Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Fail-Closed Certification of Multi-Stage AI Policies

Norkobil Saydirasulov
Gachon University
Young-Im Cho
yicho@gachon.ac.kr

Gachon University
D. A. Davronbekov
Tashkent University of Information Technologies
B. A. Nazarov
Samarkand State University named after Sharof Rashidov
Abdinabi Mukhamadiyev
Samarkand State University named after Sharof Rashidov

Article

Keywords: multi-stage AI, certification, fail-closed assurance, distribution-free risk control, multiple testing, partial order

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11317920/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Fail-Closed Certification of Multi-Stage AI Policies

Norkobil Saydirasulov[1,2], Young-Im Cho[1*], D. A. Davronbekov[3], B. A. Nazarov[4], Abdinabi Mukhamadiyev[5]

[1*]Department of Computer Engineering, Gachon University, Seongnam-si, Republic of Korea.
[2]Department of Engineering and Communications, ZARMED University, Samarkand Campus, Samarkand, Uzbekistan.
[3]Faculty of Radio and Mobile Communication, Tashkent University of Information Technologies, Tashkent, Uzbekistan.
[4]Department of Computing Systems Engineering, Faculty of Artificial Intelligence and Digital Technologies, Samarkand State University named after Sharof Rashidov, Samarkand, Uzbekistan.
[5]Department of Artificial Intelligence and Information Systems, Samarkand State University named after Sharof Rashidov, Samarkand, Uzbekistan.

*Corresponding author(s). E-mail(s): yicho@gachon.ac.kr;

Abstract

Multi-stage AI systems are deployed as policies, not as isolated predictors. Distribution-free risk-control methods can certify a fixed candidate family, but they leave open a practical question: when may several policies share an ordered testing path, and what must be checked before that order is used? We present Admissible Cascade Envelope via Learn-then-Test (ACE-LTT), a fail-closed framework that derives testing structure from policy semantics. Each constrained loss declares the decision set it concerns, comparability is fixed before test outcomes are observed, and only relations supported by the loss and inference contracts enter a testing chain. If a required premise cannot be verified, the procedure returns a typed refusal rather than a weaker certificate. Under the stated sampling assumptions, ACE-LTT controls the probability of any invalid returned certificate at level $\boldsymbol{\delta}$ and permits lossless fixed-sequence stopping within admissible chains. With the uniform allocation used here, fewer admissible chains cannot shrink the certified set in a successful execution. In a preregistered Monte Carlo study with known truth, 2.3 million admissible-cover comparisons showed no violation of the predicted set inclusion, and none of 90 validity cells met the

failure rule. An edge–cloud wildfire case separates certification validity, family informativeness, and deployment preference.

Keywords: multi-stage AI, certification, fail-closed assurance, distribution-free risk control, multiple testing, partial order

1 Introduction

Multi-stage AI systems are deployed as policies rather than as single predictors. A deployed configuration may decide what information is acquired or transmitted, when additional computation is invoked, which actions are allowed, and when responsibility is handed to an operator. These choices are coupled. Moving one threshold can change predictive performance, escalation frequency, resource use, and the set of actions available to the system. A guarantee for one predictor therefore does not, by itself, describe the behaviour of the deployed policy.

Distribution-free risk-control methods provide a useful basis for certifying candidate predictors or decision rules without a parametric model for the data-generating distribution [1–3]. Once the candidate family and valid tests are fixed, these methods can control the probability of returning candidates whose risks exceed prescribed limits. Multi-stage policies introduce an earlier design problem. Their coordinates may govern acquisition, escalation, and response authority at the same time, and a computationally convenient order need not match the order of the constrained risks. Before testing begins, one must therefore decide which policies can share an ordered path and what assumptions make that sharing legitimate.

ACE-LTT (Admissible Cascade Envelope via Learn-then-Test) treats the deployable policy as the unit of certification. The representation separates autonomous alarm authority from broader response authority, because those two choices define different operational sets. A case sent to an operator, for instance, may count as a successful response without counting as an autonomous alarm. Each constrained loss therefore states which decision set and aggregation its certificate refers to. Comparability is then derived from the policy semantics before test outcomes are opened, and a relation is used for shared testing only when all declared loss and inference requirements support it.

The ACE-LTT order is not needed for family-wise error control. A prespecified fixed-sequence procedure can remain valid under an arbitrary fixed order. The order is used here for a different reason: if the loss and inference contracts are compatible with it, then candidates can be arranged in chains for which stopping after the first failure is lossless. The same admissible relation determines the chain cover used to allocate the error budget. With the uniform allocation studied here, and conditional on successful execution, a cover with fewer admissible chains cannot return a smaller certified set. The result concerns the informativeness of the testing structure, not the validity of the underlying multiple-testing procedure.

The procedure also checks the premises on which a certificate depends. Audit-population membership, score provenance, deterministic resource bounds, estimand-

to-p-value compatibility, and the registered policy semantics are all represented explicitly. If one of these premises cannot be verified, execution ends with a typed refusal. Realised audit metadata may confirm or terminate a procedure that was fixed in advance, but it may not be used to redesign the family, chain cover, order, or error allocation. This matters in practice because a statistically valid test can still be applied to an uninformative family, and the policy selected for deployment need not coincide with the full set of policies that passed certification. We keep certification validity, family informativeness, and deployment preference separate in the analysis.

The statistical tools used after the testing structure is fixed are established ones. ACE-LTT does not propose a new finite-sample risk bound, p-value construction, or general multiple-testing principle. Its contribution is to connect the operational semantics of a multi-stage policy to a testable certification structure and to make the premises of that connection auditable. We study the construction in three ways. A dataset-free synthetic example isolates the effect of the policy order. A preregistered Monte Carlo study compares the frozen implementation with known truth in one-, two-, and three-coordinate families. Finally, an edge–cloud wildfire-detection case provides exploratory runs, a recorded refusal, and a prospective registered execution under realistic policy semantics. These experiments examine implementation and operating behaviour; the certification guarantee comes from the assumptions and testing construction stated below.

2 Related work

Risk-controlling prediction sets [2], Learn-then-Test [1], and conformal risk control [3] provide finite-sample guarantees for fixed collections of candidate procedures. ACE-LTT uses these statistical tools rather than replacing them. The issue addressed here comes one step earlier: how to organise a family of multi-stage policies, and what must be true before several policies are allowed to share a sequential testing path.

The multiplicity arguments themselves come from structured multiple testing. Closed testing [4], fixed-sequence testing [5], testing in a prespecified order [6], gatekeeping [7], and graphical procedures [8] control family-wise error for hypotheses with specified structure, while the sequential rejection principle gives a general formulation that covers several of these procedures [9]. Theorem 1(i) is an instance of that established principle. Work on partially ordered or directed-acyclic-graph (DAG)-structured hypotheses [10–12] and model selection over posets [13] likewise begins with an order or graph supplied by the problem. In ACE-LTT, the initial relation is derived pointwise from policy semantics, and an edge enters the statistical graph only if the associated losses and inference rules also certify that comparison. The cover and its error allocation are then frozen before certify-fold outcomes are read. We use a simple Bonferroni allocation across chains and make no claim that it is optimal for the resulting order.

Pareto Testing [14] and reliability-graph Pareto testing [15] are closely related structure-aware variants of Learn-then-Test. Pareto Testing uses an optimisation split to build a promising frontier and testing order. Reliability-graph testing constructs a directed acyclic graph from prior information and held-out data, then performs

graph-structured testing under false-discovery-rate (FDR) control. Those structures are estimated or supplied to improve exploration of the candidate space. ACE-LTT places a stricter requirement on the relation used for shared fixed-sequence stopping: it must follow from the declared policy semantics and remain compatible with every constrained loss and inference rule. Family-wise validity does not require this restriction. It is used to establish the properties needed here: lossless stopping within an admissible chain and, for successful executions under the stated uniform allocation, the cover-size inclusion result. Adaptive Learn-then-Test [16] changes another component of the problem by replacing conventional p-value testing with sequential, data-dependent tests based on e-processes.

Risk control has also been studied directly for cascades, routing, and adaptive computation. CALM [17] and early-exit risk control [18] calibrate exit decisions within a network. Hou et al. [19] use parallel fixed-sequence testing in an edge–cloud–expert cascade with a monotone confidence threshold. Calibrate-Then-Delegate [20] calibrates delegation thresholds with held-out data, Pareto filtering, and Learn-then-Test, obtaining finite-sample control of the constrained objective while optimising a separate operating criterion. Conformal alignment [21] treats edge–cloud escalation as a multiple-testing problem with statistical control over unreliable edge decisions. CR^2 [22] combines cost-aware device–edge routing with conformal risk control.

Multi-dimensional threshold families are not new. CARE [23], for example, jointly calibrates a two-dimensional threshold pair along a fixed, data-independent path through the grid and uses Learn-then-Test fixed-sequence testing. LEC [24] provides finite-sample selection-conditioned risk control for threshold-based selective prediction and extends the construction to two-model routing. ACE-LTT's contribution is not the use of several thresholds or monotonic threshold ordering by itself. The setting considered here has heterogeneous operational coordinates that induce a partial order, and candidates share statistical testing only when the corresponding relation is supported by the declared loss and inference contracts.

A second feature is the explicit separation of alarm authority from broader response authority. Every constrained loss identifies the operational decision set to which its certificate applies. Split discipline, audit membership, estimand-to-p-value compatibility, and validity of a proposed chain relation are represented as executable checks rather than left as analyst-side assumptions. Failure of one of these checks leads to refusal, not to a relaxed interpretation of the certificate.

That fail-closed behaviour has clear precedents. Seldonian and high-confidence policy-improvement methods return a policy only when a high-confidence safety condition can be established; otherwise they report that no acceptable solution was found [25, 26]. A recent fail-closed certification protocol similarly uses explicit gates to block unsupported deployment claims [27], and preregistration has been proposed for machine-learning research [28]. ACE-LTT does not claim novelty for these ingredients in isolation. The combination used here binds the policy family, testing structure, inference declarations, and provenance before execution, and it keeps failures of those declarations in the audit record.

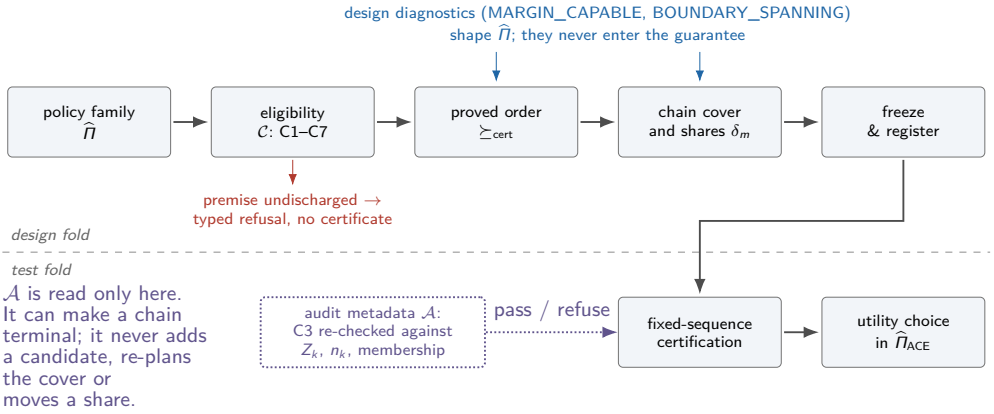

Fig. 1 The ACE-LTT design–registration–test workflow. Audit metadata $\mathcal{A}$ enters only as a verification that can refuse: it never reshapes the family, the cover, the order or the shares.

3 Methods

Figure 1 summarises the ACE-LTT design–registration–test workflow.

3.1 The policy object and its two decision sets

A **policy** is

$$\pi = (a,\ t_{\text{low}},\ t_{\text{high}},\ b,\ A,\ v,\ B_0), \qquad 0 \le t_{\text{low}} \le t_{\text{high}}, \qquad b \le A,$$

where a is the **acquisition mode**, $(t_{\text{low}}, t_{\text{high}})$ the **escalation band** on the first-stage score s, A the **alarm authority** at which the second stage may alarm autonomously, b the **response authority** below which an escalated input receives no response, $v : \mathcal{S} \to [0,1]$ a **veto authority** piecewise constant on a declared stratification, and B_0 a **payload budget level**. Writing $g_v = \min\{1,\ g + v(\sigma(x))\}$ for the verifier score after the declared veto in stratum $\sigma(x)$,

$$\text{alarm}(\pi) = \{s \ge t_{\text{high}}\} \cup \{\text{escalated} \wedge g_v \ge A\},$$

$$\text{response}(\pi) = \text{alarm}(\pi) \cup \{\text{escalated} \wedge b \le g_v < A\}.$$

A governs the alarm set, whereas b governs the broader response set. A case routed to a human operator is not an autonomous alarm, but it may still count as a successful response. Each constrained loss must therefore identify its catch set explicitly. The chain argument below is proved for both inclusions, so the two decision sets are not interchangeable.

The formulation uses two different budget quantities. B_0 is a policy coordinate in the **statistical** constraint

$$R_B(\pi) = \Pr(C_U > B_0) \le \beta,$$

where C_U is the declared per-unit payload normalisation. The transport cap $B_{\max}$, by contrast, is a **deterministic** property of the acquisition contract,

$$|r_a(x)| \le B_{\max} \qquad \forall x,$$

and is discharged by construction rather than by an observed sample maximum.

Let U denote the **units**, the grouping never split across folds. For each constrained risk k, a unit carries events $E_{k,u}$ that must be caught at least once, and the declared per-unit loss satisfies $\ell_k(\pi, u) \in [0, 1]$. Each loss also declares a **policy-independent audit indicator** $Z_k(u) \in \{0, 1\}$, determined by its empty-denominator rule: under `exclude_unit`, $Z_k(u) = 1$ iff u contains an event of type k; under `loss_zero`, $Z_k \equiv 1$. The certified estimand is the risk **on that population**,

$$R_k(\pi) = \mathbb{E}\big[\ell_k(\pi, U) \mid Z_k(U) = 1\big], \qquad n_k = \big|\{u : Z_k(u) = 1\}\big|,$$

so the empty-denominator rule does more than resolve a denominator: it *selects the estimand*. Under `exclude_unit`, the guarantee is about units carrying an event of type k, and every certification statement below is interpreted on that population.

3.2 Losses and the inference contract

A constrained loss specifies (α_k, kind, catch set, aggregation, weighting, empty-denominator rule, lemma id). Three fields require particular care. The **aggregation** must have a written monotonicity lemma, because frame-level catch-set inclusion does not by itself imply monotonicity of a unit-level loss. The **empty-denominator rule** makes the loss total before labels are inspected, preventing units from entering or leaving the audit population post hoc. The **weighting** must be policy-independent; this is necessary for the monotonicity argument, although not by itself sufficient for valid inference.

Inference contract $\mathcal{I}$. Every constrained loss declares a pair $(\widehat{\theta}_k,\ p_k)$, where $\widehat{\theta}_k$ is a statistic computed on the test fold, possibly vector-valued, and p_k is a function only of $\widehat{\theta}_k$ and the audit metadata $(n_k, \alpha_k, \dots)$. The declaration certifies:

(V) validity: p_k is valid for $H_{0,k} \colon R_k > \alpha_k$ under the sampling model of A2; and

(O) chain compatibility:

$$\pi \succeq_{\text{cert}} \pi' \quad \Longrightarrow \quad p_k(\pi) \le p_k(\pi')$$

at fixed audit metadata. The built-in pair, the only one the engine supplies, is the unweighted unit mean with the Hoeffding–Bentkus (HB) p-value

$$\widehat{\theta}_k = n_k^{-1} \sum_{u \in U_k} \ell_k(\pi, u), \qquad p_k = p_{\text{HB}}(\widehat{\theta}_k; n_k, \alpha_k),$$

using the Hoeffding–Bentkus construction [29, 30].

Property (O) is stated on $\succeq_{\text{cert}}$, rather than directly on the statistic, because a custom statistic need not inherit the per-unit loss order. **(V) alone gives family-wise validity; (O) is what the lossless stop requires.** For the built-in construction, certification therefore uses uniform unit weights, $w_{k,u} = 1/n_k$. A weighted mean represents a different estimand, so the HB p-value cannot simply be reused. Such a loss is accepted only when it names a separate statistic–p-value construction; otherwise declaration fails. The restriction applies to weights *across* units, which enter the inference procedure. Weights used internally within a unit remain part of the aggregation.

The inference contract is executable and frozen. Each loss *names* an implemented construction in the engine registry, which records the callable, the statistic it consumes, the weightings for which it is proved, and its (O) status. An unresolved name is refused when the loss is declared, rather than silently mapped to the built-in test for a different estimand. The source of each callable is digested and bound into registration, so the chain

$$\text{declared} \rightarrow \text{implemented} \rightarrow \text{registered} \rightarrow \text{unchanged at execution}$$

is checked explicitly, with failures reported as `I_INFERENCE_REGISTRATION`.

If a declared construction establishes (V) but not (O), **no** pair in the family receives a $\succeq_{\text{cert}}$ chain certificate, and the family is covered by singletons. A miss loss declared without a monotonicity lemma is handled in the same way, while a loss that claims antitonicity without naming its lemma is refused at declaration. Both cases reflect the same requirement: fixed-sequence stopping needs p_{joint} to be monotone along the chain, and the policy order alone does not establish that property.

Which losses can share an antitone chain is determined separately. Miss-type losses are antitone under catch-set inclusion. Content-of-response losses, such as false alarm, wrong-class response, precision, or operator load, need not be, because enlarging the response set can make them worse. This does not make such losses uncertifiable; it means that, without a separate proved order, a family that constrains such a loss is covered by singletons, and the multiplicity price is paid at cover-construction time. Payload loss is a third case: it is *equal* along a chain rather than merely antitone, because a, the escalation band and B_0 are fixed within that chain.

3.3 The eligibility contract

Before any p-value is computed, each candidate must satisfy the executable predicate $\mathcal{C}$. A candidate that satisfies it is *eligible*; *admissible* is reserved for relations, edges, chains and covers that satisfy the loss and inference requirements (§3.4).

Table 1: The eligibility contract $\mathcal{C}$. Each clause, what it requires, and the typed refusal code returned when it fails.

clause	requires	code
C1	scores and transmitted bytes from one and the same representation	`C1_MEASUREMENT_COHERENCE`
C2	shared detector-output digest, score floor, model identity, no dry-run scores	`C2_SCORE_CONTRACT`
C3	$n_k \geq n_{\min}(\alpha_k, \delta_m)$ **and** a digest of *which* units enter risk k	`C3_CERTIFIABILITY_AUDIT`
C4	$\lvert r_a(x)\rvert \leq B_{\max}$ for **every** input, by a static construction proof	`C4_PAYLOAD_CAP`

clause	requires	code
C5	family, order and shares depend only on the design fold and declared constants; audit metadata may refuse, never reshape	`C5_SPLIT_INDEPENDENCE`
C6	the claimed ordering is proved, per loss, **from thresholds alone**	`C6_CHAIN_CERTIFICATE`
C7	B_0, β, the unit definition and the normalisation are frozen before the run	`C7_FROZEN_PAYLOAD_SEMANTICS`

Here

$$n_{\min}(\alpha, \delta) = \left\lceil \frac{\log \delta}{\log(1-\alpha)} \right\rceil$$

is the smallest audit population for which *zero* observed loss can certify risk $\leq \alpha$ at level δ; for example, $n_{\min} = 59$ at $(\alpha, \delta) = (0.05, 0.05)$.

C3 records not only the number of audit units but also their identities. If membership in U_k could depend on a candidate's own output, the resulting truncation would become correlated with the candidate's test statistic, and the conditional argument used in Theorem 1 would no longer apply.

Three execution rules are important.

Timing of the checks. A clause may be discharged statically, from design data, or later against the audit metadata $\mathcal{A}$. Once any candidate-dependent test-fold quantity has been opened, however, $\mathcal{C}$ may only verify declarations that were already frozen. It may not filter or reshape the tested family using information observed there, because the family would then depend on its own test-fold behaviour.

Use of audit metadata. Under `exclude_unit`, Z_k is label-derived, so the design stage does not test C3 at all. Instead it applies a **design-time support proxy**: the same threshold $n_{\min}(\alpha_k, \delta_m)$, evaluated at a preregistered design reference count determined only from the design fold. Being a design-fold quantity, the proxy may be used to shape the family; being one, it is not the realised audit population and discharges no premise, so it is never called C3.

C3 itself is evaluated only after registration, against $\mathcal{A} = \{Z_k,\ n_k,\ \text{membership digests}\}$, at the provisional share before the cover is built and again at the registered share (§3.5). At that point the audit may only verify or **refuse** the frozen procedure: it can refuse the whole procedure before the cover is built (R2) or make a registered chain terminal after it (T4), and it can never add a candidate, re-plan a cover, change an order, move an error share, or switch to another design. Candidate-dependent quantities remain closed: $\ell_k(\pi, u)$, $\widehat{\theta}_k(\pi)$, $p_k(\pi)$ are not read until the family and its testing structure have been frozen. The resulting invariant, which is what A1 turns on, is

$\mathcal{A}$ may only turn a pre-fixed procedure off; it may never change what procedure will be run.

Figure 1 represents the audit stage accordingly: it is a verification step with a refusal path, not a second design stage.

C4 is data-independent. A universal bound

$$|r_a(x)| \le B_{\max} \qquad \forall x$$

cannot be established from an empirical maximum. It must follow from the acquisition construction itself — for example, fixed resolution, encoder bounds, maximum crop dimensions, or an equivalent deterministic argument. The engine refuses a candidate whose declared bound exceeds $B_{\max}$; that the declared bound comes from such a construction argument, rather than from an observed maximum, is a registration requirement rather than something the engine can check. An observed payload maximum may be reported only as a non-certifying diagnostic.

3.4 The certification order and the chain cover

Definition 1. $\pi \succeq_{\text{cert}} \pi'$ iff a, $(t_{\text{low}}, t_{\text{high}})$, B_0 and the stratification coincide, and

$$b_\pi \le b_{\pi'}, \qquad A_\pi \le A_{\pi'}, \qquad v_\pi(\sigma) \ge v_{\pi'}(\sigma) \quad \forall \sigma.$$

The relation is a data-free partial order on the policy algebra.

Lemma 1. If $\pi \succeq_{\text{cert}} \pi'$, then for every input x,

$$\text{alarm}_\pi(x) \supseteq \text{alarm}_{\pi'}(x), \qquad \text{response}_\pi(x) \supseteq \text{response}_{\pi'}(x).$$

Lemma 2. Under the declared antitonicity, $\pi \succeq_{\text{cert}} \pi'$ implies $\ell_k(\pi, u) \le \ell_k(\pi', u)$ for every unit and every constrained label-dependent loss. Hence

$$R_k(\pi) \le R_k(\pi'), \qquad \widehat{R}_k(\pi) \le \widehat{R}_k(\pi')$$

on any fold, while

$$C_u(\pi) = C_u(\pi'), \qquad Z_{B,u}(\pi) = Z_{B,u}(\pi').$$

Proof sketch (full proofs in Supplementary Information, M4.2). The acquisition mode and escalation band are identical, so the escalation and edge-alarm sets coincide. On escalated inputs $v_\pi \ge v_{\pi'}$ raises the verifier score, and $A_\pi \le A_{\pi'}$, $b_\pi \le b_{\pi'}$ give the two decision-set inclusions. The declared aggregation is non-increasing in each frame indicator, with fixed non-negative weights and a policy-independent empty-case rule; the unit loss is therefore non-increasing, and averaging preserves the order. The payload quantities are equal because the escalated set — the frames that incur transmission cost — is identical. □

The order is thus *proved for every input before any data are read.* The family is partitioned into M chains using a deterministic cover of the admissible comparability DAG, whose edges are the pairs of Definition 1 that also carry the loss and inference certificates (C6). The implementation computes an exact minimum chain cover [31], via bipartite matching [32], for comparability groups (candidates that agree on a, the

escalation band, B_0 and the stratification) of size at most 400; larger groups use the deterministic greedy fallback.

Cover minimality affects yield, not validity. Any partition into $\succeq_{\text{cert}}$-chains fixed before the test fold gives the guarantee; M determines only the multiplicity price. Chain shares (δ_m) satisfy $\sum_m \delta_m \leq \delta$ and are preregistered, keyed by chain signature, and bound to the identity of the cover. They therefore cannot be selected after observing the p-values or reassigned to a different cover.

3.5 Procedure and guarantee

The design fold is processed in the following order. **(1)** Build the candidate family under the preregistered rule. **(2)** Discharge every candidate-level clause of $\mathcal{C}$ *before any cover exists*, at a provisional uniform share, with C3 represented at design time only by the support proxy of §3.3; a candidate that fails is removed from the family and its reason code is recorded. **(3)** Compute the cover, within-chain order and δ-plan **over the surviving candidates only**, then freeze and register them together. **(4)** Re-check each survivor at the share under which it will actually be tested. At execution, the provisional and actual-share checks are evaluated against the realised audit metadata: under A4 and the uniform provisional share, a C3 failure before cover construction is family-wide (R2), whereas a post-cover failure is terminal for the affected chain (T4); neither path allows $\mathcal{A}$ to select or reshape the cover (Supplementary Information, M5.2).

The test fold is then opened. Each chain is traversed from its $\succeq_{\text{cert}}$-greatest member downwards, with

$$p_{\text{joint}}(\pi) = \max_k p_k(\pi).$$

Testing stops at the first candidate for which $p_{\text{joint}}(\pi) > \delta_m$. A preregistered utility is minimised only over the resulting certified set.

Steps (2) and (3) cannot be interchanged. Filtering before the cover is constructed ensures that the returned set is an up-set of a *registered* chain. If an ineligible member were removed from an already frozen chain, the certified remainder could contain π' while a safer policy $\pi \succeq_{\text{cert}} \pi'$ ahead of it had been dropped, contradicting Theorem 1(ii).

Because the cover is constructed from the survivors, M is itself a function of that surviving family. For an exact minimum cover, deleting candidates cannot increase M, so under uniform allocation δ_m cannot decrease; in that case step (4) is a verification. No analogous monotonicity is assumed for the deterministic greedy cover used for comparability groups larger than 400. The greedy heuristic has no deletion guarantee, so the actual-share check in step (4) remains necessary. Once the cover has been frozen, **any** failure — a share-level re-check, a chain certificate, or an evaluated unit set that does not match the audited one — is terminal for the entire chain, and its share is not redistributed. Thus $\sum_m \delta_m \leq \delta$ continues to hold because shares are allocated once over the surviving family and are never reallocated.

Assumptions.

A1 (split independence). The family, cover, order and shares are measurable functions only of the design fold and declared constants. Audit metadata $\mathcal{A}$ — the

audit indicators Z_k, the counts n_k and the membership digests — **may only turn a pre-fixed procedure off; it may never change what procedure will be run** (§3.3).

A2-V. Property (V) of $\mathcal{I}$ holds for every constrained loss. The **declared units**, not frames, are **independent and identically distributed (i.i.d.) draws** from the target unit law, as required by the built-in HB construction; arbitrary dependence is allowed within a unit.

A2-O. Property (O) holds for every constrained loss in every chain containing more than one member. A singleton requires no (O), because there is no later candidate behind a stopping decision.

A3. Each constrained chained loss satisfies its declared loss-wise antitonicity condition.

A4. For every k, n_k and membership of U_k are functions only of the split and labels and are identical across the candidate family.

The guarantee is distribution-free in the loss and the policy, but it is not assumption-free: it relies on the unit-level i.i.d. sampling model in A2-V, which is a property of the declared split rather than a theorem (§6).

Lemma 0 (the audit population may be conditioned on). Assume the units are i.i.d. and Z_k is a function of the unit alone (C3/A4). Conditional on n_k, the audit units are i.i.d. draws from the law of $U \mid Z_k(U) = 1$. Therefore $\{\ell_k(\pi, u)\}_{u \in U_k}$ are i.i.d. in $[0, 1]$, and $p_{\mathrm{HB}}(\widehat{\theta}_k; n_k, \alpha_k)$ is valid for $H_{0,k} \colon R_k(\pi) > \alpha_k$ conditionally on n_k, hence unconditionally.

Proof. Selecting units with $Z_k = 1$ is a unit-wise filter that depends neither on π nor on any other unit. Conditional on the number retained, those units are therefore i.i.d. from the conditional law, and their losses are i.i.d. in $[0, 1]$. HB validity holds at every value of n_k, and averaging over n_k gives the unconditional statement. □

C3 is needed for this conditioning argument. If Z_k depended on a candidate's own output, both the conditioning event and the estimand R_k would vary across candidates. Lemma 2 and the union bound in Theorem 1 would then compare different hypotheses. The common *membership digest* across the family checks the observable part of this requirement; the family-wide comparison is carried out under C5 (`C5_SPLIT_INDEPENDENCE`).

Lemma 3 (lossless stop). Under A2-O, with n_k and α_k constant along a chain by C3/A4, if the sequence stops at $\pi^{(j)}$, then

$$p_{\mathrm{joint}}(\pi^{(i)}) \geq p_{\mathrm{joint}}(\pi^{(j)}) > \delta_m$$

for every later $i > j$. Hence no untested member would have been certified.

Proof. (O) makes each p_k non-decreasing along the chain, and a maximum of non-decreasing functions is non-decreasing. For the built-in pair, (O) follows by composing Lemma 2 with the monotonicity of p_{HB} in empirical risk at fixed (n, α). □

Theorem 1. Under A1, A2-V, A3 and A4, with every returned candidate satisfying $\mathcal{C}$:

(i)

$$\Pr\Big[\exists \pi \in \widehat{\Pi}_{\mathrm{ACE}},\ \exists k : R_k(\pi) > \alpha_k\Big] \leq \delta;$$

(ii) $\widehat{\Pi}_{\mathrm{ACE}}$ is an up-set of each chain, and under A2-O the stop discards nothing (Lemma 3);

(iii) under A2-O, two members of the same chain whose declared statistics agree for every constrained loss, at identical audit metadata, are either both certified or neither is;

(iv) neither family-design diagnostic of §3.6 appears in the proof, and part (i) makes no claim that $\widehat{\Pi}_{\mathrm{ACE}} \neq \varnothing$;

(v) a failed premise produces a typed refusal, never a certificate. After registration, a failed chain certificate is terminal for that chain, and its share is **not** redistributed.

Proof sketch (full proof in Supplementary Information, M5.3). Fix one chain. By A1, its order and δ_m are fixed before the test fold is opened, so we may condition on the design fold. Lemma 0 gives i.i.d. audit units from the relevant conditional law, with HB valid at each realised n_k and hence unconditionally. By Lemma 2, the constrained risks are non-decreasing along the chain. The members satisfying all constraints therefore form a prefix, and every member following that prefix violates at least one constraint k_0. An erroneous certification requires the sequence to reach the first such member and not stop there; equivalently $p_{\mathrm{joint}} \leq \delta_m$ at a candidate for which H_{0,k_0} is true. By A2-V and the intersection–union principle [33], that event has probability at most δ_m. The sequence stops at the first non-rejection, so any error in the chain implies that single event. A union bound over the M chains then gives

$$\Pr(\text{any erroneous returned certificate}) \leq \sum_{m=1}^{M} \delta_m \leq \delta.$$

For part (v), converting a failed registered chain into singleton tests at shares already allocated to the chain would spend δ_m once per member. Dropping the chain is therefore the option that preserves the registered error budget. □

Two points matter when interpreting Theorem 1. First, $p_{\mathrm{joint}} > \delta_m$ does not establish that a policy is unsafe; it only means that the available data, at the realised n and registered share, do not certify it. Second, fixed-sequence testing [5] remains valid under any prespecified order. The requirement that several candidates share an ACE chain only when a relation has been proved is therefore a design choice for the stronger stopping result, not a condition for basic family-wise validity. Lemma 3 states the additional benefit.

The same relation can also improve certification yield by reducing the number of chains.

Corollary 1 (fewer chains never return less). Fix the frozen family and the realised audit metadata, and consider executions in which no registered chain is made terminal after the cover is frozen. Under uniform allocation $\delta_m = \delta/M$ the returned set is $\widehat{\Pi}_M = \{\pi : p_{\mathrm{joint}}(\pi) \leq \delta/M\}$, so for two admissible covers with $M_1 < M_2$ one has $\widehat{\Pi}_{M_2} \subseteq \widehat{\Pi}_{M_1}$, and an admissible cover of minimum size weakly maximises the certified set by inclusion. A cover is *admissible* when every multi-member chain uses only comparability edges that the loss and inference contracts certify; these can be fewer than the structural edges of Definition 1, down to none: on the synthetic instance of §4, declaring one constrained loss without its monotonicity lemma raises the admissible minimum from 2 to 8. The inclusion is weak because $r^*(n, \alpha, \delta)$ is a step function of δ. The exact cover of §3.4 attains the admissible minimum; the greedy fallback need not. Proof and scope in Supplementary

Information, M5.5.

3.6 Two family-design diagnostics, outside the guarantee

The diagnostics in this section are defined for the built-in HB inference pair. A custom inference contract must provide an analogous rejection boundary; otherwise these statuses are undefined.

Let

$$r^*(n, \alpha, \delta) = \sup\{r \le \alpha : p_{\mathrm{HB}}(r; n, \alpha) \le \delta\}$$

be the **certification boundary**, and let $g^* = \alpha - r^*$ be the corresponding finite-sample margin. A risk estimate can therefore satisfy the target level α without being small enough to certify at the available n and error share.

The count used in r^* determines when the quantity is available and what it may influence. We distinguish

$$r^{*,\mathrm{ref}} = r^*(n^{\mathrm{ref}}, \alpha, \delta),$$

computed from a **preregistered design reference count**, from

$$r^{*,\mathrm{cert}} = r^*(n^{\mathrm{cert}}, \alpha, \delta),$$

computed from the realised certify-fold audit count. The first is a design-fold quantity and is **the only one a selection rule or a diagnostic in this section may read**. The second becomes available only after $\mathcal{A}$ is opened and belongs to certification and refusal alone.

Under A1, these two quantities cannot be interchanged. With `exclude_unit`, n^{cert} is label-derived, so a family constructed or filtered using $r^{*,\mathrm{cert}}$ would depend on the certification fold. Study 2 is a historical example of this violation; §5 reports the original run and the replay audit.

To describe how much a varying policy coordinate can change a loss, let S be a preregistered **influence envelope** containing every event whose catch status that coordinate can affect. The smallest such envelope is its exact acting domain; a superset is also valid but gives a looser bound. For unit u,

$$q_k(u, S) = \frac{|E_{k,u} \cap S|}{|E_{k,u}|},$$

and on a design set D,

$$\widehat{\Gamma}_{k,D}(S) = \frac{1}{|D|} \sum_{u \in D} q_k(u, S).$$

For the event-averaged aggregation with equal event weights used in all studies here (`unit_mean_over_events`), any two policies π, π' in the same chain satisfy

$$\left|\widehat{R}_{k,D}(\pi) - \widehat{R}_{k,D}(\pi')\right| \le \widehat{\Gamma}_{k,D}(S).$$

Thus $\widehat{\Gamma}_{k,D}(S)$ is a deterministic bound on the empirical reach of the varying coordinate over D. It may estimate a corresponding population quantity, but it does **not** bound that population quantity.

The first diagnostic is defined **per chain and per loss**. For a preregistered design-side bound bar_k that the selection rule guarantees, define

$$h_k^{*,\mathrm{ref}} = \max\left(0,\ \mathrm{bar}_k - r_k^{*,\mathrm{ref}}\right).$$

A pair (C, k) is labelled `MARGIN_CAPABLE` when

$$\widehat{\Gamma}_k \geq h_k^{*,\mathrm{ref}}.$$

If a chain-level label is desired, it is the conjunction over its constrained losses. A chain may therefore be capable for one loss and not for another; that distinction is retained in the report.

The second diagnostic, `BOUNDARY_SPANNING`, records whether the design-fold slack

$$r_k^{*,\mathrm{ref}} - \widehat{R}_k^{\mathrm{des}}$$

changes sign across the chain.

Both diagnostics are declarations evaluated only on preregistered design quantities, and by Theorem 1(iv) neither appears in the guarantee. In particular, `MARGIN_CAPABLE` does not imply that any policy will cross the certification boundary, while $\widehat{\Gamma}_k < h_k^{*,\mathrm{ref}}$ does not imply that the chain will certify either all of its members or none of them.

3.7 Registration and provenance

A1 concerns *when* the procedure was fixed. Because $\mathcal{A}$ is opened only **after** the design has been frozen, the provenance trail contains **two receipts**, not one.

Registration receipt. Written before any certify-fold quantity is read, it fixes the frozen family, the cover and its identifier, the order, the δ-plan bound to that cover, every loss declaration with its lemma id and named p-value construction, the digest of the executable inference registry, the design-time diagnostics, the utility, and digests of the protocol, split, model output, weights, and analysis code. Everything quantified by A1 is recorded here.

Audit receipt. Created only afterwards, it records the realised audit metadata: the indicators Z_k, realised counts n_k, and membership digests. These values may only verify the already frozen procedure or refuse it; they cannot change the family, cover, order, shares, or branch. The audit receipt carries no design authority, and no design quantity is recomputed from it.

Execution re-derives every field of both receipts and refuses on any mismatch. A refused audit writes its own record, so refusals remain part of the provenance trail rather than appearing as missing executions.

Scope of the checks. The receipts are designed to detect changes to registered declarations, package code, and executable inference registrations between registration and execution. They do not defend against arbitrary mutation of the running interpreter or a malicious execution environment: code executing in the same process

can rebind a global used by a registered callable, and an in-process digest cannot exclude such behaviour. The claimed property is **reproducibility and detection of accidental or package-level drift, not tamper-resistance**.

3.8 Wildfire case: data and models

The wildfire studies of §5 operate on archived detector and verifier outputs; neither model was trained, fine-tuned, or rerun as part of the ACE-LTT certification experiments. The edge detector is a YOLO26n model [34] (Ultralytics 8.4.153), trained by the authors on the training split of FireSmoke-Clean v0.2; its frame score is the first-stage score s of §3.1. The verifier is Qwen2.5-VL-3B-Instruct [35], used zero-shot on frames that the detector could escalate. It receives either the frame with the detector's top boxes drawn on it (overlay) or a crop around the top box (crop), which are the two acquisition modes a, together with a fixed yes/no prompt, and its score is $g = \sigma(\ell_{\mathrm{Yes}} - \ell_{\mathrm{No}})$ from the log-probabilities of the first answer token, without sampling. The detector weights digest, the verifier model revision, the prompt text and its digest, and the inference settings are recorded with the outputs and bound into the frozen protocol lock (Supplementary Information, M13.8).

FireSmoke-Clean v0.2 is a deduplicated union of four public fire and smoke detection datasets: D-Fire [36], the ground-camera subset FASDD_CV of FASDD [37], Pyro-SDIS v1.0 [38] and FLAME2 [39]. Near-duplicate images within and across sources were linked by perceptual hashing with pixel-level verification, image embeddings and camera grouping, and each connected component of the resulting graph is one unit. Units, not frames, are assigned to the training, development, calibration and sealed test roles, so no near-duplicate group straddles two roles; the held-out unit pool was divided between calibration and the sealed test set, stratified by source.

Code availability

The frozen ACE-LTT implementation, historical code packages pinned by the Study 1 and Study 2 registrations, the registered analysis scripts, protocol files, Monte Carlo artefacts and the dataset-free synthetic reproduction are publicly archived in the ACE-LTT Reproducibility Package v1.0.1 on Zenodo (DOI: 10.5281/zenodo.23205684). The public GitHub repository is https://github.com/saydirasulov-alt/ace-ltt. Supplementary Information reports the cryptographic digests that identify the code versions, manifests and registered run records used in this study.

Use of artificial intelligence

During the preparation of this manuscript, the authors used AI to assist with language editing, wording refinement, manuscript organisation, and consistency checking. The authors reviewed and verified all AI-assisted content and remain fully responsible for the scientific content, analyses, results, references, interpretations, and final manuscript.

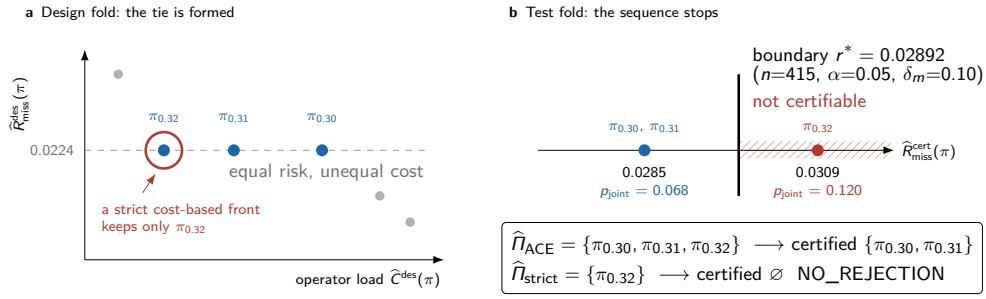

Fig. 2 The ordering witness. Panel **a**: the design fold, where the tie is formed and a strict cost-based front would discard two of its three members. Panel **b**: the test fold, where the discarded members fall inside the certification boundary and the kept member does not.

4 Controlled validation

4.1 A synthetic witness: the order is not presentational

We first use a dataset-free synthetic stream released with the code. It contains 900 units, two acquisition modes, and a declared seeded generator. No application data enter this example. Its constants were chosen to make the relevant failure and stopping behaviours visible, so it should be read as an illustration rather than empirical evidence.

On a b-grid with step 0.01, the design fold produces nine groups whose members have identical empirical risk but different operator load. Consider the group with $\widehat{R}^{\mathrm{des}} = 0.022417$, tested as a separate family with $M = 1$, $\delta_m = 0.10$, and $r^* = 0.02892$. Its three members have hand-off rates 0.1167, 0.1094 and 0.1082. On the test fold their empirical risks are 0.0285, 0.0285 and 0.0309, with $p_{\mathrm{joint}} = 0.068$, 0.068 and 0.120.

A strict cost-based front retains only the cheapest member, $b = 0.32$. That member fails, so the reduced family returns `NO_REJECTION`. The ACE order retains the tied group and tests from the safest member downward, certifying the first two policies. On this instance, across the full grid, the corresponding counts are 32 certified candidates under the ACE ordering and 4 after the strict cost reduction. **Figure 2** shows the same mechanism on the design and test folds.

The result depends on the constructed stream. Here the design-fold tie breaks on the test fold, while cost decreases as the policy becomes more aggressive. Under those two conditions, reducing the family by cost before testing removes candidates that later certify and retains the member that fails. Neither condition is guaranteed in general, so the example is only a witness. ACE consequently keeps cost out of the certification order instead of relying on behaviour that cannot be verified before the test fold is opened.

The ordinary execution path uses a separate 8-candidate family on the same stream ($b \in \{0, 0.15, 0.30, 0.45\}$ per acquisition mode): its candidates form $M = 2$ chains, the checklist is 9/9, and 5 of 8 candidates certify with both chains stopping early. Four fail-closed cases are exercised separately: an audit population below $n_{\min}$, unfrozen payload semantics, an inference-registry mismatch, and loss of chain compatibility when the miss loss is declared without its monotonicity lemma, which turns the cover

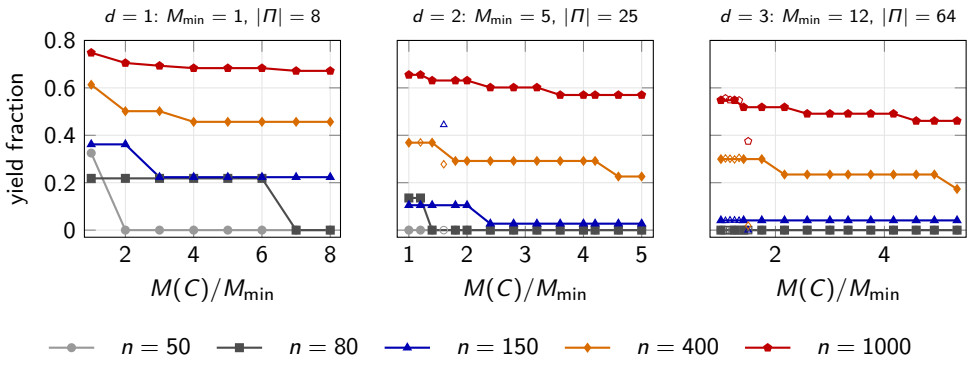

Fig. 3 Yield of the certified set against the chain count, in the preregistered slice (spread configuration, $L = 1$, lemma present). Yield is the share of non-null policies returned; a refused execution returns the empty set and counts as 0. Each point is the mean over 200 replications of the per-replication mean over covers with that M. Filled markers joined by lines: M occurs in every replication; open markers: M occurs only for some random admissible covers (counts in Supplementary Information, M15.5).

of this family from $M = 2$ chains into eight singletons and the uniform share from 0.05 to 0.0125. Supplementary Information, Appendix A, reports the full tables and reproduction digests.

4.2 A preregistered Monte Carlo check against known truth

The Monte Carlo experiment is a conformance check, not an additional proof of Theorem 1 or Corollary 1. Its purpose is to verify that the frozen engine behaves as those results require and to measure the resulting operating characteristics. The grid, generator, seeds, definitions of successful execution and refusal, and all decision rules were registered and digest-stamped before any run; the harness was written afterwards (registration and full results in Supplementary Information, M15). Truth is known in closed form: n i.i.d. units, Beta-distributed verifier scores in two strata, and $L \in \{1, 2\}$ constrained miss losses. Families vary $d \in \{1, 2, 3\}$ coordinates ($|\Pi| = 8, 25, 64$; $M_{\min} = 1, 5, 12$) under three truth configurations — a spread of risks, a least-favourable one with at least half the family in $(0.050, 0.060]$, and a global null — at $n \in \{50, 80, 150, 400, 1000\}$, $\delta = 0.10$ and $\alpha = 0.05$. Table 2 summarises the four blocks. The joint variation of up to three coordinates is the controlled counterpart of a multi-coordinate deployment; the application studies of §5 vary one coordinate per arm.

Conformance (MC-A). The run contained 36,000 replications. In the 18,000 replications with the monotonicity lemma present, each replication executed the minimum cover, ten refinements, ten random admissible covers, and the singleton cover. We observed no violation of the Corollary 1 threshold-set equality in 296,934 successful executions and no violation of its inclusion result in 2,306,485 pairs of successful admissible-cover executions. Covers with the same M also returned the same set within every replication.

Validity (MC-C). None of the 90 cells met the preregistered failure rule: a one-

Table 2 Controlled validation against known truth (preregistered R0; master seed 20261001). Violations are counted over every check made. FWER and nonzero-event intervals are two-sided 95% Clopper–Pearson; zero-event MC-D entries report the preregistered one-sided 95% upper bound.

block	quantity	result
MC-A (conformance)	A0 decision rule vs. engine, unit checks	0 / 5.87×10^8
	A1 inclusion (Corollary 1), successful cover pairs	0 / 2,306,485
	A2 threshold-set equality (Corollary 1), successful executions	0 / 296,934
MC-C (validity, $\delta = 0.10$)	cells meeting the preregistered failure rule	0 / 90
	FWER, worst cell (least-favourable, $d = 1$, $n = 50$)	0.070 [0.063, 0.077]
	FWER, pooled over 72 non-vacuous cells (descriptive)	0.0064 [0.0061, 0.0067]
MC-D (greedy, deletion)	E, registered grids $d = 1, 2, 3$ (each)	0 / 1000, $U_{0.95} = 0.0030$
	E, production regime ($\lvert\Pi\rvert = 512$)	1 / 1000 [0.00003, 0.0056]

sided 99% Clopper–Pearson lower confidence bound for the family-wise error rate (FWER) above δ. Each cell used 5000 replications. Eighteen cells are refused by C3 by construction and therefore return no certificate. The largest observed FWER was $0.070\ [0.063, 0.077]$, in the least-favourable configuration with $M = 1$, the smallest n, and risks concentrated near α. Theorem 1 supplies the bound δ; it does not identify which grid cell should be largest.

Yield (MC-B). In the preregistered figure slice (Figure 3), the observed yield gap between the minimum and the singleton cover was larger at higher d, which in this design also corresponds to a larger gap between $M_{\min}$ and $|\Pi|$. Here Δ is the number of additional policies the minimum cover returns, computed only over replications in which both executions succeed; a refused execution returns the empty set, counts as yield 0 and enters no Δ. Strict improvement is not guaranteed, and $\Delta = 0$ occurs in registered runs. At $d = 1$, $n = 80$ the minimum cover certifies while the singleton cover is refused by C3 (T4) — the support effect of a smaller M (Supplementary Information, M5.5, Remark (a)), in a cell registered for it.

Greedy cover (MC-D). We applied 1000 random deletions to a 512-candidate comparability group, which is above the 400-candidate cutoff and therefore uses the deterministic greedy fallback. In one run, deleting three candidates increased the greedy cover from 48 to 49 chains while the exact minimum remained 48. The fallback is thus neither guaranteed to be minimal on the reduced family nor monotone under deletion. The consequence is reduced yield, not loss of validity: Theorem 1 applies to any frozen admissible cover, including a greedy one, and step (4) (§3.5) catches an increase in M through the share-level re-check. Supplementary Information, M15 reproduces the instance from its registered seed.

5 Two exploratory studies, one refusal, one prospective test

We then applied ACE-LTT to an edge–cloud wildfire-detection cascade using archived detector and verifier outputs (§3.8). The policy uses a detector score s, a verifier score g under one of two acquisition modes, and an operator tier. Fire and smoke miss risk are constrained at $\alpha = 0.05$, with event-level losses averaged over units; the global error budget is $\delta = 0.10$. Units are the connected components produced by global deduplication. Studies 1–2 use two chains with $\delta_m = 0.05$. Study 3 consists of two arm-specific runs, each with $M = 2$, $\delta_m = 0.025$, and $\delta_{\text{arm}} = 0.05$, so the two

arms together use the full budget δ. Both losses use `exclude_unit`, so each certificate concerns units containing an event of the corresponding type. False alarm, verifier-call rate, and operator hand-off are not constrained risks in these studies; they enter only the preregistered utility used to choose an operating point. The design and test folds of §3 correspond to the artifact's **select** and **certify** folds. The separately named **sealed test set** is a third fold.

The evidential status of the runs is deliberately different. Studies 1 and 2 are exploratory analyses on a development split. Study 3 is prospective: its design was fully registered before any certify-fold quantity was opened. The sealed test set remains unopened. None of the application studies is presented as an empirical test of Theorem 1; Study 3 instead evaluates the capacity diagnostic of §3.6 prospectively. Each registered application chain varies only one policy coordinate.

Table 3: The four wildfire analyses of this section. Studies 1 and 2 are exploratory runs on a development split, audit 2A was refused before testing, and Study 3 is prospective. T and S are outcomes of the capacity diagnostic of §3.6, not certification results.

	family and chain coordinate	design-time diagnostics	outcome	interpretation
Study 1	veto authority λ (the value of v on its declared stratum, §3.1) per acquisition mode, 16 candidates	$\widehat{\Gamma} =$ 0.0035/0.0111 versus $g^* \approx 0.014$	5/16 certified; one chain stopped at its safest member ($p_{\text{joint}} = 0.1138$)	the varying coordinate had too little reach to make that chain informative
Audit 2A	first proposed Study-2 rule	$\widehat{\Gamma} =$ 0.0072/0.0127 versus $g^* =$ 0.0141/0.0145	**refused before testing**; nothing registered	the design filter rejected the proposed family
Study 2	response authority b per mode, λ fixed, 16 candidates	both chains `BOUNDARY_SPANNING` (joint slack +0.0092 to −0.0049 and −0.0005); checklist 9/9	historical run returned **16/16**, worst $p_{\text{joint}} = 0.00012$ (see the provenance note)	exploratory only: selection read certify-fold audit counts (A1 breach), so the returned set is not backed by Theorem 1; non-discriminating on the run
Study 3	two arms, b (arm B) or λ (arm L), one coordinate per arm; 39 registered candidates (37 distinct tuples)	6 capable / 2 incapable of 8 at registration	C3 PASS ($n_{\text{cert}} =$ 1130/1114 $\geq$ 72); certified 20/27 (B), 12/12 (L); diagnostic **T 8/8, S 4/6**	a prospectively registered sufficiency prediction failed on two capable pairs

Study 1. Veto authority was the chain coordinate and acted only within a declared stratum. One chain returned a certified prefix; the other stopped at its first member. For that chain, the estimated reach was only about one quarter to four fifths of the margin required to span the certification boundary (§3.6). The run is retained as observed: the procedure executed as specified, but one chain carried too little variation to be informative.

Design audit 2A. The first proposed Study-2 rule minimised utility subject to $\widehat{R}^{\mathrm{des}} \leq r^*$ and then checked capacity. It selected a narrow region in which the first stage had already alarmed on almost every positive event, leaving little for the chain coordinate to change. The audit refused the proposal, so nothing was registered. This refusal was produced by the criterion used at that point in development; two later corrections changed the rule but not the data. The episode shows that the design filter can reject a family. It is not evidence that the current §3.6 diagnostic has been validated. Audit 2A predates the automatic refusal-record mechanism, so its evidence comes from the development trail rather than from the self-contained record described in §3.7.

Study 2. Both response-authority chains were `BOUNDARY_SPANNING` at registration: their most aggressive members lay on the non-certifiable side of the design-fold boundary. The historical test-fold run nevertheless returned all 16 candidates. Smoke risk shifted in one direction by 0.0132–0.0186, or 0.91–$1.28 \times g^*$, while the design-fold standard error at roughly 447 smoke-positive units was already $0.52 \times g^*$. This gives a limited conclusion:

> **Observation 1.** `BOUNDARY_SPANNING` is a function of design-fold point estimates; it does not determine, and need not predict, where the test-fold sequence stops.

The observation is limited to this single instance.

A provenance issue changes how Study 2 must be interpreted. The historical selection code computed r_k^* from **certify-fold audit counts**, so the thresholds and b-ladder depended on test-fold audit metadata, contrary to A1. Replaying the rule with design-only reference counts happened to produce the same family, ladder, and cover: $\Delta r^* = 2.2 \times 10^{-4}$, while the nearest decision margin was 5.5×10^{-4}, a factor of 2.45. That numerical agreement does not restore the required independence. We therefore treat the historical Study-2 run as **exploratory method development, not a certificate backed by Theorem 1**. Study 1 is unaffected because its thresholds and grid were literal and did not use an audit count (Supplementary Information, M12).

The Study-2 record also contains a separate refusal. Two declared strata had only 4 and 13 audit units, below $n_{\min} = 59$. No certification claim was made for those strata, and the refusal was recorded.

A post-selection diagnostic on the Study-2 fold compared the cascade with a single-stage baseline at matched thresholds. It was not a registered candidate or a decision rule and has no p-value. In that descriptive comparison, the cascade halved event-miss risk with no unit worse off. The baseline estimates were below $\alpha = 0.05$ but above $r^* = 0.0359/0.0355$, whereas the cascade estimates were below r^*. Because the comparison was made after selection, it retains Study 2's exploratory status (Supplementary Information, M14).

Study 3. The final study prospectively evaluated the §3.6 capacity statement. It

used a calibration fold not used in Studies 1–2. The family, cover, chain order, δ-shares, and capability bits were registered before any certify-fold quantity was opened, and the registered procedure was then executed once. C3 passed with 1130 fire and 1114 smoke audit units, both above $n_{\min} = 72$. Certification returned 20 of 27 candidates in arm B and all 12 in arm L, with no reason codes. The T and S results below refer to the capacity diagnostic, not to those certificates.

The reach bound

$$T: \qquad \mathrm{sp}_{\mathrm{cert},k} \leq \Gamma_{\mathrm{sel},k},$$

where sp is the realised endpoint span, held for all eight registered chain–loss pairs; two held vacuously, at zero realised span. The registered sufficiency prediction

$$S: \qquad \mathrm{CAPABLE} \Rightarrow \mathrm{sp}_{\mathrm{cert},k} \geq h^*_{\mathrm{ref},k}$$

held for **four of six capable pairs** and failed for both arm-L smoke pairs. For those two pairs, the registered design-time capacity exceeded $h^*_{\mathrm{ref}} = 0.016$, whereas realised span was 1×10^{-4} and 3.3×10^{-3}.

The registered outcomes are $\mathbf{T = 8/8}$ and $\mathbf{S = 4/6}$. The reach bound held on the certify fold for all eight pairs, including two vacuous zero-span cases, even though §3.6 provides only a same-fold bound and does not guarantee that transfer. The sufficiency prediction failed on two capable pairs. These are finite-sample observations from one fold: they have neither a p-value nor a population interpretation and do not enter the certificates of Theorem 1 (Supplementary Information, M13).

6 Discussion

The four application analyses answer different questions, so the number of policies returned in each run should not be used as a common measure of success. Study 1 used a valid procedure, but one chain had too little variation to be informative. Audit 2A stopped before testing. Study 2 was designed to produce a more informative family, yet its historical selection rule violated A1 and the resulting run remains exploratory. Study 3 is the prospective execution: it produced registered certificates and, separately, tested the §3.6 capacity diagnostic, with T holding in 8/8 registered pairs and S in 4/6 capable pairs. These distinctions would be lost if only the final number of certified policies were reported.

Study 2 also shows why certification and deployment choice should be reported separately. Its registered utility assigned a high cost to operator hand-off and selected an endpoint with hand-off 0.0000, even though operator-assisted policies were present in the historical returned set. The certified or returned family and the policy preferred for deployment are therefore different objects in this example.

Limitations. The guarantee is conditional on the sampling object declared by the study. Losses are assumed **i.i.d. across declared units**; arbitrary dependence is allowed within a unit. Global deduplication places near-duplicate frames in the same declared unit so that they cannot be split across folds, but it does not establish independence between different units. Two sequences recorded by the same camera

on the same day, for example, may still be dependent. The i.i.d. condition is an assumption of the split design, not a consequence of Theorem 1.

Studies 1 and 2 use a development split and are exploratory. Study 3 is prospective with respect to its registered unit split, but it is not a new-domain evaluation: 88.6% of calibration units come from the dominant source, and the smallest source contributes only two units. Its results apply to held-out units from the same deployment. We make no confirmatory claim about out-of-domain or live-deployment performance.

The formal construction permits several policy coordinates to vary jointly, and the controlled Monte Carlo study includes $d = 2$ and $d = 3$ families. The wildfire studies are narrower: within each registered arm, only one coordinate varies, and alarm authority A is not used as a chain coordinate. The deterministic payload cap in C4 is also exercised only in the synthetic example. Payload is not a constrained risk in the three application studies, so C4 does not bind those runs. Corollary 1 applies to successful executions under the uniform allocation used here and does not use error-budget recycling. For comparability groups larger than 400 candidates, the implementation uses the deterministic greedy cover. This preserves validity but may reduce yield relative to a minimum cover.

Observation 1 rests on a single counterexample. The released engine also has only one built-in p-value construction, although it enforces the inference contract $\mathcal{I}$ at declaration. A weighted or otherwise custom estimand would therefore need its own implemented and proved construction before it could be declared. We specified, but did not implement, an uncertainty-aware version of `BOUNDARY_SPANNING` based on simultaneous one-sided bounds over the design family. Neither point changes the certificates reported here: all registered constrained losses use uniform weighting, and `BOUNDARY_SPANNING` is outside the guarantee (Theorem 1(iv)).

Utility weights are study-specific. In particular, the cost assigned to operator handoff reflects a deployment preference, not a default of ACE-LTT. If hand-off itself is to be controlled statistically, it should be declared as a constrained risk rather than represented only through the utility.

Conclusion. ACE-LTT adds an auditable policy layer around existing distribution-free risk-control machinery. The policy semantics determine which comparisons are eligible for shared testing before the certify fold is opened; the resulting relation defines the multiplicity decomposition; executable checks refuse unsupported runs; and the registration record links each certificate to the procedure that produced it. Under the stated assumptions, this provides a reproducible way to move from a multi-stage operating policy to a certificate whose scope and premises are explicit.

Data availability. The study materials are deposited in the ACE-LTT Reproducibility Package v1.0.1 on Zenodo (https://doi.org/10.5281/zenodo.23205684). The archive includes the preregistrations, protocol files, manifests, run records, detector and verifier score outputs used in the reported analyses, Monte Carlo study outputs, and synthetic reproduction materials. The controlled-validation data are synthetic and can be regenerated with the archived code.

The sealed test set was not opened or analysed in this study, and its score rows are therefore not publicly available. Complete locked test-set files are available from the corresponding author on reasonable request.

Source images and labels used to build FireSmoke-Clean v0.2 are not redistributed. They are available through the original public datasets: D-Fire (https://doi.org/10.1007/s00521-022-07467-z), FASDD (https://doi.org/10.1080/10095020.2024.2347922), Pyro-SDIS v1.0 (https://huggingface.co/datasets/pyronear/pyro-sdis), and FLAME2 (https://doi.org/10.1109/ACCESS.2022.3222805). Model weights are not distributed.

Author contributions. N.S.: methodology, software, formal analysis, writing – original draft. Y.-I.C.: conceptualization, supervision, project administration, writing – review & editing. D.A.D.: validation, writing – review & editing. B.A.N.: validation, writing – review & editing. A.M.: validation, writing – review & editing. All authors reviewed the manuscript.

Additional information. Competing interests: The authors declare no competing interests.

Funding. There was no funding for this work.

References

[1] Angelopoulos, A. N., Bates, S., Candès, E. J., Jordan, M. I. & Lei, L. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics* **19**, 1641–1662 (2025). Preprint arXiv:2110.01052.

[2] Bates, S., Angelopoulos, A. N., Lei, L., Malik, J. & Jordan, M. I. Distribution-free, risk-controlling prediction sets. *Journal of the ACM* **68**, 1–34 (2021).

[3] Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L. & Schuster, T. Conformal risk control. *International Conference on Learning Representations (ICLR)* (2024). Preprint arXiv:2208.02814.

[4] Marcus, R., Peritz, E. & Gabriel, K. R. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika* **63**, 655–660 (1976).

[5] Westfall, P. H. & Krishen, A. Optimally weighted, fixed sequence and gatekeeper multiple testing procedures. *Journal of Statistical Planning and Inference* **99**, 25–40 (2001).

[6] Rosenbaum, P. R. Testing hypotheses in order. *Biometrika* **95**, 248–252 (2008).

[7] Dmitrienko, A., Offen, W. W. & Westfall, P. H. Gatekeeping strategies for clinical trials that do not require all primary effects to be significant. *Statistics in Medicine* **22**, 2387–2400 (2003).

[8] Bretz, F., Maurer, W., Brannath, W. & Posch, M. A graphical approach to sequentially rejective multiple test procedures. *Statistics in Medicine* **28**, 586–604 (2009).

[9] Goeman, J. J. & Solari, A. The sequential rejection principle of familywise error control. *The Annals of Statistics* **38**, 3782–3810 (2010).

[10] Edwards, D. & Madsen, J. Constructing multiple test procedures for partially ordered hypothesis sets. *Statistics in Medicine* **26**, 5116–5124 (2007).

[11] Meinshausen, N. Hierarchical testing of variable importance. *Biometrika* **95**, 265–278 (2008).

[12] Ramdas, A., Chen, J., Wainwright, M. J. & Jordan, M. I. A sequential algorithm for false discovery rate control on directed acyclic graphs. *Biometrika* **106**, 69–86 (2019).

[13] Taeb, A., Bühlmann, P. & Chandrasekaran, V. Model selection over partially ordered sets. *Proceedings of the National Academy of Sciences* **121**, e2314228121 (2024). Correction: PNAS 121(46), e2420905121, 2024.

[14] Laufer-Goldshtein, B., Fisch, A., Barzilay, R. & Jaakkola, T. Efficiently controlling multiple risks with Pareto testing. *International Conference on Learning Representations (ICLR)* (2023). Preprint arXiv:2210.07913.

[15] Farzaneh, A. & Simeone, O. Multi-objective hyperparameter selection via hypothesis testing on reliability graphs. *Advances in Neural Information Processing Systems (NeurIPS)* (2025). Preprint arXiv:2501.13018.

[16] Zecchin, M., Park, S. & Simeone, O. Adaptive learn-then-test: Statistically valid and efficient hyperparameter selection. *Proceedings of the 42nd International Conference on Machine Learning, PMLR* **267**, 74018–74036 (2025).

[17] Schuster, T. *et al.* Confident adaptive language modeling. *Advances in Neural Information Processing Systems (NeurIPS)* (2022). Preprint arXiv:2207.07061.

[18] Jazbec, M. *et al.* Fast yet safe: Early-exiting with risk control. *Advances in Neural Information Processing Systems* **37**, 129825–129854 (2024).

[19] Hou, Q. *et al.* Reliable LLM-based edge–cloud–expert cascades for telecom knowledge systems. *IEEE Transactions on Communications* **74**, 9713–9728 (2026). Preprint arXiv:2512.20012.

[20] Pona, E. *et al.* Calibrate-then-delegate: Safety monitoring with risk and budget guarantees via model cascades (2026). arXiv:2604.14251.

[21] Huang, J., Park, S., Paoletti, N. & Simeone, O. Reliable inference in edge–cloud model cascades via conformal alignment (2025). arXiv:2510.17543.

[22] Xue, N. *et al.* CR^2: Cost-aware risk-controlled routing for wireless device–edge LLM inference (2026). arXiv:2605.12001.

[23] Bedi, S. *et al.* CARE: A conformal safety layer for medical summarization (2026). arXiv:2606.08969.

[24] Wang, Z. *et al.* LEC: Linear expectation constraints for selection-conditioned risk control in selective prediction and routing systems. *Proceedings of the 43rd International Conference on Machine Learning, PMLR* **306**, 124814–124840 (2026). Preprint arXiv:2512.01556.

[25] Thomas, P. S., Theocharous, G. & Ghavamzadeh, M. High confidence policy improvement. *Proceedings of the 32nd International Conference on Machine Learning, PMLR* **37**, 2380–2388 (2015).

[26] Thomas, P. S. *et al.* Preventing undesirable behavior of intelligent machines. *Science* **366**, 999–1004 (2019).

[27] Kim, G. From forecasting leaderboards to deployment decisions: A fail-closed certification protocol (2026). arXiv:2606.24996.

[28] Bertinetto, L., Henriques, J. F., Albanie, S., Paganini, M. & Varol, G. Preface. *Proceedings of the NeurIPS 2020 Workshop on Pre-registration in Machine Learning, PMLR* **148**, i–i (2021).

[29] Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* **58**, 13–30 (1963).

[30] Bentkus, V. On Hoeffding's inequalities. *The Annals of Probability* **32**, 1650–1673 (2004).

[31] Dilworth, R. P. A decomposition theorem for partially ordered sets. *Annals of Mathematics* **51**, 161–166 (1950).

[32] Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**, 83–97 (1955).

[33] Berger, R. L. Multiparameter hypothesis testing and acceptance sampling. *Technometrics* **24**, 295–300 (1982).

[34] Jocher, G. *et al.* Ultralytics YOLO26: Unified real-time end-to-end vision models (2026). arXiv:2606.03748.

[35] Bai, S. *et al.* Qwen2.5-VL technical report (2025). arXiv:2502.13923.

[36] de Venâncio, P. V. A. B., Lisboa, A. C. & Barbosa, A. V. An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Computing and Applications* **34**, 15349–15368 (2022).

[37] Wang, M., Yue, P., Jiang, L. *et al.* An open flame and smoke detection dataset

for deep learning in remote sensing based fire detection. *Geo-spatial Information Science* **28**, 511–526 (2025).

[38] Pyronear Team. Pyro-SDIS dataset. Hugging Face, tag v1.0, commit dd86e151, https://huggingface.co/datasets/pyronear/pyro-sdis (2024).

[39] Chen, X. *et al.* Wildland fire detection and monitoring using a drone-collected RGB/IR image dataset. *IEEE Access* **10**, 121301–121317 (2022).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- 02ACESciRepSupplementaryInformation.pdf