** *For tolerance $\lambda \geq 0$, let*

$$A_\lambda = \{j \in \mathcal{J}_t : u_j - d_j > \lambda\}.$$

Require a zero-error algorithm to terminate and certify every safe instance. On every instance satisfying $U_t(q^\star) \leq \lambda$, such an algorithm must query every $j \in A_\lambda$, and hence incurs cost at least $\sum_{j\in A_\lambda} \kappa_j$. Querying precisely A_λ, stopping early upon a violating answer, attains this lower bound on every safe instance.

This is an exact deterministic oracle lower bound, not a claim about arbitrary circuits or hardware. Appendix AP gives the proof and a non-product feasible-family criterion: a coordinate is necessary only when an unqueried transcript admits both a safe completion and an indistinguishable violating completion.

AO.2 Which support errors matter?

Let $\bar{q}_j = q_j^\star + \varepsilon_j$, with $\varepsilon_j \geq 0$, and define $a_j = q_j^\star - d_j$, $m = [\max_j a_j]_+$, and $\bar{m} = [\max_j (a_j + \varepsilon_j)]_+$.

Theorem 32 (Exact active-depth slack)**.**

$$\bar{m} - m = \max_{j\in\mathcal{J}_t} \big[\varepsilon_j - (m - a_j)\big]_+.$$

Consequently $\bar{m} - m \leq \max_j \varepsilon_j$, but a support error affects the certificate only after exceeding that depth's activity margin $m - a_j$.

The theorem replaces a global sup-norm statement by an exact localization: refining inactive depths is useless until their error crosses the gap to the cost-adjusted active support.

AO.3 Optimal separable refinement

For each depth j, let the finite nonempty menu $\mathcal{R}_j$ index sound caps $u_j(r)$ and finite nonnegative costs $k_j(r)$. Choices are independent and costs add. Define

$$K_t(\lambda) = \sum_{j\in\mathcal{J}_t} \min\{k_j(r) : u_j(r) - d_j \leq \lambda,\ r \in \mathcal{R}_j\},$$

where an empty minimum is $+\infty$.

Theorem 33 (Verifier cost–certificate frontier)**.** *$K_t(\lambda)$ is exactly the minimum verification cost needed to certify $U_t \leq \lambda$ in a separable refinement family. For a budget K admitting at least one feasible refinement vector, the smallest attainable certificate is*

$$\lambda_t^\star(K) = \min\{\lambda \geq 0 : K_t(\lambda) \leq K\}.$$

Shared computations are not forced into an additive fiction: they can be represented as compound queries/refinements. The separable theorem is exact when its declared cost model is exact and otherwise supplies a design baseline.

For general menus only the infimal-cost equality remains valid; an unattained infimum must not be used as an attainable budget boundary.

AP R28 COMPUTATIONAL MODEL AND PROOFS

This appendix separates three objects that were conflated in earlier revisions: the deterministic network trajectory, a sound semantic abstraction of its future, and a declared computational interface for refining that abstraction.

AP.1 MEASURABILITY AND SUPPORT SEMANTICS

Fix a prefix depth t, and let $\mathcal{F}_t$ contain the executed states, exits, costs, and all frozen deployment constants available by that depth. A verifier is a correspondence

$$\mathfrak{R}_t : \Xi_t \longmapsto 2^{[a,b]^{L-t}}$$

such that $\mathfrak{R}_t(\Xi_t)$ is nonempty and contains every continuation the verifier promises to cover. The main deterministic use is pathwise, so no probability measure is required for Theorem 23. When the certificate is used as an $\mathcal{F}_t$-measurable random variable, we additionally require

$$\Xi_t \longmapsto \sup_{v \in \mathfrak{R}_t(\Xi_t)} v_j$$

to be measurable for every future j. A standard sufficient condition is that $\mathfrak{R}_t$ is a measurable compact-valued correspondence and $v \mapsto v_j$ is continuous. Compactness is not required for the algebra: all statements use a supremum and remain valid when it is not attained.

For a known deterministic network and known input, unrestricted exact suffix execution identifies one trajectory. Hence semantic uncertainty is not intrinsic randomness. A useful verifier is a cheaper sound abstraction whose supports can be bounded without performing all skipped operations.

AP.2 PROOF OF THE PATHWISE AND JOINT-SET IDENTITIES

Proof of Lemma 22. For $j \geq t$,

$$R_j - R_t = \max\left\{0, \max_{t<k\leq j} \left(g_k(H_k) - B_t\right)\right\} - (C_j - C_t).$$

For each fixed k, nonnegative transition increments imply $C_j - C_t \geq C_k - C_t$ whenever $j \geq k$. Therefore the best time at which to exploit the newly observed utility $g_k(H_k)$ is $j = k$. Maximizing over $j \geq t$ gives

$$\max_{j\geq t}(R_j - R_t) = \max\left\{0, \max_{k>t}\left(g_k(H_k) - B_t - (C_k - C_t)\right)\right\},$$

which is the claim. Zero-cost transitions cause ties but do not invalidate the equality. □

Proof of Theorem 23. Let $f_j(v) = v_j - d_j$. By Lemma 22,

$$\sup_{v\in\mathfrak{R}_t} \mathfrak{S}_t(v) = \sup_{v\in\mathfrak{R}_t} \left[\max_{j\in\mathcal{J}_t} f_j(v)\right]_+.$$

The positive-part map is monotone and continuous, so it commutes with a supremum. A supremum commutes with a finite maximum:

$$\sup_{v\in\mathfrak{R}_t} \max_{j\in\mathcal{J}_t} f_j(v) = \max_{j\in\mathcal{J}_t} \sup_{v\in\mathfrak{R}_t} f_j(v).$$

The two inequalities follow respectively because each f_j is dominated by the finite maximum and because, for any $\epsilon > 0$, a point within ϵ of the support of a maximizing coordinate is available. Since $\sup_v f_j(v) = q_j^\star - d_j$, the result follows. No Cartesian closure or joint attainment is used. □

Scope of coupling invariance. The proof uses a scalar finite maximum and a worst-case supremum at a fixed prefix. It does not imply that marginal supports determine $\mathbb{E}[\max_j f_j]$, tail risk, vector-valued constraints, simultaneous confidence allocations, sequentially refined sets, or the cost of computing the supports. In all of those problems dependence can matter.

AP.3 THE WEIGHTED EXACT-SUPPORT QUERY MODEL

Let

$$\mathcal{Q}_t = \prod_{j \in \mathcal{J}_t} [\ell_j, u_j]$$

be a declared family of possible exact-support vectors. The unknown object is the deterministic vector $q^\star$; the product is computational uncertainty before queries, not a stochastic model of the network. Query j costs $\kappa_j \geq 0$ and reveals $q_j^\star$ exactly. An adaptive algorithm selects each next query as a function of the indices and answers already seen.

For a transcript $\tau = (S, r_S)$, define its completion class

$$\mathcal{Q}_t(\tau) = \{q \in \mathcal{Q}_t : q_j = r_j \text{ for all } j \in S\}.$$

Target-certification complexity additionally requires termination and certification of every safe instance; soundness alone allows abstention. A certificate $V(\tau)$ is zero-error when $V(\tau) \geq U_t(q)$ for every $q \in \mathcal{Q}_t(\tau)$.

Proof of Theorem 30. For every completion,

$$q_j - d_j = \begin{cases} r_j - d_j, & j \in S, \\ \leq u_j - d_j, & j \notin S. \end{cases}$$

Taking the finite maximum and positive part proves soundness of $\widehat{U}_t(S, r_S)$.

For minimality, let M denote the expression inside the positive part. If $M \leq 0$, every valid certificate is nonnegative, so zero is minimal. If $M > 0$ and it is achieved by a queried coordinate, then every completion has $U_t(q) \geq M$. If it is achieved by an unqueried coordinate $j^\star$, choose the completion $q_{j^\star} = u_{j^\star}$, keep every queried answer fixed, and choose arbitrary admissible values for the other unqueried coordinates. It has the identical transcript and $U_t(q) \geq M$. Therefore no transcript-valid certificate can be smaller than $[M]_+$. □

Proof of Corollary 31. Consider a safe instance $q^\star$ with $U_t(q^\star) \leq \lambda$. Suppose an algorithm terminates with a valid certificate at most λ while some $j \in A_\lambda$ remains unqueried. Replace only that coordinate by u_j. The resulting vector is in $\mathcal{Q}_t$, has the same transcript, and satisfies $U_t(q) > \lambda$, contradicting zero-error validity. Thus every $j \in A_\lambda$ must be queried and the cost lower bound follows.

The matching algorithm queries exactly the indices in A_λ, in any order. It reports failure immediately if an answer satisfies $r_j - d_j > \lambda$. If all answers are safe, then queried coordinates are at most λ, while every unqueried coordinate has $u_j - d_j \leq \lambda$ by definition. Theorem 30 then gives a certificate at most λ. On a safe instance the algorithm queries all and only A_λ, attaining the lower bound. □

Ordering unsafe-instance queries. The theorem fixes the exact safe-instance cost. On unsafe instances an algorithm may stop after discovering one violating coordinate. If a distribution over violations is additionally declared, query ordering becomes a separate expected-cost problem; no such distribution is needed for the zero-error statement.

AP.4 BEYOND A PRODUCT FAMILY

The product oracle is intentionally explicit. For an arbitrary feasible family $\Omega \subseteq \mathbb{R}^{|\mathcal{J}_t|}$, a query transcript induces $\Omega(\tau)$. The smallest certificate is always

$$V^\star(\tau) = \sup_{q \in \Omega(\tau)} U_t(q).$$

A query is necessary for a target-λ certificate at transcript τ whenever omitting it leaves both (i) a completion consistent with the safe instance and (ii) an indistinguishable completion with $U_t(q) > \lambda$. This witness criterion is the exact general indistinguishability principle. The closed form in Corollary 31 results from the product family's ability to raise any one unqueried coordinate independently to its upper endpoint. Applying the same lower bound to a structured neural verifier therefore requires either the product interface as the declared oracle or explicit feasible witnesses.

AP.5 EXACT ACTIVE-DEPTH SLACK

Proof of Theorem 32. Let $a_0 = 0$ and $\varepsilon_0 = 0$, so $m = \max_{j \in \{0\} \cup \mathcal{J}_t} a_j$. Since every $\varepsilon_j \geq 0$,

$$\bar{m} - m = \max_{j \in \{0\} \cup \mathcal{J}_t} (a_j + \varepsilon_j - m) = \max_{j \in \{0\} \cup \mathcal{J}_t} \big(\varepsilon_j - (m - a_j)\big).$$

The dummy term is $-m$, not zero unless $m = 0$. If $m > 0$, a future maximizing coordinate has $a_j = m$ and a nonnegative perturbed term. If $m = 0$, the dummy term is zero. Thus the maximum equals the positive part over future coordinates in both cases. The sup-norm corollary follows from $m - a_j \geq 0$. □

Active set interpretation. Let $\mathcal{A} = \{j : a_j = m\}$ when $m > 0$. Errors on $\mathcal{A}$ pass directly into the certificate, while an inactive coordinate j must first spend $m - a_j$ units of error before changing the result. When $m = 0$, the margin is $-a_j$, the distance from that depth to positive regret.

AP.6 OPTIMAL SEPARABLE VERIFIER FRONTIER

For the attained theorem assume each $\mathcal{R}_j$ is finite and nonempty, with finite nonnegative costs. Define

$$\mathcal{C}_j(\lambda) = \{r \in \mathcal{R}_j : u_j(r) - d_j \leq \lambda\}.$$

Proof of Theorem 33. A selected refinement vector $r = (r_j)_j$ certifies $U_t \leq \lambda$ if and only if every coordinate satisfies $u_j(r_j) - d_j \leq \lambda$. The "only if" direction follows because the certificate is their finite maximum; the "if" direction is immediate. Thus feasible choices factor as $\prod_j \mathcal{C}_j(\lambda)$. Under additive costs,

$$\inf_{r_j \in \mathcal{C}_j(\lambda)\ \forall j} \sum_j k_j(r_j) = \sum_j \inf_{r_j \in \mathcal{C}_j(\lambda)} k_j(r_j) = K_t(\lambda).$$

With finite menus every feasible infimum is a minimum, and the finite set of joint cap values has an attained smallest value at every feasible budget. For arbitrary menus only the displayed infimal-cost identity is justified; costs $1 + 1/n$ have infimum one but no action affordable at budget one. The attained budget inverse is not asserted in that setting. □

Compound refinements. A shared propagation pass that tightens several depths should be represented as one action with one cost and a vector of caps. The optimization then becomes a finite covering or general action-selection problem rather than the separable sum. R28 makes no claim that additive accounting is automatically valid for such implementations.

AP.7 CONCRETE SHARED-HEAD BOX COMPLEXITY

At depth j, after the scalar radius is updated, form K lower exponentials $a^-_{j,k}$, K upper exponentials $a^+_{j,k}$, and their single sum S^-_j. Then

$$q_{j,k} = \frac{a^+_{j,k}}{a^+_{j,k} + S^-_j - a^-_{j,k}}$$

costs constant scalar work per class. Computing the maximum over k and updating the running cost-adjusted certificate are also linear in K. Across $L - t$ future depths this proves Proposition 29. Standard log-sum-exp shifting can be used for numerical stability without changing the bound.

For the pairwise construction, each candidate class k uses $K - 1$ competitor margins and therefore costs $O(K^2)$ per depth in the direct implementation. Low-rank or shared-norm structure may reduce this, but no such structure is assumed in the theorem.

AP.8 COMPUTABILITY VERSUS SEMANTICS

The deterministic support theorem remains meaningful with exact real arithmetic as a mathematical statement. A deployed verifier additionally needs a finite representation, a numerical error contract, and a cost model. In R28:

1. the oracle theorem treats exact support answers and declared nonnegative query weights;
2. the cap theorem treats any sound finite-precision upper caps;
3. the CI regression checks arithmetic and source hashes but does not prove that a hardware implementation satisfies the declared weights or wall-clock latency.

This separation prevents semantic soundness, numerical reproducibility, and systems efficiency from being used as substitutes for one another.

AQ CONSOLIDATED STOPPING, VERIFICATION, AND STATISTICAL RESULTS

This appendix consolidates the valid technical results carried forward from the R20–R27 chain. The historical source files remain in the R28 directory for provenance, but the compiled R28 paper uses this corrected consolidation so superseded transcript language is not reintroduced.

AQ.1 POLICY-AGNOSTIC DECOMPOSITION

For any prefix-adapted stopping time T, recall that

$$\mathfrak{P}_T = \max_{j \le T} R_j - R_T, \qquad \mathfrak{S}_T = \left[\max_{j \ge T} R_j - R_T\right]_+, \qquad \mathfrak{G}_T = \max_j R_j - R_T.$$

The index set for the global maximum is the union of the prefix and suffix index sets, hence

$$\mathfrak{G}_T = \max\{\mathfrak{P}_T, \mathfrak{S}_T\}.$$

Consequently every suffix certificate $U_T \ge \mathfrak{S}_T$ gives

$$\mathfrak{G}_T \le \max\{\mathfrak{P}_T, U_T\}.$$

No optimality of the stopping rule is assumed.

AQ.2 PREFIX-ENVELOPE LIFTING

Suppose that, simultaneously for all $0 \le t < j \le L$, a prefix-measurable quantity $Q_{j|t}$ satisfies $g_j(H_j) \le Q_{j|t}$. Define

$$\widehat{B}^Q_{j|t} = \max\{B_t, Q_{t+1|t}, \dots, Q_{j|t}\},$$

and

$$\mathcal{U}_t(Q) = \min\left\{W, \max_{j>t}[\widehat{B}^Q_{j|t} - B_t - (C_j - C_t)]_+\right\}.$$

For each fixed t, replacing every unseen utility in the exact expression for future best-so-far utility by its simultaneous cap gives $\mathfrak{S}_t \le \mathcal{U}_t(Q)$. Because the inequalities hold for all t on the same trajectory, substituting an arbitrary prefix stopping time T requires neither a policy-specific union bound nor recalibration. Thus

$$\mathfrak{G}_T \le \max\{\mathfrak{P}_T, \mathcal{U}_T(Q)\}.$$

The construction remains valid when the cap is obtained from a heterogeneous verified set $\mathcal{H}_{j|t}$ in each depth's own representation space: one sets

$$Q_{j|t} = \sup_{h \in \mathcal{H}_{j|t}} g_j(h).$$

AQ.3 FIRST-VALLEY ACCOUNTING

Let

$$\Delta^B_j = B_{j+1} - B_j, \qquad m_j = \Delta^B_j - c_j.$$

The one-inspection first-valley rule continues while $\Delta^B_j > c_j$ and stops immediately after the first failed inspection. If the failure occurs on transition $k \to k+1$, the executed stop is $T = k+1$ and

$$\iota_T = [c_k - \Delta^B_k]_+.$$

All earlier inspected transitions increased reward strictly, while the final one changes reward by $\Delta_k^B - c_k \le 0$. Therefore the best prefix reward is attained immediately before or at that final inspection and

$$\mathfrak{P}_T = \iota_T \le c_k.$$

If no failure occurs before the terminal depth, $\mathfrak{P}_T = 0$.

For arbitrary suffix behavior, repeated use of

$$B_{j+1} - B_j = \Delta_j^B$$

and deletion of negative increments yields

$$\mathfrak{S}_t \le \mathcal{A}_t^{\mathrm{dir}} := \min\left\{ W, \sum_{j=t}^{L-1} [m_j]_+ \right\}.$$

This direct bound is pathwise but generally cannot be evaluated before the unseen increments occur; its role is analytical.

If a proposed first-valley stop is accepted only when a prefix certificate $\widehat{U}_T \le \lambda$, then on the certificate event

$$\mathfrak{G}_T \le \max\{\iota_T, \lambda\}.$$

If a fallback stop separately satisfies $\mathfrak{G}_{T_{\mathrm{fb}}} \le B_{\mathrm{fb}}$, the hybrid policy satisfies

$$\mathfrak{G}_{T_{\mathrm{hybrid}}} \le \max\{\iota_T, \lambda, B_{\mathrm{fb}}\}$$

on the intersection of the two guarantee events. Without a fallback guarantee, the certificate controls accepted overrides only.

AQ.4 Shared-head residual box

Assume

$$H_{j+1} = \mathrm{ReLU}(H_j + r_j(H_j)), \qquad \|r_j(h)\| \le \ell_j \|h\| + \rho_j.$$

Starting from observed H_t, define

$$s_t = \|H_t\|, \quad R_{t|t} = 0, \quad s_{j+1} = (1+\ell_j)s_j + \rho_j, \quad R_{j+1|t} = R_{j|t} + \ell_j s_j + \rho_j.$$

ReLU is nonexpansive. Induction gives $\|H_j\| \le s_j$ and $\|H_j - H_t\| \le R_{j|t}$.

For shared affine head $z(h) = Wh + b$, row $w_k^\top$, and temperature τ_j,

$$(z_j)_k \in [(z_t)_k - \|w_k\| R_{j|t}, (z_t)_k + \|w_k\| R_{j|t}].$$

Maximizing class k's softmax over this logit box places its own logit at the upper endpoint and every competing logit at its lower endpoint. The cap is

$$q_{j,k|t} = \frac{\exp(((z_t)_k + \|w_k\| R_{j|t})/\tau_j)}{\exp(((z_t)_k + \|w_k\| R_{j|t})/\tau_j) + \sum_{r\neq k} \exp(((z_t)_r - \|w_r\| R_{j|t})/\tau_j)}.$$

Thus $Q_{j|t} = \max_k q_{j,k|t}$ is a simultaneous prefix envelope, and the lifted certificate is pathwise valid.

A pairwise alternative bounds each competitor-target margin directly:

$$q_{j,k|t}^{\mathrm{pair}} = \left[1 + \sum_{r\neq k} \exp\left(\frac{(z_t)_r - (z_t)_k - \|w_r - w_k\| R_{j|t}}{\tau_j} \right) \right]^{-1}.$$

The minus-norm term is a lower bound on the competitor-minus-target margin, which yields an upper bound on target softmax. Since $\|w_r - w_k\| \le \|w_r\| + \|w_k\|$, this lower margin is no smaller than the separate-row lower margin; the pairwise denominator is therefore no smaller and the resulting cap no larger, so

$$\mathfrak{S}_t \le \mathcal{U}_t^{\mathrm{pair}} \le \mathcal{U}_t^{\mathrm{box}}.$$

The frozen audit found the numerical tightening immaterial at its replay tolerance; mathematical dominance does not imply empirical actionability.

AQ.5 One-sided split-conformal alternative

Before calibration, freeze a stopping rule T, prefix feature map F_T, predictor f, bounded target $V \in [0, V_{\max}]$, and level α. On n exchangeable calibration examples form residuals

$$E_i = V_i - f(F_{T_i}).$$

Let $r = \lceil (n+1)(1-\alpha) \rceil$. If $r \le n$, let $\widehat{q}$ be the r-th residual order statistic; if $r = n+1$, return the deterministic ceiling. Then

$$\mathcal{U}_T^{\mathrm{conf}} = \min\{V_{\max}, [f(F_T) + \widehat{q}]_+\}$$

satisfies

$$\mathbb{P}\{V \le \mathcal{U}_T^{\mathrm{conf}}\} \ge 1 - \alpha.$$

This is the usual exchangeable-rank argument. It is marginal for the frozen tuple and is not policy-uniform after adaptive retuning.

Taking $V = \mathfrak{S}_T$ and using the policy decomposition gives

$$\mathbb{P}\{\mathfrak{G}_T \le \max(\mathfrak{P}_T, \mathcal{U}_T^{\mathrm{conf}})\} \ge 1 - \alpha.$$

Taking $V = \mathfrak{G}_T^u$, the bounded global task regret, gives a direct architecture-general task certificate. Labels and full trajectories may be used to construct fit/calibration targets but are not deployment features.

If U_T^{det} is a pathwise suffix certificate and U_T^{conf} has coverage $1 - \alpha$, then

$$\mathbb{P}\{\mathfrak{S}_T \le \min(U_T^{\mathrm{det}}, U_T^{\mathrm{conf}})\} \ge 1 - \alpha,$$

because the deterministic event always holds and the conformal event has the declared probability.

AQ.6 Proxy-to-task transfer

Let u_j be task utility and g_j declared proxy utility. Define their best-so-far processes B_t^u, B_t^g, and let S_T be the proxy-selected return exit. With

$$D_T = \max_t |B_t^u - B_t^g| + |u_{S_T} - g_{S_T}|,$$

the reward difference between the task and proxy systems is bounded at both the hindsight optimum and the chosen return. The triangle inequality gives

$$\mathfrak{G}_T^u \le \mathfrak{G}_T^g + D_T.$$

This separates stopping certification for the declared proxy from empirical alignment between proxy and task.

For classification with deterministic smallest-index tie-breaking, let $\widehat{y}_j$ be the predicted label and $A_t = \{\widehat{y}_0, \dots, \widehat{y}_t\}$. If a verified future set proves $\widehat{y}_j \in A_t$ for every $j > t$, then for every unknown true label: if the label is in A_t, best prefix 0–1 utility is already one; if it is outside A_t, no verified future prediction is correct. Hence $\mathfrak{S}_t^u = 0$. This is a sufficient pathwise statement, not a claim about its frequency in the frozen data.

AQ.7 Statistical and systems interpretation

Deterministic envelope soundness, split-conformal coverage, proxy alignment, and policy quality are separate claims. A valid deterministic certificate can be vacuous; a conformal certificate can fail under transport; a proxy-safe stop can reduce task accuracy; and a compute-saving depth choice can lose after verifier/controller overhead. The R28 theory and audit keep these layers separate throughout.

AR R28 audit contract, matched-compute protocol, and limitations

AR.1 Immutable populations and permitted comparisons

R28 preserves three different populations and does not combine their numbers:

1. the once-opened official CIFAR-100 test used for the R22 table;
2. the predeclared 4,000-example unused CIFAR-100 training split first opened in R23 after the budget family was frozen;
3. the official-test and later unused-train SVHN populations used for the R24/R25 transport diagnostics.

The R22 CCI/per-price/line-DP comparisons and the R23 q25 comparison therefore answer different questions. A mean depth or accuracy from one population is not used to manufacture a matched comparator on another.

AR.2 Exact R23 q25 arithmetic

For each price, the q25 hybrid is compared with the $c_{\max}$ hybrid in the same immutable R23 JSON. The latter accepts zero valley overrides and is the frozen fallback on every row. Let $M_{\text{terminal}} = 246.765568$ million MACs be the audited terminal-path convention. The immutable analysis artifact stores both the same-population $c_{\max}$-to-q25 MAC difference and its fraction of this terminal path. The maximum uncounted verifier-plus-controller overhead compatible with a positive net MAC saving is exactly that stored difference. The executable R28 check gives

p	.01	.02	.04	.08	.16
$s_p(\%)$	5.258347	5.024776	4.550180	4.202824	2.620374
$M_{\text{break}}(\text{M})$	12.970103	12.388929	11.217131	10.366632	6.459064.

These are overhead ceilings, not measured runtimes. A deployed implementation must count feature extraction not already shared with the network, cap construction, policy evaluation, memory traffic, and synchronization against the same cost convention. A claim of wall-clock benefit additionally requires hardware timing.

The frozen q25 task-accuracy changes relative to the same-row $c_{\max}$ fallback are $-2.03, -2.28, -2.70, -2.85, -2.18$ percentage points. Proxy regret improves only at prices .08 and .16. Because the residual-box and trivial headroom decisions are identical row by row, none of these actions can be attributed to geometric verifier sharpness.

AR.3 Compute-matched baseline protocol

A valid matched-compute comparison on a future untouched row ledger must be fixed before outcomes are opened.

Comparator family. Include at minimum: (i) every fixed compute depth; (ii) a randomized mixture of adjacent fixed depths so any mean MAC target can be matched exactly in expectation; (iii) a confidence-only rule with thresholds fit on the policy-fit/tune populations; and (iv) the frozen policy fallback. All methods must use the same return-with-recall convention and the same controller/verifier overhead accounting.

Matching rule. For each price and candidate method, choose comparator parameters using only the fit/tune ledger to minimize absolute mean-MAC discrepancy, with predeclared tie-breaking toward the shallower comparator. On the untouched evaluation rows, report both the achieved MAC discrepancy and outcomes. If exact matching is impossible because costs are discrete, randomize only between the two adjacent declared operating points using a seed committed before evaluation.

Uncertainty. Report paired rowwise differences in proxy Bellman regret, task regret, task accuracy, and total measured cost. Use a predeclared paired bootstrap or a finite-sample concentration interval; for the randomized comparator, include the randomization seed and either condition on it or repeat over a predeclared seed set. Multiple prices require an explicit pointwise-versus-familywise statement.

Required row ledger. At minimum each row must contain example identifier/hash, price, method, executed-depth histogram contribution, returned exit, proxy payoff/regret, task outcome/regret, backbone MACs, verifier MACs, controller MACs, and measured latency when a latency claim is made. The existing R23 aggregate artifact does not contain enough per-example comparator outcomes

to reconstruct this analysis after the fact. R28 therefore does not claim that the q25 gate beats a compute-matched fixed-depth or confidence baseline.

AR.4 What the negative fresh result establishes

The rowwise identity

$$\mathcal{U}_t^{\mathrm{box}} = \mathcal{U}_t^{\mathrm{trivial}}$$

on every gate-relevant R23 row is stronger than equality of rounded averages. It establishes that this concrete shared-head scalar-radius/softmax-box verifier contributes no decision-relevant tightening on that population. It does not imply that:

- the prefix state contains no useful information;
- every neural verifier must be trivial;
- pairwise, polyhedral, branch-and-bound, or learned sound refinements cannot lower a cost-relevant support; or
- a better policy cannot improve task utility at matched cost.

The active-depth identity in Theorem 32 specifies a necessary quantitative target for refinement: a new method must reduce at least one support error beyond its activity margin, and its cost must remain below the saved continuation budget.

AR.5 Historical strictness at the available resolution

The historical digits checkpoint preserves the exact mean trivial headroom, the rounded mean box certificate, sample size 297, width bound 0.9, and a statement that the box is rowwise no larger. These aggregates, after allowing the documented rounding interval, force at least five rows to have strict headroom-minus-box gain greater than 10^{-6}. They do not identify which rows and do not reconstruct an empirical quantile. R28 continues to use the exact phrase "at least $5/297$ rows" and does not upgrade it to rowwise replay evidence.

AR.6 Label-stability condition

The label-free condition in the main paper is deterministic and sufficient: under fixed tie-breaking, every verified future argmax must lie in the set of labels already observed by the prefix. It then follows for every unknown true label that no future exit can improve the best-so-far 0–1 utility. The current frozen artifacts do not record how often this condition holds, so R28 makes no frequency or actionability claim. A future audit must store the certified future-label set and the observed prefix-label set per row before reporting coverage.

AR.7 SVHN transport and conformal scope

The predeclared official-test SVHN audit gives 7,492 covered rows out of 8,000, or 93.65%, at nominal 95%. Under the continuous-residual iid split-conformal reference calculation, the committed lower-tail probability is 6.09×10^{-4}. The later same-source unused-train diagnostic gives 7,587/8,000, or 94.84%, with lower-tail probability 0.338. Its rows were outcome-fresh, but its diagnostic design followed exposure to the official result; it is therefore supportive sensitivity evidence, not a replacement confirmation.

The feature-free conformal target quantile is 1.05333. Prefix features reduce the same-source mean upper bound to 1.01757 (official test: 1.01553), while 40.84% of same-source bounds still hit the deterministic ceiling 1.08. These numbers support the narrow conclusion that prefix features improve average tightness but do not make the current certificate useful for small regret budgets.

AR.8 What CI can and cannot certify

The R28 workflow can check:

- source/artifact presence and immutable historical files;

- the exact support-query and active-slack identities on exhaustive finite grids;
- the separable frontier against brute-force enumeration;
- R23-derived acceptance, saving, quality, and break-even arithmetic;
- LaTeX compilation, unresolved references, and the main-text page guard.

It cannot certify:

- that a declared verifier set contains the true future under untested model assumptions;
- exchangeability or distributional transport;
- that scalar operation counts predict device latency;
- a compute-matched quality comparison absent the required row ledger.

These limits are part of the scientific claim rather than exceptions hidden in repository notes.

AS EXECUTED NEURAL EVIDENCE: FULL LEDGER AND REPRODUCIBILITY

The committed script fixes seed 290925, 600 full-batch Adam steps at learning rate .006, equal cross-entropy weight for four heads, and a 256-by-256 grid. Training uses float32; frozen deployment uses exported float64 weights. The train/validation/evaluation row counts are 1077/270/450. Split hashes, model bytes, grid edges, primary label criterion and complete row IDs are stored. The root box is the training/validation prefix range expanded by five percent of coordinate range plus one. No test outcome chooses a gate threshold. A smaller grid prototype was evaluated during development, so the final results are exposed development evidence, not a fresh confirmation.

Matched comparisons. The confidence threshold is fixed from validation at the validation label-gate acceptance fraction. Exact mean network-MAC matching uses adjacent fixed depths whose cumulative MACs bracket the hybrid mean; a second mixture uses only the first and final exit. Mixture correctness is integrated analytically over an input-independent routing coin. Therefore these rows report expected compute and accuracy, not observed values of a nonexistent randomized run. The bootstrap resamples the paired evaluation rows 2,000 times with seed 991; it is a descriptive interval on this exposed split, without selection-adjusted confirmatory inference. All methods return the maximum-confidence observed exit, with shallow tie-breaking. The primary label gate itself needs no softmax on accepted rows.

Method	Mean depth	MACs	Acc. (%)	Paired Δ 95% (pp)
Fixed 0	0.000	2132.0	92.44	–
Fixed 1	1.000	2900.0	91.33	–
Fixed 2	2.000	5268.0	90.89	–
Fixed 3 / full recall	3.000	5940.0	90.22	–
Compiled label	0.967	3359.0	90.22	[+0.00, +0.00]
Confidence (validation frozen)	1.047	3460.6	90.44	[+0.00, +0.67]
Matched adjacent expectation	1.194	3359.0	91.25	[-0.17, +2.32]
Matched prefix/full expectation	0.967	3359.0	91.73	[+0.30, +2.86]

Table 48: Complete same-row ledger. Accuracy differences are comparator minus compiled gate. Fixed-depth point estimates have no reported paired interval.

Observable support slack, not unknown nonlinear robust supports. For the frozen deterministic network, full-trajectory audit computes the exact singleton-future utility $q_j^{\mathrm{real}} = g_j(H_j)$. We apply the active-slack identity with these utilities and the computed box caps, not with unknown supports of a larger nonlinear reachability set. This is a justified observable relaxation diagnostic; the full trajectories are audit-only. The zero coordinate is ordered first, so it wins ties at zero regret.

At price 0.01, uncertified first-valley ASTI executes mean depth 1.653333 at 4337.796 mean network MACs, and has 91.56% accuracy. At price 0.02, uncertified first-valley ASTI executes mean depth 1.568889 at 4145.369 mean network MACs, and has 91.56% accuracy. At price 0.04, uncertified

Price	Active counts $(0, 1, 2, 3)$	Strict gain	Box gate	Trivial gate	ASTI acc.
0.01	(175,233,22,20)	149	208	208	91.56%
0.02	(213,201,22,14)	137	233	233	91.56%
0.04	(238,186,13,13)	131	260	260	91.56%

Table 49: All 450 rows. Strict gain means trivial-minus-box greater than 10^{-6}; proxy gates use $\lambda = c$. Label stability is a separate gate.

first-valley ASTI executes mean depth 1.486667 at 3992.124 mean network MACs, and has 91.56% accuracy.

No learned line-DP comparator is trained for this new demonstrator; the old line-DP results retain their original population. The R23 ResNet aggregate cannot supply an honest same-row reconstruction of these comparisons; its matched-compute request remains open rather than being replaced by digits.

Measured cost and storage. The network-MAC counts are prefix 2,132 and suffix increments 768, 2,368, and 672, including each executed affine head. They exclude scalar softmax, memory and dispatch. Complete wall-clock measurements include the actual NumPy implementation of those operations. Three warmups precede each measurement; batch one uses nine repetitions and the others 25. Full and compiled workloads are timed separately, not interleaved, on a shared CPU; measurements are descriptive and can vary with load. Every batch size tested is reported, including slowdowns. No GPU or old-ResNet timing is inferred.

Batch	Full p10/p50/p90 (ms)	Hybrid p10/p50/p90 (ms)	Ratio	Preprocess break-even
1	20.569/20.606/20.703	23.305/23.393/23.457	0.88	none
32	1.450/1.466/1.541	1.629/1.647/1.699	0.89	none
450	0.724/0.730/0.739	0.405/0.420/0.444	1.74	3.376M

Table 50: Times cover 450 examples. Break-even includes all measured verification preprocessing and proof serialization, but excludes training the common model.

The 65,536-cell table contains 42,742 all-future certified-label cells. Online label/edge storage is 135,184 bytes; the full uncompressed proof arrays occupy 33,165,328 bytes. Model training costs 1.865 seconds and is common to both inference paths. Verification preprocessing costs 2.329 seconds, including 1.136 seconds of interval compilation. The deployment environment is AMD EPYC 9V74 80-Core Processor, one thread, Python 3.13.15, NumPy 2.3.5, and PyTorch 2.10.0+cpu.

What checks establish. The independent checker recomputes the frozen network's predictions on every logged row, reconstructs intervals for occupied cells, rechecks label separation, verifies matched mean MACs, and enumerates explicit neural endpoint witnesses and finite shared-proof paths. These checks test a real network as well as small theorem examples. They do not prove the numerical library, imply coverage under distribution shift, or establish fresh-population statistical generalization. The standard floating-point contract is stated in Appendix AN; all empirical interval containment checks passed.