Supplementary Information

Fail-Closed Certification of Multi-Stage AI Policies

Norkobil Saydirasulov, Young-Im Cho, D. A. Davronbekov, B. A. Nazarov, Abdinabi Mukhamadiyev

This document contains the full formal development supporting the manuscript: the policy object and its two decision sets, the loss and inference contracts, the eligibility contract and its fail-closed statuses, the certification order with its chain certificate and cover, the procedure and central theorem, the design diagnostics, registration and provenance, the threat model, the two arm-specific instances of the prospective study, the paired single-stage baseline, the preregistered Monte Carlo study, and a generic worked instance with its reproduction digests. Section numbers of the form Mn are referenced from the manuscript.

This is the **Supplementary Information** to the manuscript. Section 3 of the paper is a condensed core of what follows; everything here that the main text omits — complete proofs, the chain-cover implementation, package and data digests, release notes, refusal logs, the generic worked instance in full, the provenance record and the carried-forward v2.4 items — belongs to this document.

Formal development

Written method-first: it states the framework, its assumptions and its one central theorem independently of any application, and it cites the three registered fire/smoke studies only where an empirical illustration is needed. Every numerical value quoted here is measured: the generic ones by `cascade.ace_toy` (a dataset-free synthetic instance shipped with the code), the application ones by the registered runs — Studies 1 and 2 as recorded in `ACE_provenance_record.md`, Study 3 from the artifacts listed in M13.8.

Numbering: M1–M7 are the method, M8 registration and provenance, M9 the generic worked instance, M10 the positioning table, M11 the contributions, M12 what is *not* claimed, M13 Study 3 in full and M14 the paired descriptive baseline comparison.

M1 Problem formulation and the policy object

M1.1 The deployment object

A cascaded perception system makes several coupled decisions for each observation: what representation to transmit, whether to invoke a second stage, who may act on its output, and what resources that decision consumes. ACE-LTT therefore certifies the deployable policy rather than an isolated predictor.

A **policy** is

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

with the following coordinates:

coordinate	meaning	governs
a	**acquisition mode**: representation transmitted to the second stage	payload and verifier input
$(t_{\text{low}}, t_{\text{high}})$	**escalation band** on first-stage score s	when the second stage is invoked
$A \in [0,1]$	**alarm authority**: verifier threshold for autonomous alarm	alarm set
$b \le A$	**response authority**: lowest verifier score producing any response, including operator hand-off	response set
$v : \mathcal{S} \to [0,1]$	**veto / discount authority**, piecewise constant on declared stratification $\mathcal{S}$	evidence handling within strata
B_0	**payload budget level**	statistical payload constraint

The statistical budget coordinate B_0 is a policy coordinate; the deterministic transport cap $B_{\max}$ is a parameter of the protocol's acquisition contract and not of any policy:

$$\underbrace{R_B(\pi) = \Pr(C_U > B_0) \le \beta}_{\text{statistical; tested with a } p\text{-value (M2)}} \qquad \underbrace{|r_a(x)| \le B_{\max} \quad \forall x}_{\text{deterministic; property of the acquisition mode (M3, C4)}} .$$

Here $r_a(x)$ is the representation transmitted under acquisition mode a, and C_U is the declared per-unit payload normalisation. The first statement concerns a probability and is tested statistically. The second concerns every possible input and must follow from the acquisition construction — such as resolution, encoder bounds, or crop geometry — rather than from a maximum observed in data.

Let g denote the verifier score and

$$g_v = \min(1,\, g + v(\sigma(x)))$$

its value after the declared stratum-dependent authority. The full decision map is

$$d_\pi(x) \in \{\mathsf{silent},\ \mathsf{edge\text{-}alarm},\ \mathsf{verifier\text{-}alarm},\ \mathsf{dismissed},\ \mathsf{hand\text{-}off}\}.$$

M1.2 Two decision sets, separately governed

The policy induces two distinct decision sets:

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

and

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

A governs autonomous alarms, whereas b governs the broader response set. An operator hand-off is not an autonomous alarm, but it is also not a miss when the declared catch set is `response`.

Because $b \le A$,

$$\text{response}(\pi) = \{s \ge t_{\text{high}}\} \cup \{\text{escalated} \wedge g_v \ge b\}.$$

The alarm threshold A disappears from this expression. A loss whose catch set is `response` is therefore **structurally independent of** A; this is an algebraic property of the policy family, not an empirical finding from any study.

Declaration rule. Every constrained loss names its catch set explicitly. The ordering results in M4.2 prove the required inclusions separately for `alarm` and `response`, preventing one from being substituted silently for the other.

M1.3 What is being certified

Let U denote the **units**, the grouping that is never split across folds. In the application studies these are global deduplication components; more generally, a unit should contain observations that cannot reasonably be treated as exchangeable across folds.

For each constrained risk k, unit u contains events $E_{k,u}$ that must be caught at least once, and the per-unit loss satisfies

$$\ell_k(\pi, u) \in [0, 1].$$

Each loss also declares a **policy-independent audit indicator**

$$Z_k(u) \in \{0, 1\},$$

fixed by its empty-denominator rule:

- under `exclude_unit`, $Z_k(u) = 1$ iff u contains at least one event of type k;
- under `loss_zero`, $Z_k \equiv 1$;
- under `not_certified`, the declaration is refused.

Define

$$U_k = \{u : Z_k(u) = 1\}, \qquad n_k = |U_k|.$$

The estimand and its empirical counterpart are

$$R_k(\pi) = \mathbb{E}\big[\ell_k(\pi, U) \,\big|\, Z_k(U) = 1\big], \qquad \widehat{R}_k(\pi) = \frac{1}{n_k} \sum_{u \in U_k} \ell_k(\pi, u).$$

This conditioning is substantive. Under `exclude_unit`, U_k is label-derived, so both its membership and n_k are random, and the quantity the HB p-value is valid for is the risk *conditional* on membership, not the marginal risk over all units. The certified claim is therefore

> among units carrying an event of type k, the risk is at most α_k,

rather than a marginal claim over all units, and it must be reported that way. Lemma 0 in M4.2 establishes when this conditioning is valid and identifies C3 as the corresponding audit requirement. Under `loss_zero`, the conditioning event is trivial and R_k becomes the marginal unit-level risk — the same procedure, a different claim. In either case, certification concerns R_k, not individual frames.

M1.4 Guarantee at a glance

Given a policy family frozen before the test fold is opened, a chain cover derived from a **proved** certification order, and preregistered error shares whose sum is at most δ, ACE-LTT returns a certified set with three properties.

Validity. The probability that the returned set contains any policy whose true risk exceeds its declared level for any constrained loss is at most δ.

Lossless stopping. Within each valid chain, the returned policies form a prefix of the certification order, so no aggressive policy is returned while a safer one that dominates it is rejected; and under the chain-compatibility condition every policy left untested after a stopping event would also have failed the registered test.

Explicit scope. The family-design diagnostics do not enter the validity argument, and the guarantee does not imply that the certified set is non-empty, informative, or preferable for deployment.

Every premise used by this statement is represented by an executable check. If a required premise cannot be discharged, ACE-LTT returns a typed refusal rather than a certificate.

The formal guarantee is Theorem 1 in M5.3; its structural lemmas are developed in M4.2.

M2 Losses and their monotonicity contracts

M2.1 A loss is a declaration, not a formula

Each constrained loss is declared as

$$\ell_k = (\text{name},\ \alpha_k,\ \text{kind},\ \text{catch set},\ \text{aggregation},\ \text{weighting},\ \text{empty-denominator rule},\ \text{lemma id}).$$

The certificate is written against this declaration. Three fields require particular care.

Aggregation. Frame-level catch-set inclusion does not by itself imply monotonicity of a unit-level loss. An aggregation may participate in an ACE chain only when its required monotonicity lemma is available. The released implementation recognises `unit_any_positive_frame`, `unit_mean_over_positives`, and `unit_mean_over_events`. An aggregation without the required lemma receives singleton treatment rather than inheriting a chain order by assumption.

Empty-denominator rule. An aggregation containing a denominator is incomplete until its empty case is specified. Choosing that rule after observing labels could change which units enter the audit population. The supported rules are `exclude_unit`, `loss_zero`, and `not_certified`; all three are policy-independent, so the monotonicity lemma survives each, and all are fixed before testing. The first two define an estimand, whereas `not_certified` refuses certification when the empty case arises.

Weighting. `uniform`, or unit weights declared before the run and independent of policy output; a weighting that reacts to the policy's behaviour breaks the lemma. Policy independence is required for the monotonicity argument, but it is not sufficient for inference: the statistic and its p-value must also be valid for the same weighted functional.

M2.1a The inference contract

A loss declaration therefore specifies a pair rather than a statistic alone:

$$(\,\widehat{\theta}_k,\ p_k\,).$$

The first component is the statistic actually computed on the certify fold; the second is a p-value construction proved for that statistic. Keeping them together is what makes the weighting field safe, because the monotonicity lemma of M4.2 survives any non-negative policy-independent weighting while the p-value does not.

> **Inference contract $\mathcal{I}$.** Every constrained loss declares $(\widehat{\theta}_k, p_k)$ satisfying two properties.
>
> **(V) Validity.** p_k is valid for
>
> $$H_{0,k}(\pi)\colon R_k(\pi) > \alpha_k$$
>
> under the sampling model in A2-V.
>
> **(O) Chain compatibility.** p_k depends only on $\widehat{\theta}_k$ and the declared audit metadata $(n_k, \alpha_k, \dots)$, and

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

at fixed audit metadata.

The two requirements have different roles. (V) supplies Theorem 1(i); (O) supplies the lossless-stop result of Lemma 3. Property (O) is stated on the policy order rather than directly on the statistic because a custom statistic need not inherit the per-unit ordering proved in Lemma 2.

The released engine supplies one built-in pair:

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

the unweighted unit mean tested using the Hoeffding–Bentkus construction [1, 2]. Its validity uses n_k i.i.d. observations in $[0, 1]$; its chain compatibility follows from Lemma 2 together with monotonicity of p_{HB} in its first argument at fixed (n_k, α_k).

A valid construction that does not establish (O) may still certify its own risk, but it cannot share a multi-member ACE chain: a candidate could receive a smaller p-value than a strictly safer one, and the stop would then discard certifiable members. Its candidates are covered as singletons, or may follow a separately proved testing order. An aggregation without its required monotonicity lemma receives the same singleton treatment.

Consequence (weighting). The built-in construction requires uniform unit weighting,

$$w_{k,u} = \frac{1}{n_k}.$$

An unequal weighted estimand $\sum_u w_{k,u}\, \ell_k(\pi, u)$ is a different functional with different concentration behaviour, so the HB p-value cannot be transferred to it. Such an estimand can be declared only when the declaration supplies and names a p-value construction proved for it; absent that, the loss is not certifiable by the built-in engine and the candidate is refused. An estimand is never silently passed to a test proved for another functional.

Scope. The inference contract governs weights **across units**, because these enter the statistical procedure. Weights within a unit belong to the aggregation producing $\ell_k(\pi, u)$; they remain subject to the declared aggregation and its monotonicity lemma rather than to the built-in HB weighting rule. So the weighting field may shape what the risk means, but it may not change how the risk is tested without declaring the replacement. Assumption A2-V in M5.1 therefore requires validity for every constrained loss, while A2-O is additionally required for every constrained loss appearing in a multi-member chain.

Executable registration. The released engine enforces $\mathcal{I}$ when the loss is declared. The declared construction name must resolve against a read-only registry containing the implemented callable, the statistic it consumes, the weightings it is proved for, its (O) status, and a version identifier. A name that does not resolve is refused before family construction.

The source of each registered callable is digested. That digest is bound into registration and recomputed before any p-value is evaluated, yielding the checked path

$$\text{declared} \to \text{implemented} \to \text{registered} \to \text{unchanged at execution.}$$

A mismatch returns `I_INFERENCE_REGISTRATION`. A construction without certified (O) status receives no $\succeq_{\text{cert}}$ chain certificate and therefore falls to singleton chains under `I_CHAIN_COMPATIBILITY`.

The current release implements exactly **one** inference construction: the built-in HB pair. A weighted or otherwise custom estimand therefore requires extending the registry with a proved

construction before it can be declared. This is an implementation limitation, not an extension of the theorem.

The source digest also has a limited threat model. It pins the registered callable's source, not the complete transitive state of the running interpreter; for example, rebinding a module global called by that function need not change the callable's own source digest. M12 states this limitation explicitly. All registered studies and the synthetic instance of M9 use uniform unit weighting, so this limitation does not alter any certificate reported here. Wherever the rest of this Method writes $\sum_u w_{k,u}(\cdot)$, the built-in case is $w_{k,u} = 1/n_k$, and the general form is stated only to mark where a future weighted estimand would have to supply its own inference.

M2.2 Which losses may share a chain

Antitonicity (A3). ℓ_k is *antitone under catch-set inclusion* if, for every unit u and every pair of policies whose declared catch sets are nested pointwise, the larger catch set gives the smaller loss. For a loss used in a multi-member ACE chain, its declared antitonicity condition requires that enlarging the relevant catch set cannot increase the unit loss.

Miss-type losses satisfy this property under the declared aggregation lemmas. Content-of-response losses generally do not: false alarm, wrong-class response, precision, and operator load may worsen when more responses are added.

The consequence is narrower than saying that such a loss cannot be certified:

> A non-antitone loss may still be constrained and certified. Without a separately proved order, however, it may not share an ACE antitone chain. Its candidates are covered as singletons, and the multiplicity price is paid at cover-construction time and is visible in M before registration (M5.3(v)).

In the three application studies, false alarm, verifier-call rate, and operator hand-off are therefore excluded from the constrained family and used instead in the preregistered operating-point utility (M5.2, step 6), because there the question is which certified policy to deploy, not whether the guarantee holds. This is a modelling choice of those studies, not a framework-level prohibition.

Payload loss is a third case. With

$$Z_{B,u} = 1\{C_u > B_0\},$$

where C_u is the declared per-unit payload normalisation, the loss is equal along a chain rather than merely antitone, because acquisition mode a, the escalation band, and B_0 are fixed within that chain. It can therefore participate in the joint test without weakening the chain ordering.

M3 The eligibility contract and fail-closed status

M3.1 The contract

Before any p-value exists, each candidate must satisfy an executable predicate $\mathcal{C}$ with seven clauses. Each clause is a deterministic function of the protocol, the split and the labels — never of the candidate's behaviour on the certify fold. A candidate that satisfies $\mathcal{C}$ is *eligible*; *admissible* is reserved for relations, edges, chains and covers that satisfy the loss and inference requirements (M5.5).

clause	requirement	failure code
C1	**measurement coherence:** scores and transmitted bytes come from the same representation	`C1_MEASUREMENT_COHERENCE`
C2	**score contract:** shared detector-output digest, score floor, model identity, and no dry-run scores	`C2_SCORE_CONTRACT`
C3	**audit support and membership:** $n_k \geq n_{\min}(\alpha_k, \delta_m)$ and a digest identifying the units entering risk k	`C3_CERTIFIABILITY_AUDIT`
C4	**payload cap:** $\lvert r_a(x)\rvert \leq B_{\max}$ for every input, established by a static construction proof — never by an observed maximum	`C4_PAYLOAD_CAP`
C5	**split independence:** family, order, and shares depend only on the select fold and declared constants; audit metadata may refuse, never reshape (M5.1, A1)	`C5_SPLIT_INDEPENDENCE`
C6	**chain certificate:** the claimed ordering is proved, per loss, from thresholds alone	`C6_CHAIN_CERTIFICATE`
C7	**frozen payload semantics:** B_0, β, unit definition, and normalisation are fixed before execution	`C7_FROZEN_PAYLOAD_SEMANTICS`

For the built-in test,

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

is the smallest audit population for which zero observed loss can certify risk $\leq \alpha$ at level δ. At $(\alpha, \delta_m) = (0.05, 0.05)$, $n_{\min} = 59$. Below it, no amount of good behaviour on the data can produce a certificate, and ACE-LTT says so before it looks.

M3.2 Why membership matters

C3 records the identity of the audit units as well as n_k. If membership in U_k could depend on a candidate's own output, the candidate could change the population on which its loss is evaluated, and a chain could be truncated by an event that correlates with its p-values. The conditional argument in Lemma 0 and Theorem 1(i) would then no longer concern one common estimand.

The implementation therefore records a digest of the sorted unit identifiers and requires it to agree across the candidate family. A missing or inconsistent digest is a refusal, not a default.

M3.3 When the clauses are evaluated

Checking $\mathcal{C}$ merely "before the first p-value" is not enough. If a candidate were removed from the family on the basis of something observed on the certify fold, the tested family would be a function of the certify data and the conditional argument in the proof of Theorem 1(i) would no longer apply, even if the removal happened first in program order. The relevant distinction is therefore between folds, not program order.

Construction time. Static information and select-fold quantities may be used to construct the candidate family. Under `exclude_unit`, the future audit population is label-derived, so the design stage **does not test C3 at all**. It instead applies a preregistered design-time support proxy based on a reference count from the select fold. Being a design-fold quantity, the proxy may shape the family; being only a proxy, it discharges no premise and is never called C3.

C4 is stricter, and takes no data at any fold. A statement

$$\lvert r_a(x)\rvert \leq B_{\max} \qquad \forall x$$

cannot be proved by a sample maximum: $\max_{x \in D_{\text{design}}} |r_a(x)| \le B_{\max}$ is a statement about D_{design}, not about x. It must follow from the acquisition construction itself: fixed resolution, encoder bounds, maximum crop dimensions, framing bounds, or an equivalent deterministic argument. The engine refuses a candidate whose declared bound exceeds $B_{\max}$ (`C4_PAYLOAD_CAP`); 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. The empirical payload maximum remains available as a diagnostic that the construction bound is not absurdly loose, never as the cap.

Registration. The surviving family, cover, order, error shares, loss declarations, and inference declarations are frozen.

Execution. After registration, $\mathcal{C}$ may only re-verify the frozen declarations. Formal C3 is evaluated once against the realised audit metadata

$$\mathcal{A} = (Z_k, n_k, \text{membership digests}).$$

It may **refuse** the frozen procedure: it can make a candidate or an entire chain terminal. It may never add a candidate, rebuild a cover, change an order, move a share, or switch to another design.

Candidate-dependent certify-fold quantities remain unavailable until the testing structure is frozen:

$$\ell_k(\pi, u), \qquad \widehat{\theta}_k(\pi), \qquad p_k(\pi).$$

The 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.

No application study constrains payload. Studies 1 and 2 use $B_{\max} = \infty$, so C4 is vacuous in each, and Study 3 declares no payload loss for C4 to apply to. The synthetic instance in M9 exercises C4 directly: each escalated frame transmits exactly 22 KiB under a_1 or 41 KiB under a_2, **by construction**, against a declared transport cap $B_{\max} = 48$ KiB. Its design-fold empirical maxima of C_u, 17.6 and 32.8 KiB, are strictly below the construction bound, because the design fold never observed the worst case. That gap is the whole argument for the rule: a sample maximum would have "proved" a cap of 33 KiB that the system can exceed on the next input.

M3.4 Fail-closed means a status

A failed premise produces a reason code and no certificate. The refusal vocabulary includes `C1`–`C7` together with `NO_REJECTION` (the sequence stopped at its first member) and `EMPTY_FAMILY`.

A refused execution remains an auditable object in the same way a successful one is. Four such paths are exercised on the dataset-free instance of M9, in full in Appendix A.

M4 The certification order, chains, and multiplicity control

M4.1 Two orders

ACE-LTT distinguishes a bookkeeping order from the order used for certification.

The **structural order** $\succeq_{\text{str}}$ records relations useful for policy construction: raising t_{high} widens the uncertain band but *shrinks* the edge-alarm set, so the alarm and escalation inclusions

can hold together only at equal t_{high}. It makes no statistical claim and never determines a test sequence.

The **certification order** is

Definition 1. $\pi \succeq_{\text{cert}} \pi'$ iff acquisition mode, escalation band, B_0, and stratification coincide, and

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

This is a **partial order** on the policy algebra and contains no observed data. It is a claim about every possible input, provable from the thresholds alone.

M4.2 The chain certificate

Lemma 1 (two-set inclusion). If $\pi \succeq_{\text{cert}} \pi'$, then for every observation x,

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

Proof. The acquisition mode and escalation band are equal, so the escalation and edge-alarm sets coincide. On escalated observations, $v_\pi \ge v_{\pi'}$ gives $g_{v,\pi}(x) \ge g_{v,\pi'}(x)$; with $A_\pi \le A_{\pi'}$ this gives the verifier-alarm inclusion, and with $b_\pi \le b_{\pi'}$ the response inclusion. □

Lemma 2 (loss antitonicity). For a constrained loss used in a multi-member chain, under its declared antitonicity condition (A3),

$$\pi \succeq_{\text{cert}} \pi' \implies \ell_k(\pi, u) \le \ell_k(\pi', u)$$

for every unit u. Hence

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

on any fold, and $C_u(\pi) = C_u(\pi')$, $Z_{B,u}(\pi) = Z_{B,u}(\pi')$ for every u.

Proof. Lemma 1 supplies pointwise inclusion of the declared catch set. The registered aggregation lemma is non-increasing in each frame indicator with non-negative policy-independent weights, and the empty-denominator rule is policy-independent, so the unit loss is non-increasing. Averaging preserves it. Payload is unchanged because acquisition mode, escalation band, and B_0 are fixed, so the escalated set — the frames that pay — is identical. □

Lemma 3 below depends on Lemma 2 and on the chain-compatibility part of the inference contract, of which HB is the built-in instance; the procedure it refers to is written out in M5.2. Lemma 0, stated first, rests instead on the sampling model of A2-V together with clause C3 and assumption A4.

Lemma 0 (conditioning on the audit population). Assume units are i.i.d. and Z_k is a function of the unit alone and identical across the family (clause C3, assumption A4). Conditional on n_k, the units in U_k are i.i.d. draws from

$$\mathcal{L}(U \mid Z_k(U) = 1).$$

Hence the corresponding losses are i.i.d. in $[0, 1]$, and the built-in HB p-value is valid for

$$H_{0,k}(\pi) \colon R_k(\pi) > \alpha_k$$

conditionally on n_k, and therefore unconditionally.

Proof. Selection by Z_k is applied unit by unit and depends neither on the policy nor on any other unit, so it induces no dependence among the retained units. Conditional on retaining m units, those observations are i.i.d. from the conditional law, and their losses are i.i.d. in $[0, 1]$ because $\ell_k(\pi, \cdot)$ is a fixed measurable function of a unit. HB validity holds at every m, and averaging over the distribution of n_k gives the unconditional statement. □

Why C3 is the hypothesis and not hygiene. C3 is a hypothesis of the inference argument rather than bookkeeping. If Z_k could depend on the candidate's own output, the conditioning event — and with it the estimand R_k — would differ from candidate to candidate: Lemma 2 would compare two risks defined on different populations, and the union bound of Theorem 1 would combine hypotheses that are not about one common quantity. The same remark explains why the empty-denominator rule must be declared *before* any label is seen: it is the definition of the estimand, and choosing it afterwards would be choosing what was certified after seeing the answer.

> **Lemma 3 (lossless stop).** Suppose all constrained losses of a multi-member chain declare **chain-compatible** pairs (M2.1a, part 2), so that A2-O holds, with n_k and α_k constant across members by clause C3/A4. If testing stops at $\pi^{(j)}$ because
>
> $$p_{\text{joint}}(\pi^{(j)}) > \delta_m,$$
>
> then for every later $i > j$,
>
> $$p_{\text{joint}}(\pi^{(i)}) \geq p_{\text{joint}}(\pi^{(j)}) > \delta_m.$$
>
> No member that the stop leaves untested would have been certified had it been tested.

Proof. A2-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, chain compatibility is a property of the construction: by Lemma 2 the empirical risk is non-decreasing along the chain, and for fixed (n, α) the Hoeffding–Bentkus p-value is non-decreasing in the empirical risk, because both bounds it minimises are — the Bentkus term through $\Pr[\text{Bin}(n, \alpha) \leq \lceil n\widehat{r} \rceil]$ and the Hoeffding term through a KL-type exponent that shrinks as $\widehat{r}$ approaches α — and a minimum of non-decreasing functions is non-decreasing. □

Fixed-sequence testing does not require a monotone risk order for validity. Under an arbitrary ordering the stop is a genuine loss: it can land on a candidate that blocks safer, certifiable candidates queued behind it. Under $\succeq_{\text{cert}}$ the stop costs nothing, because everything behind it is provably no safer and empirically no easier to certify.

Lemma 2 is also where ACE-LTT differs from a testing order that is merely asserted: the order is proved for every input before any data is read, and the certificate object that records it names, per loss, the catch set, the aggregation, the weighting, the empty-denominator rule and the id of the written lemma. If any of those is missing for any constrained loss, no certificate is issued and the candidates are not chained.

M4.3 Chain cover

A chain is a totally ordered subset under $\succeq_{\text{cert}}$. The candidate family is partitioned into M chains by a deterministic cover of the comparability DAG, computed from the policy algebra alone.

For a comparability group containing at most 400 candidates, the implementation computes an exact minimum chain cover [3] using bipartite matching [4]. Above that declared size it

uses a deterministic greedy cover, which remains a valid chain partition but is not guaranteed minimal.

All three registered studies lie in the exact regime — 16 candidates in two groups in each of Studies 1 and 2, and in Study 3 arm-wise groups of 27 and 12 candidates (M13.2), every one far below the 400-candidate threshold. The ordered chains and the cover algorithm identifier are hashed into `chain_cover_id`, so permuting members inside a chain changes the identifier and a cover produced by a different path is a different object.

Minimality is a yield property, not a validity property. Any valid pre-test partition into certification chains supports Theorem 1; what minimality buys is the size of M, and M is the multiplicity price, so a non-minimal cover is conservative — it loses certification yield, never validity.

On the synthetic instance of M9, $m = 8$ candidates cover into $M = 2$ chains at $\delta_m = 0.05$. Removing the required monotonicity lemma turns the same 8 candidates into 8 singleton chains at $\delta_m = 0.0125$ — a four-fold smaller per-chain share ($0.05 \to 0.0125$), with $\sum_m \delta_m = \delta = 0.10$ in both cases, and no fail-safe path in the second.

M4.4 The δ-plan

Shares satisfy

$$\sum_{m=1}^{M} \delta_m \leq \delta$$

and are preregistered, keyed to the chain signature and bound to the `chain_cover_id` they were written against, so a split can be neither chosen after the p-values are seen nor re-pointed at a different cover. A chain with no declared share is a failure rather than an invitation to allocate one during execution.

Uniform allocation $\delta_m = \delta/M$ is the declared default. Studies 1 and 2 use $M = 2$ and $\delta_m = 0.05$. Study 3 uses two arm-level instances, so the arm-wise cover gives $M = 4$ and $\delta_m = 0.025$, with

$$\delta_B = \delta_L = 0.05, \qquad \delta_B + \delta_L = 0.10$$

(M13.2); study-level validity follows by the union bound over the two arms, not from any independence assumption between them.

Scope note. Any bound on M quoted in the registration documents is a property of the registered equal-allocation designs, not a limit of ACE-LTT. The framework admits any preregistered (δ_m) with $\sum_m \delta_m \leq \delta$.

M5 The procedure, and the central theorem

M5.1 Assumptions

- **A1 (split independence).** $\widehat{\Pi}$, the chain cover, the intra-chain order and (δ_m) are measurable functions of the select fold, the protocol and the declared constants only.

  Audit metadata may veto, never choose. Write $\mathcal{A}$ for the audit indicators Z_k, the counts n_k and the membership digests on the certify fold. Under `exclude_unit` Z_k is label-derived, so the design stage does not test clause C3 at all. It applies a **design-time support proxy**: the C3 threshold $n_{\min}(\alpha_k, \delta_m)$ evaluated at a preregistered design reference count, which is a function of the select fold, and that is why the proxy may shape the family. Because a design reference count is not the audit population, the proxy discharges nothing

and is never called C3. **C3 itself** is tested once, against $\mathcal{A}$, and there it may only **refuse**: it can make a candidate or a whole chain terminal, and it can never add a candidate, re-plan a cover, move a share or switch a design. The invariant is

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

This is what keeps the theorem's conditioning simple: because nothing in $(\widehat{\Pi}, C_1..C_M, \delta_1..\delta_M)$ is a function of $\mathcal{A}$, the proof conditions on the select fold alone and needs no statement about validity conditional on the full audit configuration. That statement would not be free: conditioning on $Z_{k'}$ while testing loss k leaves the retained units of k independent but **not** identically distributed whenever $Z_{k'}$ is correlated with ℓ_k, so the conditional target becomes a random mixture rather than R_k. Lemma 0 licenses conditioning on one loss's own n_k, and that is all the proof uses. A refusal driven by $\mathcal{A}$ needs no such argument, because a refusal certifies nothing.

What may never be read before the family is frozen: $\widehat{\theta}_k(\pi)$, $\ell_k(\pi, u)$ and $p_k(\pi)$ — every candidate-dependent certify-fold quantity.

- **A2-V (validity).** Discharged for **every** constrained loss: for each policy π and loss k, the declared pair $(\widehat{\theta}_k, p_k)$ of $\mathcal{I}$ (M2.1a) gives a valid p-value for $H_{0,k}(\pi)\colon R_k(\pi) > \alpha_k$ from $\widehat{\theta}_k$, conditional on the audit population. **A2-V includes the sampling model stated below: the declared units are i.i.d. draws from the target unit law.** That is the hypothesis Lemma 0 consumes, and it is why Theorem 1's hypothesis list needs no further sampling assumption.

- **A2-O (chain compatibility).** Discharged **additionally** for every constrained loss of every chain with more than one member: $\pi \succeq_{\text{cert}} \pi' \Rightarrow p_k(\pi) \leq p_k(\pi')$ at fixed audit metadata. A singleton chain needs no A2-O, having nothing behind the stop. Theorem 1(i) rests on A2-V; Lemma 3, the lossless half of (ii), and (iii) rest on A2-O as well.

- **The sampling object both rest on.** ACE-LTT's built-in pair uses Hoeffding–Bentkus, whose construction requires independent observations bounded in $[0, 1]$. **Certification therefore treats the declared units, not the frames, as the independent sampling objects: the unit-level losses $\ell_k(\pi, u) \in [0, 1]$ are taken as** i.i.d. draws across the declared units** — independent, and identically distributed as the target unit law — while arbitrary dependence is permitted *within* a unit.** Lemma 0 uses the identically-distributed half, not only independence. This is the whole reason the unit is defined as the grouping that must never be split across folds — in the registered studies, a global deduplication component, so that near-duplicate frames and repeated views of the same scene stay on one side of the split and inside one summand. Everything downstream — n_k, $n_{\min}$, r^*, the audit membership digest of clause C3 — is counted in these units.

- **A3 (loss-wise antitonicity).** As declared in M2, with a written lemma per loss.

- **A4 (policy-independent audit population).** n_k and the membership of U_k are functions of the split and the labels only, identical across the family (clause C3).

M5.2 The procedure

1. **Select fold, family.** Build $\widehat{\Pi}$ under the preregistered selection rule. Check the family-wide preconditions (frozen payload semantics; the audit count and membership identical across candidates).

2. **Select fold, eligibility — before any cover exists.** Evaluate every candidate-level clause of $\mathcal{C}$ at a *provisional* uniform share, before any p-value. A candidate that fails leaves $\widehat{\Pi}$ entirely, with its reason codes.
3. **Select fold, cover and shares — over the survivors only.** Compute the chain cover, the intra-chain order and the δ-plan on the surviving family; run the pre-freeze checklist; freeze and register the cover identifier, the shares and the inference registration together.
4. **Select fold, re-check.** Evaluate $\mathcal{C}$ for each survivor at the share δ_m it will actually be tested at. This pass is authoritative (see the note below).
5. **Certify fold.** In each chain independently, walk from its $\succeq_{\text{cert}}$-greatest (most conservative) member downwards, computing the intersection–union p-value

$$p_{\text{joint}}(\pi) \;=\; \max_{k}\, p_k(\pi),$$

   and **stop at the first member with** $p_{\text{joint}}(\pi) > \delta_m$. Certify the prefix tested before the stop.
6. **Operating point.** Minimise a preregistered utility over the certified set.

When the clauses are evaluated. Steps 2 and 4 are logical stages of the certification procedure rather than fold-local data reads. Before registration, every clause that can be discharged from declared facts and design-fold quantities is checked; for C3, the design stage uses only the preregistered support proxy of main-text §3.3 and does not discharge C3. At execution, the certification call evaluates the realised audit metadata $\mathcal{A}$. In the provisional-share pass, A4 makes n_k, audit membership, and α_k common across the candidate family, while the provisional share is uniform. Consequently, a realised C3 failure at this stage is family-wide: it refuses the procedure before a cover is registered and cannot induce an alternative cover. After the cover is frozen, a failure at the actual-share re-check is terminal for the affected chain and cannot re-plan the cover or redistribute its share. Thus $\mathcal{A}$ may turn a pre-fixed procedure off but cannot choose or reshape the procedure. These two paths are recorded as R2 and T4 in M15.2.

Why steps 2 and 3 are in this order, and why step 4 is not a formality. Suppose the cover were computed over all candidates and ineligible members removed from inside their frozen chains. Take $\pi_1 \succeq_{\text{cert}} \pi_2$ in one chain with π_1 ineligible. Removing π_1 and testing π_2 can certify $\{\pi_2\}$, which is not an up-set of the registered chain $[\pi_1, \pi_2]$, and M5.3(ii) would be false as stated. Filtering first removes the case: after step 2 every member of every registered chain is eligible, so a registered chain is never partially populated. Step 4 then exists because M depends on which candidates survived, and $n_{\min}$ depends on $\delta_m = \delta/M$ under uniform allocation. For an **exact** minimum cover, deletion cannot increase M, so δ_m cannot fall and $n_{\min}$ cannot rise: there step 4 verifies. For the deterministic **greedy** cover used for comparability groups larger than 400, no such monotonicity is assumed and none is claimed: deletion may raise M, and under a registered non-uniform plan the share of a surviving chain can fall for either reason. Step 4 may therefore genuinely bind even under uniform allocation, and when it does the whole chain is terminal.

Failure after the cover is frozen. Any failure at or after step 4 — a re-check, a chain certificate, or an evaluated unit set that does not match the audited one in count **or** membership — is terminal for the whole chain: no member of it is tested or certified, and its share is **not** redistributed. Converting a failed chain into singletons at shares already allocated would spend δ_m once per member, so dropping it is the only option that leaves $\sum_m \delta_m \le \delta$ intact. Shares are allocated once, over the survivors, and never reallocated.

M5.3 Theorem

Theorem 1 (ACE-LTT). Under A1, A2-V, A3, A4, with every returned candidate satisfying $\mathcal{C}$ and the procedure of M5.2:

(i) Validity. $\Pr\left[\exists\, \pi \in \widehat{\Pi}_{\mathrm{ACE}},\ \exists k:\ R_k(\pi) > \alpha_k\right] \ \le\ \delta.$

(ii) Fail-safe ordering, and a lossless stop. $\widehat{\Pi}_{\mathrm{ACE}}$ is an up-set of each chain: if π is certified and $\pi' \succeq_{\mathrm{cert}} \pi$ lies in the same chain, then π' is certified — the procedure can never return an aggressive policy while rejecting a safer one that dominates it. Moreover, under A2-O for that chain, the stop discards nothing: every member left untested after it would itself have failed at δ_m (Lemma 3).

(iii) Tie completeness. Under A2-O, if two members of the same chain have equal declared statistics $\widehat{\theta}_k$ for every constrained loss, at identical audit metadata, then either **both are certified, or neither is certified**. In particular no cost-based tie-break applied before testing can be justified by the certification procedure.

(iv) Diagnostics are not validity. Neither the margin-capacity status (M7.2) nor the boundary-spanning status (M7.3) appears anywhere in the proof of (i)–(iii). Negating either status changes *which* family is offered and therefore what the procedure can return, but does not change the bound in (i). Conversely (i) carries no claim that $\widehat{\Pi}_{\mathrm{ACE}} \neq \varnothing$.

(v) Fail-closed, with no dynamic re-planning. If any clause of $\mathcal{C}$ fails, the affected candidate is removed before any p-value is computed and the run returns reason codes. The chain certificate is handled in two regimes, and only the first may change the cover:

- *Before registration.* If a pair cannot be certified — most often because a constrained loss carries no written monotonicity lemma — those candidates are simply not chained. The cover is built with them as singletons and the δ-plan is written against **that** cover, so the multiplicity price is paid up front and is visible in M.
- *After registration.* A chain-certificate failure at execution is terminal for that chain: the chain is dropped, none of its candidates is tested or certified, and its share is **not** redistributed.

In no failure path is a certificate issued on an unverified premise, and in no failure path does the realised $\sum_m \delta_m$ exceed δ.

Proof of (i). Condition on the select fold. By A1 the family, the cover, every intra-chain order and every share are then deterministic, and by Lemma 0 the audit units of each loss k are i.i.d. draws from $\mathcal{L}(U \mid Z_k(U) = 1)$ with $p_{\mathrm{HB}}(\cdot; n_k, \alpha_k)$ valid for $H_{0,k}$ conditionally on n_k and hence unconditionally. Fix a chain $C_m = (\pi^{(1)} \succeq_{\mathrm{cert}} \cdots \succeq_{\mathrm{cert}} \pi^{(J)})$. Let

$$j^\star \ =\ \max\{\, j:\ R_k(\pi^{(i)}) \le \alpha_k \ \ \forall k,\ \forall i \le j \,\}$$

(with $j^\star = 0$ if even $\pi^{(1)}$ violates a constraint). By Lemma 2, R_k is non-decreasing along the chain, so $\{\pi^{(1)}, \ldots, \pi^{(j^\star)}\}$ is exactly the set of members satisfying every constraint, and every member after it violates at least one. The procedure returns a member with a violated constraint only if it reaches index $j^\star + 1$ and does not stop there, i.e. only if $p_{\mathrm{joint}}(\pi^{(j^\star+1)}) \le \delta_m$. Since $\pi^{(j^\star+1)}$ violates some constraint k_0, H_{0,k_0} holds for it, and $p_{\mathrm{joint}} = \max_k p_k \ge p_{k_0}$ is a valid p-value for the union hypothesis by the intersection–union principle [5]; by A2-V and A4 the event $\{p_{\mathrm{joint}}(\pi^{(j^\star+1)}) \le \delta_m\}$ has probability at most δ_m. Because the sequence is fixed and stops at the first non-rejection, *any* error in chain m implies that single event, so $\Pr[\text{error in } C_m \mid \text{select fold}] \le \delta_m$. A union bound over the M chains with $\sum_m \delta_m \le \delta$, and expectation over the select fold, gives (i). □

(ii) The up-set claim is immediate: the procedure returns a prefix of each chain in $\succeq_{\mathrm{cert}}$ order. The lossless-stop claim is Lemma 3. *(iii)* Let $\pi \succeq_{\mathrm{cert}} \pi'$ lie in one chain with $\widehat{\theta}_k(\pi) = \widehat{\theta}_k(\pi')$ for

every constrained loss, at identical audit metadata. Because each p_k is a function of the declared statistic and that metadata alone (M2.1a), the two endpoints share one $p_{\text{joint}} = \max_k p_k$. By A2-O each p_k is non-decreasing along the chain, so every member *between* them carries the same p_{joint} as well. If $p_{\text{joint}} \le \delta_m$ the sequence passes through all of them; if $p_{\text{joint}} > \delta_m$ it stops at the first, so none is certified. A2-O is what rules out the remaining possibility — an intermediate member with a larger p-value stopping the sequence between two equal endpoints, certifying π and leaving π' untested. Note that "not certified" here means *stopped at or never reached*: a member after the stop is **untested**, not rejected, which is why the statement is "both or neither" rather than "certified or rejected together". *(iv)* No step of the proof reads a capacity or a slack. *(v)* is the construction of M5.2.

Why the post-registration regime of (v) must be terminal. Converting a failed chain into singletons *at the shares already allocated* would spend δ_m once per member: a chain of five members at $\delta_m = 0.05$ would consume 0.25, and (i) would be false. Re-planning the split at execution is no better, because the plan is bound to the registered cover and the failure is observed after the family was frozen. Dropping the chain is the only option that leaves $\sum_m \delta_m \le \delta$ intact without re-opening the registration.

Implementation note. The released code implements exactly this: on a failed chain certificate the chain is skipped, with `C6_CHAIN_CERTIFICATE` recorded, and no member of it reaches a p-value. The reachable route to singleton chains is the preregistration one, where the cover is built with the un-chainable candidates as singletons *before* δ is allocated — measured on the instance of M9 as $M : 2 \to 8$, $\delta_m : 0.05 \to 0.0125$, $\sum_m \delta_m = 0.10$ in both. Because the released cover is constructed with the same relation that later verifies it, a family assembled through the public path cannot reach the runtime branch at all; that branch exists to catch a chain assembled by other means, such as a registration record replayed against a different loss declaration.

M5.4 What the theorem does and does not answer

The reviewer question "is this an LTT wrapper?" is answered inside the proof, not beside it. Classical Learn-then-Test supplies A2 and the fixed-sequence argument. Everything that makes the bound *usable on a cascade* is supplied here: Lemma 1 and Lemma 2 (a testing order proved over all inputs from the policy algebra, with two separately governed decision sets), the chain cover that turns m candidates into M multiplicity units, the contract $\mathcal{C}$ that makes the premises executable and the failures typed, and clause C3/A4 without which the conditional argument in the proof is not available. Three of these are load-bearing for the statement itself:

- without Lemma 2, the chain order is an assumption rather than a proved relation and (ii) fails;
- without A4, the truncation point can correlate with the p-values and (i) fails;
- without the catch-set declaration, the certificate is about a set the system does not measure.

The cover is different in kind, and the difference is worth stating precisely rather than overclaiming. Fixed-sequence testing controls the family-wise error rate under **any** pre-specified ordering; monotone risk along the sequence is not needed for validity, and ACE does not claim otherwise. What the proved order adds is that the **stop has a semantic direction and costs nothing**: by Lemma 2 every untested member is no safer under the proved loss order, and by Lemma 3 every untested member would also have failed at δ_m — whereas an arbitrary ordering can stop on a candidate that blocks safer, certifiable ones queued behind it. Note what is *not* claimed: a member with $p_{\text{joint}} > \delta_m$ is not thereby shown to be unsafe. It is shown not to be certifiable from this data at this n and this share, which is a statement about the evidence, not about the policy. What the cover adds is the *shape* of the multiplicity budget: ACE's design rule is that candidates share one chain-level error budget only when a comparability relation between them has been proved, and incomparable candidates therefore receive separately

registered testing paths rather than being interleaved into one sequence. That is an ACE design rule, not a general statistical necessity.

M5.5 Proof and extensions of Corollary 1

Theorem 1 holds for any partition of the family into $\succeq_{\text{cert}}$-chains fixed before the certify fold is read. This section asks which of those partitions returns more, and answers it for uniform allocation.

Two graphs. Over the frozen family Π, the **structural comparability graph** is

$$G_{\text{str}} = (\Pi, E_{\text{str}}), \qquad E_{\text{str}} = \{(\pi, \pi') : \pi \succeq_{\text{cert}} \pi'\},$$

with $\succeq_{\text{cert}}$ as in Definition 1. It reads no loss, no aggregation, no lemma and no inference declaration. The **admissible certification graph** keeps only the structural edges that the statistical contracts license for shared testing:

$$E_{\text{adm}} = \{(\pi, \pi') \in E_{\text{str}} : \text{every constrained loss supplies its antitonicity lemma and its declared pair certifies (O)}\},$$

and $G_{\text{adm}} = (\Pi, E_{\text{adm}})$.

Always $E_{\text{adm}} \subseteq E_{\text{str}}$. The engine's chain-relation certificate is the executable test for membership in E_{adm}, and `C6_CHAIN_CERTIFICATE` and `I_CHAIN_COMPATIBILITY` are its refusal codes. A cover C is **admissible** when every multi-member chain of C uses only E_{adm} edges; write $\mathfrak{C}_{\text{adm}}$ for the set of admissible covers of Π and $M(C)$ for the number of chains of C. The structural relation comes from policy semantics; the admissible relation is what the executable contracts certify, and only the second may carry a shared error budget.

Proposition 3 (monotonicity of the returned set in the chain count). Let Π be the family frozen at step (3) of M5.2, with the audit metadata realised and fixed. Assume uniform allocation $\delta_m = \delta/M(C)$; A4 and C3, so that n_k and α_k are common to Π and independent of the cover; and A3. Call an execution under cover C **successful** if no registered chain is made terminal after the cover is frozen. Then for every $C \in \mathfrak{C}_{\text{adm}}$ whose execution is successful,

$$\widehat{\Pi}(C) = \{\pi \in \Pi : p_{\text{joint}}(\pi) \le \delta/M(C)\}, \tag{1}$$

and for two such covers with $M(C_1) < M(C_2)$,

$$\widehat{\Pi}(C_2) \subseteq \widehat{\Pi}(C_1). \tag{2}$$

Nothing in the statement depends on how C_1 and C_2 were computed.

Proof. Fix a chain of C and index its members along the order the procedure walks, from the $\succeq_{\text{cert}}$-greatest downwards: $\pi^{(1)} \succeq_{\text{cert}} \cdots \succeq_{\text{cert}} \pi^{(L)}$. By A3 and Lemma 2 the empirical risks are non-decreasing in the index. If $L \ge 2$, admissibility supplies A2-O for every constrained loss of the chain, so each p_k is non-decreasing in the index; a maximum of non-decreasing functions is non-decreasing, so p_{joint} is non-decreasing too. If $L = 1$ there is nothing to order. The engine stops at the first index j with $p_{\text{joint}}(\pi^{(j)}) > \delta_m$ — strictly, so a candidate at exactly δ_m is certified — and returns $\pi^{(1)}, \ldots, \pi^{(j-1)}$. By monotonicity the set $\{i : p_{\text{joint}}(\pi^{(i)}) \le \delta_m\}$ is an initial segment, and it is exactly $\{1, \ldots, j-1\}$: each of its members is reached before the stop, and by Lemma 3 nothing after the stop belongs to it. So each chain returns its own threshold set. Because the execution is successful, no chain is dropped, and the returned set is the union of these per-chain sets. The statistic $\widehat{\theta}_k(\pi)$ is a function of the candidate and the realised fold, and

n_k, α_k are common and cover-independent, so $p_{\text{joint}}(\pi)$ does not depend on C; under uniform allocation every chain carries the same $\delta/M(C)$, which gives (1). Then $M(C_1) < M(C_2)$ gives $\delta/M(C_1) > \delta/M(C_2)$, and (1) is a sublevel set of a fixed function, which gives (2). □

> **Corollary 1 (fewer chains never return less).** Let $M^{\text{adm}}_{\min} = \min_{C \in \mathfrak{C}_{\text{adm}}} M(C)$ and let $C_{\min}$ attain it. Under the hypotheses of Proposition 3, $\widehat{\Pi}(C) \subseteq \widehat{\Pi}(C_{\min})$ for every $C \in \mathfrak{C}_{\text{adm}}$ whose execution is successful.

Corollary 1 is a statement about set inclusion, not about power: there is no frequentist power function here, and, as remark (b) shows, the improvement need not be strict.

Remark (a): the support requirement moves the same way. Step (4) re-checks each survivor at the share it will actually be tested at, and C3 requires $n_k \geq n_{\min}(\alpha_k, \delta_m)$ with $n_{\min}(\alpha, \delta) = \lceil \log \delta / \log(1-\alpha) \rceil$, which is weakly decreasing in δ: for $\delta \in (0,1)$ both logarithms are negative, and $|\log \delta|$ shrinks as δ grows while $\log(1-\alpha)$ is fixed. At $\alpha = 0.05$:

M	1	2	4	8	16	32
δ_m	0.100	0.050	0.025	0.0125	0.0063	0.0031
$n_{\min}$	45	59	72	86	99	113

A smaller M therefore imposes a weaker support requirement, so a cover with fewer chains cannot be disadvantaged by C3.

Remark (b): the inclusion is weak. The rejection boundary is $r^*(n, \alpha, \delta) = k^*(n)/n$ with k^* integer-valued, and $n_{\min}$ is a ceiling. Both are step functions of δ, so reducing M may move neither, in which case (2) holds with equality. The size of the gain is a property of the instance.

Remark (c): exact versus greedy. Proposition 3 compares two given covers and is silent on how they were produced. What depends on the covering algorithm is whether the procedure *reaches* $M^{\text{adm}}_{\min}$, and how step (4) behaves.

- *In the exact regime* (comparability groups of at most 400 candidates, M4.3), the cover is a minimum chain cover computed by bipartite matching on G_{adm}, so the procedure attains $M^{\text{adm}}_{\min}$ and Corollary 1 applies to the cover it actually freezes. Deleting candidates cannot increase the size of a minimum cover, so under uniform allocation δ_m cannot fall and step (4) acts as a verification.
- *In the greedy regime* the frozen cover is admissible and Theorem 1 holds for it, but it carries no minimality guarantee. Proposition 3 still applies to the M it has; Corollary 1 does not apply to it. Nor is a greedy cover's chain count guaranteed to be monotone under deletion: removing candidates may leave it unchanged or raise it, and a raised count lowers δ_m and tightens C3, so step (4) may be a genuine filter there rather than a verification. M15.6 reports a preregistered instance: deleting three candidates from a 512-candidate family raises the greedy count from 48 to 49 while the exact minimum stays at 48.

All three registered studies and the generic instance lie in the exact regime (M4.3).

Remark (d): terminal chains. Equality (1) is stated for successful executions. If a registered chain is made terminal after the cover is frozen — by an authoritative re-check, a chain certificate, or an evaluated unit set that does not match the audited one — its members are returned by no chain and the returned set is a strict subset of the threshold set. This is the fail-closed semantics of Theorem 1(v), not a limitation of the covering algorithm, and it cannot be undone by re-covering: the share of a terminal chain is not redistributed.

The admissibility restriction is not vacuous. $M_{\min}(G_{\text{adm}})$ can exceed $M_{\min}(G_{\text{str}})$, and the generic instance of M9 contains a measured case. Its eight candidates vary one coordinate at fixed acquisition mode, escalation band, B_0 and stratification, and cover into two chains of four.

Remove the required monotonicity lemma from a constrained loss and change nothing else. G_{str} is unchanged, because Definition 1 contains no loss, aggregation or lemma. But the constrained losses are declared for the family, and without its lemma the aggregation no longer supports Lemma 2, so the built-in pair loses (O) — for that pair, (O) *is* Lemma 2 composed with the monotonicity of p_{HB}. Every structural edge then fails the chain-relation certificate, $E_{\text{adm}} = \varnothing$, and every candidate is a singleton:

	G_{str}	G_{adm}
minimum chain cover	2	8
δ_m at $\delta = 0.10$	0.05	0.0125
$n_{\min}$ at $\alpha = 0.05$	59	86
$\sum_m \delta_m$	0.10	0.10

A four-fold reduction in the per-chain share at an unchanged total budget, forced by the inference contract rather than by the policy order. A procedure that read comparability off the policy algebra alone and covered G_{str} would allocate $\delta/2$ to chains it is not entitled to test as chains.

Out of scope: non-uniform allocation. With shares that differ across chains, M does not order the individual δ_m, the returned set is no longer a single sublevel set, and neither (1) nor (2) holds in general.

M6 Necessity of tie-preserving selection

Multi-objective candidate filtering selects a front of the (risk, cost) trade-off on the select fold and tests along it; the literature admits many orderings on such a front. The implementation ACE-LTT rules out is a specific and natural one: a **strict cost-based front**, which removes all but the cheapest member of an empirical-risk tie before any testing happens. ACE-LTT takes the **weak** front — ties preserved — and then discards the cost information entirely: the testing order is $\succeq_{\text{cert}}$, never cost.

> **Proposition 1 (tie preservation is not an implementation detail).** There exist instances — realizable, not merely conceivable — on which a strict cost-based front returns $\widehat{\Pi}_{\text{strict-certified}} = \varnothing$ while $\widehat{\Pi}_{\text{ACE}} \neq \varnothing$ on the same data, same losses, same δ.

Measured witness (`cascade.ace_toy`, generic synthetic instance, seed 20260925; no application data). On a b grid of step 0.01 the select fold produces nine tie groups. In the group at $\widehat{R}^{\text{sel}} = 0.022417$:

member	select cost (operator load)	$\widehat{R}^{\text{cert}}$	p_{joint}	kept by the strict front
$\pi_{b=0.30}$	0.1167	0.0285	0.0675	no
$\pi_{b=0.31}$	0.1094	0.0285	0.0675	no
$\pi_{b=0.32}$	**0.1082**	0.0309	0.12	**yes** (cheapest)

with $r^*(n{=}415,\ \alpha{=}0.05,\ \delta_m{=}0.10) = 0.02892$. The strict front keeps exactly the member that fails, and returns `NO_REJECTION`; the ACE order keeps the group, tests it from the safest member downwards and certifies two policies. Over the whole grid the difference is 32 certified candidates (ACE) against 4 (strict front). **Figure 2** renders this instance.

The mechanism is **conditional**, and the conditions are worth stating exactly, because they are properties of an instance and not theorems. Whenever (a) candidates that tie on the fold that *chooses* them separate on the fold that *tests* them — different samples, so this is generic but not guaranteed — and (b) cost decreases monotonically with aggressiveness, a tie-break by

cost systematically exposes the most aggressive member of the equivalence class to the test, and can therefore discard members that would have certified. Neither condition is verifiable before the test fold is opened, which is why ACE removes the freedom rather than betting on it, and why the enabling mechanism is a property of this instance rather than a theorem. The witness is reproducible from the shipped package alone, with no application data.

M7 Family-design diagnostics: capacity and spanning

Theorem 1(iv) already separates these from validity. What follows is how a family is *designed*, and both statuses are preregistration objects computed on select-fold data.

Scope. Everything in this section is defined for the **built-in HB inference pair** (M2.1a), whose rejection boundary r^* is what the capacity and spanning statuses are measured against. A custom inference contract must supply an analogous boundary for its own construction; without one these diagnostic statuses are undefined for that loss. This is a scope limit on the diagnostics, not on Theorem 1, which by (iv) does not read them.

M7.1 The certification boundary and the finite-sample margin

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

r^* is the largest empirical risk on n units that a valid p-value can still reject at level δ. It depends on n, and only on preregistered quantities when that n is a **design reference count**. The distinction is not cosmetic, and it is the one Study 2 got wrong (M12). Two boundaries, two namespaces, used throughout: $r_k^{*,\mathrm{ref}} = r^*(n_k^{\mathrm{ref}}, \alpha_k, \delta_m)$ is a design-fold quantity and is the **only** one a selection rule may use; $r_k^{*,\mathrm{cert}} = r^*(n_k^{\mathrm{cert}}, \alpha_k, \delta_m)$ is post-$\mathcal{A}$ and belongs to verification, refusal and the certificate alone. A bare r^* in this document is the generic function, never a design input: wherever a design decision uses it, which n was supplied is stated. The gap g^* is the price of finite data: **feasible-on-select is not certifiable-on-test**. In Study 1 this was $r^* = 0.0360/0.0361$ against $\alpha = 0.05$, i.e. $g^* \approx 0.014$ — a candidate selected under "$\widehat{R}^{\mathrm{sel}} \le \alpha$" can be handed to the testing stage 0.014 above anything the data can certify.

M7.2 Capacity (main text §3.6)

The chain's varying coordinate acts only on part of the data. Let S be a preregistered **influence envelope** for that coordinate: any event set that *contains* every event whose catch status the coordinate can change. The minimal such envelope is the coordinate's exact acting domain — for a b ladder, events with no edge-alarmed positive frame and at least one escalated positive frame — and any superset is permitted, at the price of a looser bound: Proposition 2 needs only that no event outside S can change, so enlarging S keeps the bound valid and weakens it. Which envelope a study uses is part of its registration, and a study that registers a superset says so. S is defined from thresholds and observations, never from the chain's coordinate, so it is the same set for every member of the chain. Write the per-unit reach

$$q_k(u,S) \;=\; \frac{|E_{k,u} \cap S|}{|E_{k,u}|} \;\in [0,1].$$

Two different quantities must be kept apart, and an earlier version of this section conflated them.

Proposition 2 (finite-sample, deterministic). Let D be any fixed set of units and

$$\widehat{\Gamma}_{k,D}(S) \;=\; \sum_{u\in D} w_{k,u}\, q_k(u,S) \;\overset{\text{built-in case}}{=}\; \frac{1}{|D|}\sum_{u\in D} q_k(u,S)$$

computed with the loss's own weights, event definition and empty-denominator rule, for an aggregation in which a change in the catch status of one event moves the unit loss by at most $1/|E_{k,u}|$ (the event-averaged aggregation with equal event weights, `unit_mean_over_events`; for `unit_any_positive_frame` with several events per unit the bound can fail). Weights are uniform, $w_{k,u} = 1/|D|$, wherever the built-in HB inference is used (M2.1a). Then for any two members π, π' of the same chain,

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

Proof. Only events in S can change catch status between two members of the chain, and each such change moves the unit's declared aggregate by at most $1/|E_{k,u}|$; hence $|\ell_k(\pi,u) - \ell_k(\pi',u)| \le q_k(u,S)$ pointwise. Summing with the non-negative weights gives the claim. No sampling assumption and no expectation enters. □

Corollary 2a (population form). With $\Gamma_k^{\mathrm{pop}}(S) = \mathbb{E}_u[q_k(u,S)]$,

$$\left|R_k(\pi) - R_k(\pi')\right| \;\le\; \Gamma_k^{\mathrm{pop}}(S),$$

because the per-unit bound in the proof holds almost surely.

Which of the two the framework computes. The design-time capacity is $\widehat{\Gamma}$ on the select fold. It bounds the chain's reach on *that fold* exactly (Proposition 2) and is, under the sampling assumption of A2 with uniform weights, an unbiased estimate of $\Gamma_k^{\mathrm{pop}}(S)$ — but it is not itself a bound on $\Gamma_k^{\mathrm{pop}}(S)$. So the capacity status below is a statement about the design fold plus an estimate, which is one more reason it may inform family design and may never enter the validity argument (Theorem 1(iv)).

For a preregistered bar bar_k that the selection rule guarantees — $\widehat{R}_k^{\mathrm{sel}} \le \mathrm{bar}_k$ — the margin the chain would have to supply is

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

with $r_k^{*,\mathrm{ref}}$ and no other boundary flavour, by M7.1: the status decides what family is registered, so it may read only the design reference count. The chain is **MARGIN_CAPABLE** for loss k iff $\widehat{\Gamma}_k(S) \ge h_k^{*,\mathrm{ref}}$. With $\mathrm{bar}_k = r_k^{*,\mathrm{ref}}$ the requirement is 0 and the condition is vacuous — correctly so, since such a rule already hands over only certifiable-side candidates.

Two statements that must not be made. (a) `MARGIN_CAPABLE` does **not** mean any member is guaranteed to cross the boundary; it means only that the coordinate carries, in principle, as much capacity as the margin requires. (b) $\widehat{\Gamma}_k < h_k^*$ does **not** imply that the chain certifies all of its members or none: with $r^* = 0.036$ and $\widehat{\Gamma} = 0.007$, members at 0.034 and 0.040 differ by less than $\widehat{\Gamma}$ and still fall on opposite sides. An earlier version of this section asserted (b); it is withdrawn, and the capacity is used only as the upper bound of Proposition 2.

Provenance of the capacity. $\widehat{\Gamma}$ is label-dependent, so every capacity carries a source. A `design` capacity, computed on the select fold, may inform registration. A `realized` capacity (certify or test fold, after opening) is **diagnostic only** and can never accept a family, change a chain or change an order — admitting it would let the family adapt to the labels it is about to be tested on.

M7.3 Boundary spanning (main text §3.6)

Capacity bounds what the coordinate *could* do. Spanning asks whether, on the select fold, it *does* cross:

$$s_k(\pi) = r_k^{*,\mathrm{ref}} - \widehat{R}_k^{\mathrm{sel}}(\pi), \qquad s_{\mathrm{joint}}(\pi) = \min_k s_k(\pi),$$

$$\textbf{BOUNDARY_SPANNING} \iff \max_{\pi} s_{\text{joint}}(\pi) \geq 0 \quad \text{and} \quad \min_{\pi} s_{\text{joint}}(\pi) < 0.$$

A family failing this gives the fixed sequence nothing to select between: every member falls on the same side as the select fold sees it.

> **Observation 1 (boundary spanning is not an outcome predictor).** `BOUNDARY_SPANNING` is a function of select-fold point estimates. It does not determine, and need not predict, where the certify-fold sequence will stop.

This is stated as an observation and not as a proposition on purpose: it is established by counterexample, not by a probability argument. The counterexample is Study 2. Both chains were `BOUNDARY_SPANNING` at registration — the most aggressive member of each ladder sat on the non-certifiable side of the select-fold boundary (joint slack -0.0049 and -0.0005) — and on the certify fold **all 16 members certified**, worst $p_{\text{joint}} = 0.00012$. The mechanism was a one-directional fold-to-fold shift: on the smoke risk 0.0132–0.0186, i.e. 0.91–$1.28 \times g^*$, with a select-fold standard error already $0.52 \times g^*$ at ≈447 smoke-positive select units.

What that single instance supports is the negative claim in the observation, and no more. It does **not** establish a general law of the form "whenever the select standard error is comparable to g^*, the two disagree"; that would need a probability statement this draft does not have. The criterion did real work regardless — it refused the audit-2A family for a structural reason before any test — and its role in the framework is to filter designs, which Theorem 1(iv) already keeps clear of the guarantee.

A conservative successor is specified but **applied in none of the three registered studies**: `ROBUST_BOUNDARY_SPANNING`, requiring $U_{\text{safe}} < r^*$ and $L_{\text{aggressive}} > r^*$ with *simultaneous* one-sided bounds over the full select candidate family, or a separate design/select split. A nominal per-candidate interval will not do, since the thresholds are chosen from the same fold.

M8 Registration and provenance

The theorem's A1 is a claim about *when* things were decided. ACE-LTT makes that claim checkable rather than asserted.

Because $\mathcal{A}$ is opened only *after* the design is fixed, the trail is **two receipts**, and the split between them is the mechanical form of the refuse-only invariant.

- **Registration receipt (R1).** Written before any certify-fold quantity is read, a JSON record fixes: the frozen family with every policy's coordinates; the chain cover and its `chain_cover_id`; the intra-chain order; the δ-plan bound to that cover; every loss declaration including its lemma id *and* its named p-value construction; the `inference_registry_id` of the executable inference table; the design-time capacities, design reference counts and select slacks; the utility; and full 64-character SHA-256 digests of the protocol, the split file, the detector output, the detector weights and the analysis code. A digest of the declaration itself is recomputable from the record. Everything A1 quantifies over lives here.

- **Audit receipt.** Created only afterwards, it records the audit metadata $\mathcal{A}$: the indicators Z_k, the realized counts n_k and the membership digests, together with the digest of the registration receipt it verifies. Those values can only verify the already frozen procedure or refuse it; they cannot alter the family, the cover, the order, the shares or the branch. An audit receipt carries no design quantity, and no design quantity is ever recomputed from one.

Why the order matters. A single record that froze the family and the audit counts in the same act would be unimplementable under the refuse-only invariant: it would require $\mathcal{A}$ to be available while the design was still being chosen, which is exactly the dependence A1 forbids. The registered studies of §5 are reported against this two-receipt reading, and the one place the historical Study-2 selection rule departed from it is disclosed in §5 with its measured sensitivity margin.

- **Analysis-code identity.** `analysis_code_sha256` is taken over an explicit list of modules. Anything not on that list — the self-test, the post-registration diagnostic, the generic worked instance of M9 — cannot change it, which is why adding those did not disturb either registration. A release that changes a module *on* the list does change it, and then the affected study no longer verifies against the installed tree: it remains re-executable from the archived package its registration pins, which we ship. The table in Appendix A states which object is pinned to which package. That asymmetry is the mechanism, not a defect in it — a certificate is about the code that produced it.
- **Runtime-tree identity.** `MANIFEST.sha256` covers every shipped file with an explicit ignore list, so a stale module left behind by an overlay install is reported rather than silently imported. (This check is not decorative: it is what flagged the new module of M9 as an untracked addition during this work.)
- **Re-derivation on execution.** The execution stage re-derives thresholds, ladders, chains, cover id and design capacities from the registered declaration and refuses on any mismatch.
- **Refusals are records.** A design audit that refuses to register writes its own timestamped audit file. One earlier refusal (design audit 2A) predates that feature; the honest consequence is stated in the provenance document rather than papered over.

M9 A generic worked instance (summary; full instance in Appendix A)

To separate method from application, the entire path — family design, chain cover, budget allocation, certification, operating point, and the fail-closed refusals — is instantiated on a dataset-free synthetic stream released with the code: 900 units, two acquisition modes, a declared seeded generator, an 8-candidate family, one miss loss and one payload loss, and no application data of any kind. It is an **illustration, not evidence**: the generator is declared and seeded, and its constants were chosen so that the instance exhibits the behaviours the method is built around.

The one fact the main text needs from it is the ordering witness of M6, rendered in **Figure 2**. On an empirical-risk tie whose cheapest member fails on the certify fold while its tied, more conservative siblings pass,

> a strict cost-based tie-break certifies **0** policies; the ACE order certifies **2** — same data, same losses, same error budget.

Beyond that witness, the instance runs the nominal path to a certified set (5 of 8 candidates, both chains stopping early, $r^* = 0.02651$ against $\alpha = 0.05$), closes the pre-freeze checklist 9/9, and exercises four fail-closed refusals — an audit population below $n_{\min}$, unfrozen payload semantics, an inference registry that does not match the registered one, and the cover inflation that follows when a loss carries no monotonicity lemma ($M : 2 \to 8$, $\delta_m : 0.05 \to 0.0125$). **Appendix A** gives the full tables, the refusal outcomes, the capacity and spanning statuses under three declarations of the influence envelope, and the reproduction digests.

M10 Positioning

M10.1 The closest cascade works

Replaces the earlier two-column comparison. Columns are the closest works actually read (§2 of the main text); rows marked **prior art** are properties ACE-LTT does *not* claim. The scope of every other row is **the compared works**, and nothing more: "ACE contribution" below means *we found no counterpart in the works in these columns, which we read in full.* It is not a claim about the literature as a whole, which we have not surveyed exhaustively and do not claim to have.

axis	Pareto Testing	Hou et al. (edge–cloud–expert)	Calibrate-Then-Delegate	**ACE-LTT**
finite-sample certification by FST	yes	yes	yes	yes — **prior art**
certified set, then utility/cost selection	yes	yes	yes	yes — **prior art**
several constrained risks at once	yes	one alignment risk	one constrained, one optimised	yes, via $p_{\text{joint}} = \max_k p_k$ — **general idea prior**
a resource budget as a constrained risk	possible	cost is a free objective	yes (delegation budget)	yes — **prior-art ingredient**
structurally monotone testing order	no: estimated ordering from an optimisation split	**yes**, in the confidence threshold	yes in the budget-controlled variant; heuristic in the performance-controlled one	yes — **the property itself is prior art**
multi-coordinate policy algebra $(b, A, v, \dots)$	no	two-threshold grid	single-threshold policy	**ACE contribution; no counterpart in these works**
partial order $\to$ *computed* chain cover	no	fixed-ε grid factorisation	no	**ACE contribution; no counterpart in these works** — M is derived, not an axis count
alarm and response as separate decision sets	no	no	no	**ACE contribution; no counterpart in these works**
executable C1–C7, typed refusals, fold discipline	no	no	no	**ACE contribution; the one we defend hardest** (the fail-closed *idea* is prior art, see M12)
estimand $\leftrightarrow$ valid *and* chain-compatible p-value ($\mathcal{I}$)	no	no	no	**ACE contribution; no counterpart in these works**
validity $\neq$ informativeness $\neq$ deployment utility, formally	not formally separated	no	no	**ACE contribution; not formally separated in these works**
registration / provenance / refusal-as-record	no	no	no	**ACE contribution (engineering); preregistration itself is prior art**

Two consequences for how the paper must be written. First, the **statistical skeleton of Theorem 1(i) is not novel**: the first-true-null argument with a per-chain δ/M and a union bound appears in this literature already. What is new is binding it to a policy algebra, a derived cover, and executable premises. Second, "proved order" must not be sold as first-of-its-kind — structural monotonicity in a scalar threshold is established practice. The defensible chain is

multi-coordinate policy algebra $\to$ partial order $\to$ chain cover $\to$ loss-wise executable certificate.

Lemma 3 should therefore be presented as what a proved comparability *additionally* buys — a lossless stop — and not as an improvement to fixed-sequence validity, which needs no ordering assumption at all.

The honest one-line summary: ACE-LTT inherits its p-values, its fixed-sequence argument **and the use of a structurally monotone order** from this literature, and contributes the *structure* around them — a multi-coordinate policy algebra with two separately governed decision sets, a multiplicity geometry derived from that algebra rather than from a grid, executable premises that fail closed, and the explicit separation of validity from informativeness and from deployment preference.

M10.2 Structured multiple testing, structure-aware Learn-then-Test, and refusal semantics

The table of M10.1 compares ACE-LTT with the closest cascade works. This one places it against the works a reviewer is most likely to raise as prior art for the *structure*: where an order or graph over hypotheses comes from, what error it controls, and what a wrong structure costs. Every row was checked against the publisher's or proceedings' own record (October 2026).

work	structure over candidates	where the structure comes from	error controlled	cost of a wrong structure	relation to ACE-LTT
sequential rejection principle [6]	any monotone rejection scheme	input	FWER	n/a	**Theorem 1(i) is an instance; no new multiplicity principle claimed**
partially ordered hypotheses [7]	partial order of priority	input (trial design)	FWER	—	same object type, but the order is declared; ACE derives and certifies it
hierarchical / DAG testing [8, 9]	tree / DAG	input (variable clusters, ontology)	FWER / FDR	—	logical structure given by the domain
model selection over posets [10]	graded poset of models	input (model class)	false discoveries	—	poset is the model space, not a certified risk order
Pareto Testing [11]	Pareto front + p-value order	estimated on an optimisation split	FWER	power only	validity kept; stop not lossless
reliability-graph testing [12]	reliability DAG	estimated from data and prior beliefs	FDR (DAG-GER)	power only	closest in spirit; ACE's edges are proved, not estimated, and refused when a contract fails
CARE [13]	2-D threshold grid (τ, γ)	one data-independent total order (by $\tau + \gamma$)	risk control via LTT–FST	power only	multi-threshold families are prior art; one fixed path, not a certified partial order or chain cover
LEC [14]	one or two routing thresholds	calibration constraint	FDR given selection	—	scalar retention/routing thresholds
adaptive LTT [15]	none (e-processes)	—	FWER / FDR	—	different testing engine; complementary
early-exit risk control [16, 17]	scalar exit threshold	structural monotonicity	risk control	—	single coordinate, single network
learning to defer [18, 19]	learned rejector	training	none (consistency)	—	learns the deferral rule; certifies nothing
Seldonian / HCPI [20, 21]	one candidate after search	—	high-confidence safety test	—	**refusal semantics ("no solution found") are prior art**
conformal policy control [22]	mixing with a safe reference policy	likelihood-ratio threshold	expected constraint risk	—	one scalar dial, no partial order
testable learning [23, 24]	—	—	test-then-certify for learners	—	shares the *accept-or-reject the premise* stance; different object

Three consequences for the main text. First, ACE-LTT's order is not a new kind of multiple-testing structure; it is a new *source* for one — a label-free certificate from the policy algebra, restricted by the loss and inference contracts, which is why E_{adm} can be strictly smaller than E_{str} (M5.5). Second, estimated structures (Pareto Testing, reliability graphs) and a proved one buy different things: both keep validity; in ACE-LTT the proved order together with chain

compatibility establishes lossless stopping (Lemma 3) and threshold-set equality (Proposition 3), and these properties do not follow merely from an estimated order or reliability graph. Third, the chain-cover Bonferroni allocation is a deliberate choice of the simplest auditable allocation, not an optimum (M12).

M11 Contributions

These are the main text's C1–C4, stated at the supplement's level of detail. The labels match §1 of the main text one for one.

C1. We formalise cascade deployment as a **policy algebra with separately governed alarm and response sets** — jointly parameterising acquisition, escalation, alarm authority, response authority, veto and payload budget — and require every constrained loss to declare the decision set, aggregation, weighting and empty-case semantics its certificate refers to. Theorem 1 is then a single guarantee covering family-wise validity, fail-safe ordering and the explicit exclusion of the design diagnostics from the validity argument.

C2. We define **executable eligibility *and inference* contracts**. The eligibility contract (M3) prevents certification when measurement–cost coherence, audit support and membership, payload semantics, or policy-independent unit membership are violated — each with a typed refusal code rather than a degraded number, and each evaluated at a declared fold. The inference contract $\mathcal{I}$ (M2.1a) binds each estimand to a p-value construction proved valid for it and, inside a multi-member chain, compatible with the declared policy order; a stale or unmatched inference registration refuses before any p-value is computed.

C3. We derive a **label-free partial order over multi-coordinate cascade policies**, prove the decision-set inclusions for every input (Lemma 1) and require a written per-loss anti-tonicity lemma (A3, Lemma 2), and obtain the multiplicity structure from a deterministic **chain cover** of the induced comparability graph — exact minimum in the regime our studies use — with explicit share allocation. The order is not required for fixed-sequence validity; it is what makes the stop lossless within each chain (Lemma 3). Two formal conditions (main text §3.6: capacity and spanning) separate *valid ordering* from *informative family design*, and a measured instance exhibits a cost-based tie-break returning an empty certified set where the ACE order certifies.

C4. We **evaluate** ACE-LTT through **two exploratory preregistered studies on a development split, one refused preregistration design audit, and one prospective registered test on a fold none of them read** — demonstrating that certification validity and family informativeness can diverge, while deployment preference remains a separate, post-certification decision — all of it under an **auditable provenance scheme** (M8, M13.8) in which registrations *and* refusals are artefacts. The confirmatory element is confined to the prospective test (M13), whose two primary predictions did not both hold; the sealed test set is unopened throughout.

M12 What is not claimed

- Theorem 1 gives no lower bound on $|\widehat{\Pi}_{\mathrm{ACE}}|$. An empty certified set is a valid outcome. Study 1 contains an early-stopping chain (the crop chain stopped at its safest member, $p_{\mathrm{joint}} = 0.1138$); Study 2 contains none — its historical run returned all 16 candidates, worst $p_{\mathrm{joint}} = 0.00012$. On the generic instance of M9 both chains stop early.

- Studies 1 and 2 are **exploratory / method-development** runs on a development split; they are not confirmatory guarantees on fresh data. Study 3 (M13) is prospective on a calibration fold neither of them read, but it is one fold evaluated once, and what it tests is the capacity *diagnostic*, not Theorem 1. The sealed test set remains unopened for all three.
- `BOUNDARY_SPANNING` and `MARGIN_CAPABLE` are design-time statuses computed from select-fold point estimates. Neither predicts the certify fold, and Study 2 measures how large the gap can be. Study 3 (M13) tested the capacity statement prospectively and returned a split result: the reach bound held for all eight registered pairs, the sufficiency direction failed for two of six capable pairs, so capacity is reported as a bound that survived a fresh fold and **not** as a sufficient condition.
- The guarantee is silent about any stratum whose audit population is below $n_{\min}$; in Study 2 the restricted strata hold 4 and 13 units against $n_{\min} = 59$, and the refusal is on the record.
- A2-V's unit-level i.i.d. sampling model is an **assumption about the split, not a theorem**. Defining the unit as a global deduplication component keeps near-duplicate frames and repeated views, the dominant source of dependence in this data, inside one unit rather than split across units; it does not establish that two distinct units — say two sequences from the same camera on the same day — are independent. The honest statement is that the guarantee is conditional on this assumption, and that the unit definition was chosen to make it as defensible as the available grouping allows.
- A per-loss monotonicity lemma is a *mathematical obligation on the analyst*, not something the framework can discover. What the framework enforces is that the obligation was discharged and named before the run.
- **Study 2's family selection read certify-fold audit counts, and its outcome is not presented as a Theorem-1-backed certificate.** `select_stage` computes $r_k^* = r^*(n_k, \alpha_k, \delta_m)$ from the audit counts it is handed, and both `ace2_register` and `ace2_dev` hand it the counts computed on the **certify** mask. Its thresholds and its b ladder are therefore functions of test-fold audit metadata, which A1 does not permit. A *realized-family sensitivity audit* — replaying that selection on the archived package (`analysis_code_sha256 a7336412...`) with design-only reference counts $n_k^{\text{ref}} = \lfloor n_k^{\text{sel}}(1 - \text{sf})/\text{sf} \rfloor$, i.e. $(1162, 1078)$ against the historical $(1141, 1043)$ — reproduces the registered thresholds, the identical ladders, the identical family digest `85ecb498...` and the identical cover `80cc3c33...`. So the dependence did not alter what was tested in that realization. The margin is not comfortable: the shift in r^* is 2.1–$2.2 \cdot 10^{-4}$ while the nearest decision margin in the registered slack profile is $5.5 \cdot 10^{-4}$ (the overlay chain's smoke slack), a factor of 2.45. And equality in one realization is **not** independence of the procedure: validity is a property of the selection rule under repeated sampling, and a different certify fold could give different counts, a different r^* and a different family. The historical run is therefore retained as an exploratory method-development instance. Study 1 is unaffected — `select_thresholds` reads only α and select-fold data, and its family is built from literal thresholds and the declared λ grid. Study 3 was specified so that audit metadata is read only where a clause can refuse, never where it selects, and its executed run records that separation holding: the audit receipt carries no design quantity, and formal C3 had a refusal available on the realized count and did not need it (M13.4).
- **The provenance mechanism is not a security mechanism.** The runtime-tree manifest, the analysis-code digest, the registered declaration, the audit membership digests and the inference-registry digest detect changes to registered declarations, to package code and to the executable inference registration between registration and execution. They are not intended to defend against arbitrary mutation of the running interpreter or against a malicious execution environment: code executing in the same process can rebind a global

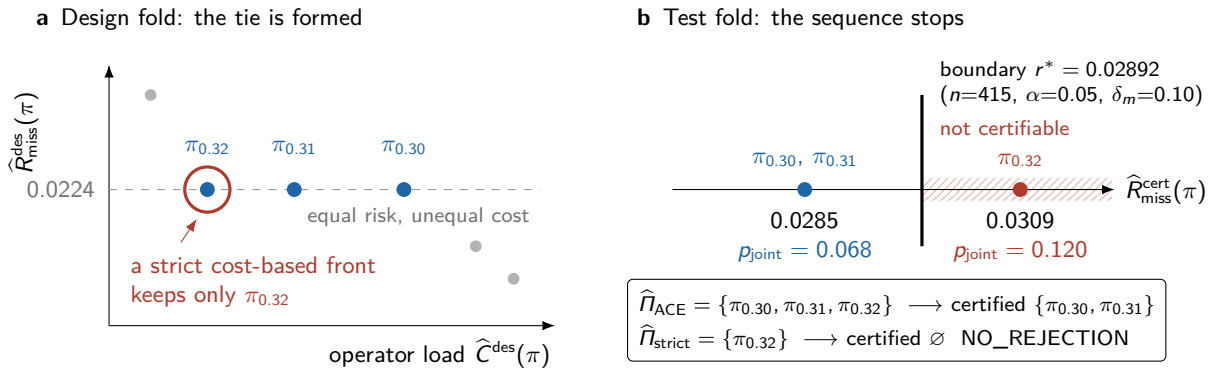

Figure 1: The ordering witness, as in the main text. Panel **a**: the select fold, a tie group at equal risk and unequal operator load, with the member a strict front would keep circled. Panel **b**: the certify fold with the certification boundary r^*, the measured certify risks and p-values, and the two outcomes. All values measured on the generic synthetic instance.

that a registered callable calls — or the verifier itself — and no digest computed by that process can exclude this. What is claimed is reproducibility and the detection of accidental or package-level drift, not tamper-resistance.

- **No new multiplicity principle, and no optimality of the allocation.** Theorem 1(i) is an instance of the sequential rejection principle [6]. The chain-cover Bonferroni split is the simplest auditable allocation on the certified order, not the most powerful family-wise procedure for it. In particular, a graphical procedure [25] could pass the share of a chain that is certified in full to the remaining chains and stay valid; ACE-LTT does not recycle shares, because a terminal chain's share is never redistributed (Theorem 1(v)) and because Corollary 1 is stated for the non-recycling allocation. A recycling variant is left open.
- **Fail-closed certification is not claimed as novel.** Seldonian and high-confidence policy improvement algorithms return "no solution found" when their safety test fails [20, 21], and a 2026 protocol for forecasting-leaderboard deployment decisions refuses to certify when its preconditions are undischarged [26]. What we claim is the specific contract set of M3 and the inference contract of M2.1, discharged by the same code that runs the test.
- The hand-off price used in Study 2's utility belongs to that study, not to ACE-LTT. A study whose chain is the operator tier should constrain hand-off rather than price it, or report the Pareto front over the certified set.

Figure 2 — source

`fig_witness_inline.tex` (TikZ; compiles with `pdflatex` against `fig1_preamble.tex`). Panel (a): the select fold, a tie group at equal risk and unequal operator load, with the member a strict front would keep circled. Panel (b): the certify fold with the certification boundary r^*, the measured certify risks and p-values, and the two outcomes — $\widehat{\Pi}_{\text{ACE}}$ certifies two policies, $\widehat{\Pi}_{\text{strict}}$ certifies none. All values measured on the generic synthetic instance.

M13 Study 3 — a prospective test of the capacity statement

M7.2's capacity condition is a design-time diagnostic, and Theorem 1(iv) excludes it from the validity argument. Studies 1 and 2 could not test it: both computed it on the same development

split their families were built from, so neither says whether the quantity carries any content for a fold the family was not designed on. Study 3 asks exactly that, prospectively. The family, the arm-wise cover, the intra-chain order, the δ shares, the influence envelopes, the reference bar, every capability bit, the exclusion sets and the verdict rules were fixed in a frozen preregistration (R_0) and then written to a registration receipt (R_1) **before any certify-fold quantity was opened**; the procedure then ran once.

This section reports Study 3 in full. The main text (§5) keeps one table row and four sentences, because Study 3 is evidence about the framework's discipline and not a premise of any claim in §3 — it can be deleted without touching Theorem 1 or either diagnostic.

M13.1 The access boundary, and what "prospective" means here

The calibration fold is split by the frozen protocol into **1814 select units** and **4231 certify units** (6045 calibration units in all; 12349 frames on the certify side). Studies 1 and 2 ran on the development split and read none of it. The sealed test fold is unrelated to Study 3 and remains sealed: no unseal or final receipt exists in any of the four protocol instances.

Three properties were enforced mechanically rather than by discipline:

- **Two-stage loading.** The design stage can reach calibration-**select** rows only. No certify label, score or feature is readable before R_1 exists on disk.
- **$\mathcal{A}$ opens once, after R_1.** The audit-metadata object carries realized certify counts and membership digests, and nothing else about the certify fold.
- **Refuse-only.** Design chooses; $\mathcal{A}$ may only refuse. Every quantity entering `CAPABLE`, T, S, N, `KNIFE_EDGE` and R2 is a function of the design fold, the declared constants and n^{ref}, all known at R_1. The realized n_k^{cert} can certify, feed X, or refuse — it cannot move a prediction or a denominator.

The one-shot property is also mechanical: the runner reserves its log atomically before the module starts, and `ace3_dev` refuses an existing run report. There is no resume path and no partial re-run.

M13.2 The design, as registered at R_1

quantity	value
n_k^{sel} (positive units, calibration-select)	fire 488, smoke 454
$n_k^{\mathrm{ref}} = \lfloor \frac{7}{3} n_k^{\mathrm{sel}} \rfloor$	fire 1138, smoke 1059
branch (Rule R1, from n^{ref} alone)	**full**: both $\geq 72 = n_{\min}(0.05, 0.025)$
M, δ_m, δ_{arm}, δ	4, 0.025, 0.05, 0.10
$r_k^{*,\mathrm{ref}} = r^*(n_k^{\mathrm{ref}}, \alpha, \delta_m)$	fire 0.034270650148391715, smoke 0.033994334191083914
reference bar bar_k	$\alpha_k = 0.05$
$h_k^{*,\mathrm{ref}} = \max(0, \mathrm{bar}_k - r_k^{*,\mathrm{ref}})$	fire 0.015729349851608287, smoke 0.01600566580891609
chain covers	arm B `cdae0eb78084eb21...` ($M = 2$), arm L `3f4a679978493de6...` ($M = 2$)
candidates	27 in arm B, 12 in arm L: **39 registered candidate instances, 37 distinct policy tuples**
R_2	**INFORMATIVE**, 6 capable / 2 incapable of the 8 eligible pairs — decided at R_1

Two points about this table are preregistration decisions rather than measurements, and both cut against the study's own convenience.

The reference bar is α, not $r^{*,\mathrm{ref}}$. Study 2 registered the tight selection bound $\widehat{R}_k^{\mathrm{sel}} \leq r_k^*$, which makes $h^* = 0$ and the capacity condition vacuously true. Study 3 registers the weaker but still selection-guaranteed bar $\mathrm{bar}_k = \alpha_k$ precisely in order to have a non-vacuous margin to fail against. The effect is not one-directional: a larger $h_k^{*,\mathrm{ref}}$ makes `CAPABLE` harder to reach and can move the R_2 verdict either way.

The cover is arm-wise, not minimal over the union. `build_chains` groups by `(acq, t_low, t_high, B0, strata)`, and that key holds the stratification *keys*, not the λ values, so at one mode and the same thresholds arms B and L share a group and the minimum cover interleaves them (measured: `['B|b=0','B|b=0.1','B|b=0.25','L|lam=0.4','B|b=0.4']`). Such a chain varies two coordinates, so neither S_B nor S_L is its envelope and the (C, k) primitives have nothing to attach to. The cover is therefore computed within an arm, δ is split $\delta_B = \delta_L = 0.05$ with one chain per (arm, mode) and $\delta_m = 0.025$ — arithmetically the same $4 \times 0.025 = \delta$ the full branch declares. Validity is untouched, since any $\succeq_{\text{cert}}$-chain partition fixed before the test fold gives the guarantee and minimality is a yield property (M4.3). The price paid is certification yield; what is bought is chains with a well-defined varying coordinate.

How Study 3 instantiates Theorem 1: twice, once per arm. The two arms are not one family covered into four chains; they are **two arm-specific instances of Theorem 1**, each with its own family, its own cover and its own budget — arm B with 27 candidates in 2 chains at $\delta_B = 0.05$, arm L with 12 candidates in 2 chains at $\delta_L = 0.05$. Within each arm the cover is a genuine partition, so Theorem 1 applies unchanged. Study-level validity is then one union bound over the two instances, $\Pr(E_B \cup E_L) \leq \Pr(E_B) + \Pr(E_L) \leq 0.05 + 0.05 = \delta$.

This matters because the two arms **overlap in one policy per acquisition mode**: arm B fixes $\lambda = 0.4$ and arm L fixes $b = 0.25$, so `B|·|b=0.25` and `L|·|lam=0.4` denote the same seven-coordinate operational policy under different names. The registered count of 39 is therefore 39 *candidate instances* across the two arms and **37 distinct policy tuples**; the two duplicated tuples are each tested twice, once in each arm, at that arm's own share. Nothing about validity turns on this — the union bound above never required the two families to be disjoint — but the count is stated both ways rather than letting $27 + 12$ imply a partition of 39 distinct policies.

The two arms, and their envelopes:

	arm B	arm L
varying coordinate	response authority b	veto authority λ
fixed other coordinate	$\lambda = 0.4$ (pre-Study-1 working point)	$b = 0.25$ (Study 2's selected operating point, as design information only)
envelope	S_B — **minimal**: a positive event whose only positive frames lie in $[t_{\text{low}}, t_{\text{high}})$	S_L — **conservative**: any positive event with a positive frame in `StratumSpec('small_smoke', 0.02)`
ladder	ascending g-grid edges $b \leq A$ at the frozen thresholds	$\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, safest first

S_L includes events the edge stage already catches, whose status λ cannot change, so $\widehat{\Gamma}_L$ is a valid upper bound that may be loose. The direction of that looseness was written down before the run: a larger $\widehat{\Gamma}_L$ makes **T easier** and `CAPABLE_L` easier to trigger, which makes **S harder**. No tightened envelope was substituted afterwards.

M13.3 The predictions, as registered

With $\text{sp}_k^D(C) = \widehat{R}_{k,D}(\pi^{(J)}) - \widehat{R}_{k,D}(\pi^{(1)})$ (Lemma 2):

> **T** (primary). $\text{sp}_k^{\text{cert}}(C) \leq \widehat{\Gamma}_k^{\text{sel}}(S)$. **S** (primary). $\text{CAPABLE}_k(C) = 1 \Rightarrow \text{sp}_k^{\text{cert}}(C) \geq h_k^{*,\text{ref}}$. **N** (derived, never evidence). $\text{CAPABLE}_k(C) = 0 \wedge \text{T}_{C,k} \Rightarrow \text{sp}_k^{\text{cert}}(C) < h_k^{*,\text{ref}}$. **X** (secondary, confounded). $\text{CAPABLE}_k(C) = 1$ and $\widehat{R}_k^{\text{sel}}(\pi^{(1)}) > r_k^{*,\text{ref}} \Rightarrow \exists \pi : \widehat{R}_{k,\text{cert}}(\pi) \leq r_k^{*,\text{cert}}$.

X is the only primitive touching $r^{*,\text{cert}}$, so it reads the actual certificate boundary rather than a design surrogate — and it is secondary because it mixes *where the chain lands* with *how far the coordinate moves it.* Comparisons are float64 at $\tau = 10^{-12}$; a pair with $|\widehat{\Gamma}_k^{\text{sel}} - h_k^{*,\text{ref}}| \leq 10^{-6}$ is `KNIFE_EDGE` and excluded from S and from R_2's count, both quantities being known at R_1 so the exclusion is blind to the outcome. A pair whose chain has $|C| = 1$ is `NO_REACH`: sp $\equiv 0$ makes T trivial and S trivially false, so it is excluded from T, S, N and R_2 and reported with

its numbers. The realized cover produced no `KNIFE_EDGE` pair and no $|C| = 1$ chain, so neither exclusion was exercised on a registered primary pair.

`CHAIN_TERMINAL` is the fail-closed counterpart: any post-registration terminal chain touching a registered primary pair forces `P-4.5 = REFUSED`, with no study-level T or S verdict, no recomputation of R_2 and no shrinking of the denominator. It differs from `NO_REACH` because it is observed during execution rather than decided at R_1.

M13.4 What $\mathcal{A}$ returned, and formal C3

```
n_cert                fire 1130   smoke 1114          both >= 72  ->  formal C3 PASS
membership digests    fire 02c363b2e9a9...   smoke 8a10f4194f73...   equal to the registered digests
r*_cert               fire 0.0345133   smoke 0.0341113
h*_C3 = max(0, r*_ref - r*_cert)        fire 0   smoke 0
carries_no_design_quantity              true
```

After $\mathcal{A}$ was opened, the realized audit populations were 1130 fire units and 1114 smoke units — one below and one above the corresponding design reference count, which carries no implication for the boundary, since $r^*(n, \alpha, \delta) = k^*(n)/n$ with k^* integer-valued and is not monotone in n over short ranges. Formal C3 checked those counts against the frozen full-branch requirement $n_{\min} = 72$ and passed both losses. The resulting certification boundaries were $r^{*,\mathrm{cert}}_{\mathrm{fire}} = 0.0345133$ and $r^{*,\mathrm{cert}}_{\mathrm{smoke}} = 0.0341113$, giving $h^*_{\mathrm{C3}} = \max(0,\ r^{*,\mathrm{ref}} - r^{*,\mathrm{cert}}) = 0$ for both. These audit-side quantities **verified** the registered procedure; none of them altered its branch, family, cover, order or shares. That is the refuse-only invariant doing its work — $\mathcal{A}$ arrived, was checked, and re-selected nothing.

M13.5 The per-pair result

J is the realized chain length, $\Gamma_{\mathrm{sel}} = \widehat{\Gamma}^{\mathrm{sel}}_k(S)$ the design-fold capacity of the chain's own arm at its own mode, $\mathrm{sp}_{\mathrm{cert}}$ the realized reach on the certify fold.

(C, k)	J	Γ_{sel}	$h^{*,\mathrm{ref}}$	CAPABLE	$\mathrm{sp}_{\mathrm{cert}}$	T	S	N	X
`B|crop|fire`	15	0.0554	0.0157	yes	0.0352	✓	✓	—	✓
`B|crop|smoke`	15	0.0684	0.0160	yes	0.0493	✓	✓	—	✓
`B|overlay|fire`	12	0.0554	0.0157	yes	0.0278	✓	✓	—	✓
`B|overlay|smoke`	12	0.0684	0.0160	yes	0.0412	✓	✓	—	✓
`L|crop|fire`	6	0.0021	0.0157	no	0.0000	✓	—	✓	✓
`L|crop|smoke`	6	0.0177	0.0160	yes	0.0001	✓	×	—	✓
`L|overlay|fire`	6	0.0021	0.0157	no	0.0000	✓	—	✓	✓
`L|overlay|smoke`	6	0.0177	0.0160	yes	0.0033	✓	×	—	✓

Study-level verdicts, as recorded.

```
T          8/8 held      all_held = True
S          4/6 held      all_held = False      (6 capable pairs eligible; both arm-L smoke pairs failed)
N          2/2           derived, reported, never a third supported hypothesis and never in a tally
X          8/8 held      secondary and confounded
P-4.5      INFORMATIVE   as registered at R1
chain_terminal_observed  false
R2_recomputed            false
denominator_shrunk       false
```

Two of the eight T inequalities had exactly zero realized endpoint span and were therefore non-discriminating; they remain in the registered denominator because the pre-specified `NO_REACH` exclusion applied only to singleton chains. Both are arm-L fire pairs, and both are `CAPABLE = no`, so they carry no weight in S either. T's non-trivial support is six pairs of eight, and S's four successes are all in arm B.

M13.6 Certificates and operating points

The same run is an ordinary registered ACE-LTT certification on fresh units, and that part of it succeeded:

```
arm B     delta 0.05   M = 2   20 of 27 members certified   no reason codes
arm L     delta 0.05   M = 2   12 of 12 members certified   no reason codes
```

The preregistered utility $(c_{\text{calls}}, c_{\text{FA}}, c_{\text{handoff}}) = (1.0, 10.0, 20.0)$ is applied **within each arm and never pooled across arms**, because the two arms carry separate δ shares and comparing their utilities would be comparing policies certified under different multiplicity budgets:

arm	operating point	utility	FA	cloud calls	hand-off
B	`B|crop|b=0.5`	2.4465 over 20 certified	0.0464	0.1422	0.0920
L	`L|overlay|lam=0`	0.7121 over 12 certified	0.0330	0.1422	0.0120

These are selection-side quantities measured on the certify fold, not unbiased performance estimates. The utility plays no part in T, S, N or X.

M13.7 Where the failures sit, and what that does and does not license

Both S failures are arm L's smoke pairs, and the shortfall is not marginal: realized reach 1×10^{-4} and 3.3×10^{-3} against a margin of 0.0160 — between one and two orders of magnitude. "S nearly held" is not an available reading.

Arm L is also the arm whose envelope the preregistration flagged as the conservative one, and the consequence was written down in advance: a loose $\widehat{\Gamma}_L$ raises T's right-hand side and makes `CAPABLE_L` easier to trigger, which makes S harder to satisfy. That locates the failure; it does not discount it. The registered reading of "T holds and S fails" was likewise fixed in advance — *capacity transfers as a bound but is not sufficient, and M7.2 stays a necessary-condition diagnostic* — so the conclusion drawn here is the one the preregistration assigned to this outcome before the outcome was visible. No tightened envelope, no recomputed R_2, no smaller denominator, and no post-hoc promotion of X or N to fill the gap.

M13.8 Provenance

object	digest	recorded
protocol instance `protocol_v249`	lock inner `7a30ea4a43fc7b3b...`, lock file `01707a2314ce78f5...`	frozen 2026-09-28T01:40:18Z
`analysis_code_sha256`	`aa461f6b8f1f2b556544e78680565ccc5cd 4fbad56dbd8fe85185dd1069464b7`	pinned in R_1
registration receipt R_1	`d9bcf680f1374d8124fe7549147538af372 2683feda6e72f8821dee8da0c6b42`	2026-09-28T02:09:58Z, `audit_metadata_present false`
audit receipt ($\mathcal{A}$)	`e52ade7dd6ebe5e2c07de3c700e77b1b5bb 940706e88e91a5d79cfa4463e1182`	2026-09-28T02:49:22Z, `carries_no_design_quantity true`
run report	`dad2494631c6ad5b95f75b3a3acec4ca11b f306fda9994ee243cdbe3e32e697d`	the one-shot execution
refusal artifact	absent	no `ace3_REFUSED.json` was written

All four protocol instances in this work carry byte-identical `protocol.yaml` (`f585e490...`) and `splits.csv` (`362d8689...`), so the fold definition Study 3 used is the fold definition frozen for the whole series.

The frozen preregistration (R_0) carries a **dated post-run addendum of 28 September 2026** recording this outcome against the declaration: the R_0 text itself is byte-unchanged, and the addendum states the verdicts, the two failed pairs and the four artifact digests without amending any registered rule, classification or denominator.

The registered diagnostic set is larger than the table above — per pair it also records $\widehat{\Gamma}_k^{\text{cert}}$, sp_k^{sel}, the anchor shift $\Delta_k = \widehat{R}_k^{\text{cert}}(\pi^{(1)}) - \widehat{R}_k^{\text{sel}}(\pi^{(1)})$ as a multiple of g_k^*, the select-fold standard error, the per-loss counts and the stop index — and it lives in the run report at the digest above, which is what ships. Those quantities are reported, never tested.

An independent recomputation. The verdicts were not taken from the report. A separate post-run audit re-derives $h^{*,\text{ref}}$, the R_1 status bits, the R_2 verdict, the chain endpoints,

each $\mathrm{sp}_k^{\mathrm{cert}}$, the T/S/N/X tallies, $p_{\mathrm{joint}} = \max_k p_k$ and the certified prefix of each chain from the registration and the report, and compares. It also checks that the terminal- chain status was reconstructed from the registered cover rather than read from the report's own field — the engine drops terminal chains from its chain list, so reading that field could not have detected the condition it is supposed to detect. The audit's first run failed on a defect of its own (it parsed the acquisition mode from the wrong field of the policy name and rebuilt zero endpoints); the defect and its correction are on the record in the change-set, and the corrected audit reproduces every number in this section.

M14 A paired descriptive comparison against a single-stage baseline

This belongs to **Study 2** and is descriptive throughout: no p-value, no decision rule, and the baseline was not a registered candidate. It is reported because it is the one place where the framework's output can be put next to the obvious alternative on the same units.

The comparison holds the first-stage threshold fixed and removes the second stage, on the 4192 shared certify units of Study 2's fold:

quantity	cascade `overlay|b=0.25`	single stage, same t_{high}	difference
fire event-miss risk	**0.0222** (109/2626 events)	0.0450 (168/2626)	−0.0228
smoke event-miss risk	**0.0180** (56/3330)	0.0416 (127/3330)	−0.0237
false alarm	0.0738 (287/3891)	0.0439 (171/3891)	+0.0298
cloud calls	0.0715 (789/11037)	0.0000	+0.0715
hand-off	0.0000	0.0000	0

Paired per-unit differences (cascade − single stage):

risk	mean	90% interval	units worse off	units better off
fire	−0.02282	[−0.03010, −0.01584]	**0**	31 of 1141
smoke	−0.02365	[−0.03178, −0.01631]	**0**	31 of 1043

The constrained miss risk roughly halves, and monotonically: not one unit is worse off. The gain concentrates in 31 units where about 0.8 of each unit's positive events flip from missed to caught — units the first stage misses and the verifier promotes out of the uncertain band. The price is +0.0298 false alarms and 7.15% second-stage calls. The intervals are a descriptive cluster bootstrap on a policy chosen after the certify fold was seen, so they are post-selection and not confidence statements.

The sharpest observation is about certifiability rather than risk. The single-stage baseline sits at fire 0.0450 and smoke 0.0416 — **under** $\alpha = 0.05$ and **above** the certification boundary $r^* = 0.0359/0.0355$ of that fold. Had it been a candidate, its HB p-values would have been 0.733 and 0.374 against $\delta_m = 0.05$; the certified cascade's are $6 \cdot 10^{-6}$ and $< 10^{-6}$. So on this fold, at these thresholds, the cascade is certifiable and the single-stage baseline is not. That is the distinction the whole apparatus exists to be able to draw — and it is stated descriptively, because the baseline was not in the registered family. A confirmatory version requires the single stage declared as a candidate in a future registration.

M15 Controlled validation: the preregistered Monte Carlo study

This section reports the study summarised in main-text §4.2. It is controlled validation of the method, not application evidence, and it proves nothing beyond M5.5: it checks that the frozen

engine implements Proposition 3, Corollary 1 and Theorem 1, and measures their consequences against known truth.

M15.1 Registration and provenance

The design was frozen as registration R0 before any code for the study existed, in the order frozen engine → frozen R0 → harness. R0 fixes the data-generating process, the grids, the generator values, the seeds (master seed 20261001, one child stream per replication), the definitions of a successful execution and of the two refusal paths, and every decision rule with its failure threshold. R0 does not contain its own digest; it is recorded in a detached freeze record, which also pins the engine and the generator values.

item	SHA-256
registration R0	`a8e0b6fca234f043d7ead1e4a6f7d00b605a455888c1101cf5042 74c6271cedd`
engine `ace_ltt.py` (v2.4.7, byte-identical through v2.4.9)	`ca93d2f3e4eafa85255f512318c0e7a350d259640518696ddbfed 23e93960241`
generator values	`0f0ed400138b15a28b3408b323e7964f420d698dddb1fc3d5b5e5 284cc902ab8`
harness (written after R0)	`3c091eb22370232a50d85353dc06989685c6d9370edc719889372 c530f0399df`
run manifest	`2cfd7bec570b6ff60e550e6752367771b236311a9c6e81f93d91c a98de185c0b`

A timing pilot on off-grid families, run after R0 and before the registered run, projected 330 s for the validity block against a registered limit of 72 h, so the registered 5000 replications per cell were used. The registered run was executed once (24 worker processes, Python 3.12.4, NumPy 1.26.4, SciPy 1.13.1).

M15.2 Design

Each replication draws n i.i.d. units, a stratum per unit ($\Pr(S1) = 0.3$), and for each of L risk types a Beta-distributed verifier score; the event of a unit is caught iff $\min(1, g + v(\sigma)) \geq b$, which is the engine's own `response` rule for this configuration and is checked unit by unit (A0). True risks are closed-form, $R_k(\pi) = \sum_\sigma \Pr(\sigma) F_{k\sigma}(b_\pi - v_\pi(\sigma))$. Grid levels for the three truth configurations were computed from this formula only, by a registered constructive procedure. Two refusal paths are distinguished: **R2**, every candidate refused at step 2 at the provisional share, before any cover is registered; and **T4**, step 2 passes but the re-check at the executed share makes every chain terminal. Non-minimum covers are executed by a transcription of the engine's procedure built from its own functions (Path L), whose agreement with the engine's `certify_family` on the minimum cover is itself a registered check (A6).

block	role	cells	scale
MC-A	conformance with Proposition 3 and Remark (a)	180 (90 with the lemma present, 90 removed)	200 replications
MC-B	yield of the minimum cover (effect size, no pass/fail)	runs of MC-A	—
MC-C	family-wise error against known truth	90	5000 replications
MC-D	greedy cover under deletion	3 registered posets + one 512-candidate poset	1000 deletion sets

M15.3 Conformance (MC-A) and generator check (G1)

check	requirement	observed
A0 engine decision rule, per unit, policy and loss	0 mismatches	0 / 586,656,000

check	requirement	observed
A1 inclusion (2), successful cover pairs	0 violations	0 / 2,306,485
A2 equality (1) with strict `>`, successful executions	0 mismatches	0 / 296,934
A3 per-loss p-values non-decreasing along admissible edges	0 violations	0
A4 Remark (a): smaller M refused by C3 while larger M succeeds	0	0
A5 refusal path (pass / R2 / T4) as predicted from $n_{\min}$	0 disagreements	0
A6 Path L identical to `certify_family` on $C_{\min}$	0 mismatches	0 / 36,000
G1 pooled empirical risk vs closed form, $\lvert z\rvert > 5$	no trigger	0

As an additional implementation-conformance diagnostic of equality (1), not a further proof, all covers of equal M returned identical sets within every replication.

M15.4 Validity (MC-C)

Headline: 0 of 90 cells met the preregistered failure rule (one-sided 99% Clopper–Pearson lower bound of the cell FWER above $\delta = 0.10$). The largest observed FWER, 0.070 $[0.063, 0.077]$, occurred in the least-favourable setting with $M = 1$, $n = 50$ and risks concentrated near α; Theorem 1 bounds the FWER but does not predict which cell attains the maximum. Descriptively, the rate pooled over the 72 non-vacuous cells is 0.0064 $[0.0061, 0.0067]$; it averages different registered cells and is not an estimate of a single population FWER. The 18 vacuous cells are R2 by construction and certify nothing. The singleton (Bonferroni) baseline has FWER at most 0.0034 in every cell. Configurations: `spr` spread, `lf` least-favourable, `gn` global null; intervals are two-sided 95% Clopper–Pearson.

config	d	L	n	FWER	95% CI	99% lower	singleton FWER	refusal
`spr`	1	1	50	0.0476	[0.0419, 0.0539]	0.0409	0.0000	—
`spr`	1	1	80	0.0096	[0.0071, 0.0127]	0.0067	0.0000	—
`spr`	1	1	150	0.0102	[0.0076, 0.0134]	0.0072	0.0022	—
`spr`	1	1	400	0.0076	[0.0054, 0.0104]	0.0050	0.0002	—
`spr`	1	1	1000	0.0030	[0.0017, 0.0049]	0.0015	0.0004	—
`spr`	1	2	50	0.0042	[0.0026, 0.0064]	0.0024	0.0000	—
`spr`	1	2	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	1	2	150	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	1	2	400	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	1	2	1000	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	2	1	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`spr`	2	1	80	0.0196	[0.0159, 0.0238]	0.0153	0.0000	—
`spr`	2	1	150	0.0054	[0.0036, 0.0078]	0.0033	0.0006	—
`spr`	2	1	400	0.0028	[0.0015, 0.0047]	0.0014	0.0004	—
`spr`	2	1	1000	0.0026	[0.0014, 0.0044]	0.0012	0.0006	—
`spr`	2	2	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`spr`	2	2	80	0.0004	[0.0000, 0.0014]	0.0000	0.0000	—

config	d	L	n	FWER	95% CI	99% lower	singleton FWER	refusal
`spr`	2	2	150	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	2	2	400	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	2	2	1000	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	3	1	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`spr`	3	1	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`spr`	3	1	150	0.0006	[0.0001, 0.0018]	0.0001	0.0006	—
`spr`	3	1	400	0.0028	[0.0015, 0.0047]	0.0014	0.0006	—
`spr`	3	1	1000	0.0044	[0.0028, 0.0067]	0.0025	0.0012	—
`spr`	3	2	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`spr`	3	2	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`spr`	3	2	150	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	3	2	400	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`spr`	3	2	1000	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`lf`	1	1	50	0.0700	[0.0631, 0.0774]	0.0618	0.0000	—
`lf`	1	1	80	0.0162	[0.0129, 0.0201]	0.0123	0.0000	—
`lf`	1	1	150	0.0132	[0.0102, 0.0168]	0.0097	0.0026	—
`lf`	1	1	400	0.0274	[0.0231, 0.0323]	0.0223	0.0028	—
`lf`	1	1	1000	0.0180	[0.0145, 0.0221]	0.0139	0.0016	—
`lf`	1	2	50	0.0050	[0.0032, 0.0074]	0.0030	0.0000	—
`lf`	1	2	80	0.0004	[0.0000, 0.0014]	0.0000	0.0000	—
`lf`	1	2	150	0.0006	[0.0001, 0.0018]	0.0001	0.0000	—
`lf`	1	2	400	0.0004	[0.0000, 0.0014]	0.0000	0.0000	—
`lf`	1	2	1000	0.0008	[0.0002, 0.0020]	0.0002	0.0000	—
`lf`	2	1	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`lf`	2	1	80	0.0156	[0.0124, 0.0194]	0.0118	0.0000	—
`lf`	2	1	150	0.0036	[0.0021, 0.0057]	0.0019	0.0006	—
`lf`	2	1	400	0.0038	[0.0023, 0.0059]	0.0021	0.0006	—
`lf`	2	1	1000	0.0038	[0.0023, 0.0059]	0.0021	0.0010	—
`lf`	2	2	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`lf`	2	2	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`lf`	2	2	150	0.0002	[0.0000, 0.0011]	0.0000	0.0000	—
`lf`	2	2	400	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`lf`	2	2	1000	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`lf`	3	1	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`lf`	3	1	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)

config	d	L	n	FWER	95% CI	99% lower	singleton FWER	refusal
`lf`	3	1	150	0.0006	[0.0001, 0.0018]	0.0001	0.0006	—
`lf`	3	1	400	0.0008	[0.0002, 0.0020]	0.0002	0.0000	—
`lf`	3	1	1000	0.0026	[0.0014, 0.0044]	0.0012	0.0002	—
`lf`	3	2	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`lf`	3	2	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`lf`	3	2	150	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`lf`	3	2	400	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`lf`	3	2	1000	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`gn`	1	1	50	0.0666	[0.0598, 0.0739]	0.0586	0.0000	—
`gn`	1	1	80	0.0130	[0.0100, 0.0165]	0.0096	0.0000	—
`gn`	1	1	150	0.0166	[0.0132, 0.0205]	0.0127	0.0026	—
`gn`	1	1	400	0.0190	[0.0154, 0.0232]	0.0148	0.0034	—
`gn`	1	1	1000	0.0132	[0.0102, 0.0168]	0.0097	0.0012	—
`gn`	1	2	50	0.0040	[0.0024, 0.0062]	0.0022	0.0000	—
`gn`	1	2	80	0.0002	[0.0000, 0.0011]	0.0000	0.0000	—
`gn`	1	2	150	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`gn`	1	2	400	0.0004	[0.0000, 0.0014]	0.0000	0.0000	—
`gn`	1	2	1000	0.0002	[0.0000, 0.0011]	0.0000	0.0000	—
`gn`	2	1	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`gn`	2	1	80	0.0154	[0.0122, 0.0192]	0.0116	0.0000	—
`gn`	2	1	150	0.0022	[0.0011, 0.0039]	0.0010	0.0004	—
`gn`	2	1	400	0.0024	[0.0012, 0.0042]	0.0011	0.0002	—
`gn`	2	1	1000	0.0018	[0.0008, 0.0034]	0.0007	0.0000	—
`gn`	2	2	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`gn`	2	2	80	0.0004	[0.0000, 0.0014]	0.0000	0.0000	—
`gn`	2	2	150	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`gn`	2	2	400	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`gn`	2	2	1000	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
`gn`	3	1	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`gn`	3	1	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`gn`	3	1	150	0.0004	[0.0000, 0.0014]	0.0000	0.0004	—
`gn`	3	1	400	0.0012	[0.0004, 0.0026]	0.0004	0.0000	—
`gn`	3	1	1000	0.0004	[0.0000, 0.0014]	0.0000	0.0000	—
`gn`	3	2	50	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)
`gn`	3	2	80	0.0000	[0.0000, 0.0007]	0.0000	0.0000	R2 (vacuous)

config	d	L	n	FWER	95% CI	99% lower	singleton FWER	refusal
gn	3	2	150	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
gn	3	2	400	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—
gn	3	2	1000	0.0000	[0.0000, 0.0007]	0.0000	0.0000	—

M15.5 Yield (MC-B)

Yield is the share of non-null policies returned; a refused execution returns the empty set and has yield 0. $\Delta(C) = |\widehat{\Pi}(C_{\min})| - |\widehat{\Pi}(C)|$ is computed only over pairs in which both executions succeed (— where the singleton execution is refused). Yield is not defined for the global null, which has no non-null policy (—); Δ is still reported there. No pair occurred in which C succeeded while $C_{\min}$ was refused. Lemma present, 200 replications per cell; Δ is shown for the singleton cover.

config	d	L	n	yield $C_{\min}$	yield singletons	mean Δ (singletons)	$\Pr(\Delta > 0)$
spr	1	1	50	0.325	0.000	—	—
spr	1	1	80	0.218	0.000	—	—
spr	1	1	150	0.362	0.223	0.41	0.37
spr	1	1	400	0.613	0.457	0.47	0.47
spr	1	1	1000	0.748	0.672	0.23	0.23
spr	1	2	50	0.212	0.000	—	—
spr	1	2	80	0.095	0.000	—	—
spr	1	2	150	0.310	0.148	0.48	0.47
spr	1	2	400	0.597	0.423	0.52	0.52
spr	1	2	1000	0.713	0.637	0.23	0.23
spr	2	1	50	0.000	0.000	—	—
spr	2	1	80	0.136	0.000	—	—
spr	2	1	150	0.105	0.027	0.72	0.28
spr	2	1	400	0.369	0.226	1.28	0.77
spr	2	1	1000	0.655	0.570	0.77	0.64
spr	2	2	50	0.000	0.000	—	—
spr	2	2	80	0.030	0.000	—	—
spr	2	2	150	0.019	0.004	0.13	0.06
spr	2	2	400	0.209	0.086	1.11	0.56
spr	2	2	1000	0.523	0.456	0.60	0.48
spr	3	1	50	0.000	0.000	—	—
spr	3	1	80	0.000	0.000	—	—
spr	3	1	150	0.041	0.041	0.00	0.00
spr	3	1	400	0.300	0.173	3.28	0.95
spr	3	1	1000	0.548	0.461	2.29	0.94
spr	3	2	50	0.000	0.000	—	—
spr	3	2	80	0.000	0.000	—	—
spr	3	2	150	0.006	0.006	0.00	0.00
spr	3	2	400	0.204	0.106	2.73	0.94
spr	3	2	1000	0.450	0.371	2.22	0.94
lf	1	1	50	0.355	0.000	—	—
lf	1	1	80	0.200	0.000	—	—
lf	1	1	150	0.515	0.300	0.27	0.23
lf	1	1	400	0.920	0.725	0.24	0.21
lf	1	1	1000	1.000	0.995	0.04	0.03
lf	1	2	50	0.160	0.000	—	—
lf	1	2	80	0.080	0.000	—	—
lf	1	2	150	0.240	0.090	0.15	0.15
lf	1	2	400	0.930	0.615	0.32	0.32
lf	1	2	1000	1.000	0.995	0.01	0.01
lf	2	1	50	0.000	0.000	—	—
lf	2	1	80	0.225	0.000	—	—
lf	2	1	150	0.253	0.054	0.99	0.23
lf	2	1	400	0.720	0.526	0.99	0.27
lf	2	1	1000	1.000	0.997	0.04	0.01
lf	2	2	50	0.000	0.000	—	—
lf	2	2	80	0.087	0.000	—	—
lf	2	2	150	0.081	0.001	0.40	0.10
lf	2	2	400	0.714	0.450	1.32	0.40
lf	2	2	1000	0.999	0.994	0.03	0.01

config	d	L	n	yield $C_{\min}$	yield singletons	mean Δ (singletons)	$\Pr(\Delta > 0)$
`lf`	3	1	50	0.000	0.000	—	—
`lf`	3	1	80	0.000	0.000	—	—
`lf`	3	1	150	0.081	0.081	0.00	0.00
`lf`	3	1	400	0.670	0.408	4.18	0.46
`lf`	3	1	1000	1.000	0.996	0.17	0.03
`lf`	3	2	50	0.000	0.000	—	—
`lf`	3	2	80	0.000	0.000	—	—
`lf`	3	2	150	0.008	0.008	0.00	0.00
`lf`	3	2	400	0.701	0.359	5.47	0.59
`lf`	3	2	1000	0.998	0.985	0.20	0.04
`gn`	1	1	50	—	—	—	—
`gn`	1	1	80	—	—	—	—
`gn`	1	1	150	—	—	0.00	0.00
`gn`	1	1	400	—	—	0.01	0.01
`gn`	1	1	1000	—	—	0.01	0.01
`gn`	1	2	50	—	—	—	—
`gn`	1	2	80	—	—	—	—
`gn`	1	2	150	—	—	0.00	0.00
`gn`	1	2	400	—	—	0.00	0.00
`gn`	1	2	1000	—	—	0.00	0.00
`gn`	2	1	50	—	—	—	—
`gn`	2	1	80	—	—	—	—
`gn`	2	1	150	—	—	0.01	0.01
`gn`	2	1	400	—	—	0.01	0.01
`gn`	2	1	1000	—	—	0.00	0.00
`gn`	2	2	50	—	—	—	—
`gn`	2	2	80	—	—	—	—
`gn`	2	2	150	—	—	0.00	0.00
`gn`	2	2	400	—	—	0.00	0.00
`gn`	2	2	1000	—	—	0.00	0.00
`gn`	3	1	50	—	—	—	—
`gn`	3	1	80	—	—	—	—
`gn`	3	1	150	—	—	0.00	0.00
`gn`	3	1	400	—	—	0.00	0.00
`gn`	3	1	1000	—	—	0.00	0.00
`gn`	3	2	50	—	—	—	—
`gn`	3	2	80	—	—	—	—
`gn`	3	2	150	—	—	0.00	0.00
`gn`	3	2	400	—	—	0.00	0.00
`gn`	3	2	1000	—	—	0.00	0.00

In the registered figure slice (main-text Figure 3) each point is the mean over replications of the per-replication mean over covers of that M, so that replications carry equal weight. M values that occur in all 200 replications are joined by lines; the values below occur only for some random admissible covers and are shown as open, unjoined markers.

d	n	$M/M_{\min}$	replications containing M	mean yield
2	50	1.600	2	0.000
2	150	1.600	1	0.444
2	400	1.200	198	0.370
2	400	1.600	2	0.278
3	50	1.083	111	0.000
3	50	1.167	196	0.000
3	50	1.333	171	0.000
3	50	1.500	5	0.000
3	80	1.083	119	0.000
3	80	1.167	191	0.000
3	80	1.333	154	0.000
3	80	1.500	3	0.000
3	150	1.083	101	0.040
3	150	1.167	196	0.041
3	150	1.250	199	0.041
3	150	1.333	166	0.040
3	150	1.500	4	0.000
3	400	1.083	110	0.302
3	400	1.167	199	0.301
3	400	1.250	198	0.298
3	400	1.333	162	0.305

d	n	$M/M_{\min}$	replications containing M	mean yield
3	400	1.500	2	0.019
3	1000	1.083	123	0.556
3	1000	1.167	199	0.549
3	1000	1.333	167	0.547
3	1000	1.500	4	0.375

M15.6 Greedy cover under deletion (MC-D), and the registered witness

arm	poset	$M_{\text{greedy}} - M_{\text{exact}}$ on Π	event E	interval
D1	$d = 1$ (8)	0	0 / 1000	$U_{0.95} = 0.0030$
D1	$d = 2$ (25)	0	0 / 1000	$U_{0.95} = 0.0030$
D1	$d = 3$ (64)	0	0 / 1000	$U_{0.95} = 0.0030$
D2	$8 \times 8 \times 8$ (512, production greedy)	0	1 / 1000	[0.000025, 0.0056]

All registered consistency checks (the exact count never rises under deletion, the greedy count is never below the exact one, every chain is certified) held in every deletion set of every arm.

E is the event that deleting candidates raises the greedy chain count. Zero-event intervals are the preregistered one-sided 95% upper bound $1 - 0.05^{1/1000}$; the non-zero interval is two-sided 95% Clopper–Pearson. The D1 arm calls the greedy routine directly on posets below the engine's greedy threshold; D2 runs the engine's production path, which selects the greedy routine above 400 candidates.

Witness (archived under the registered witness rule; seed `SeedSequence(20261001, spawn_key=(3, 3210))`). On the 512-candidate family the production greedy cover and the exact cover both have 48 chains. Deleting the three candidates $(b, v(S0), v(S1)) = (0.10, 0.02, 0.08)$, $(0.10, 0.00, 0.08)$ and $(0.30, 0.06, 0.02)$ raises the greedy cover to 49 chains while the exact cover stays at 48; under uniform allocation δ_m falls from $0.10/48$ to $0.10/49$. The instance was reproduced from the seed and the frozen engine: the same deletion set, the same two greedy covers, and every chain certified. It shows both halves of Remark (c) of M5.5: the greedy count is not monotone under deletion, and on the reduced family the greedy cover is not minimum.

M15.7 Disclosures

1. Pilot and run were started with one BLAS thread per process (`OMP_NUM_THREADS`, `OPENBLAS_NUM_THREADS`, `MKL_NUM_THREADS` = 1); these settings affect timing only and are not in the manifest.
2. The registered phrase "mean over replications and covers at each observed M" is read as the nested mean described in M15.5; refused executions count as yield 0. Both readings were fixed before inclusion in the paper and change weighting only.
3. The full-family greedy cover of D2 is stored once in the run output rather than in the witness file; it is deterministic and was recomputed and matched.
4. Implementation choices that change no value: Path L memoises the engine's `joint_pvalue` output within a replication; the exact cover in MC-D reads a precomputed table of the engine's chain relation; the D1 posets use the `spread` grid levels, since the poset is the same in every configuration.

Appendix A — the generic worked instance in full

To separate method from application, the framework is instantiated on a dataset-free synthetic stream: 900 units, 5392 frames, two acquisition modes, a declared seeded generator, an 8-candidate family ($b \in \{0, 0.15, 0.30, 0.45\}$ per mode at $A = 0.55$), one miss loss (`unit_mean_-`

over_events, catch set response, $\alpha = 0.05$) and one payload loss ($\beta = 0.05$). This is an **illustration, not evidence**: the generator constants were chosen so that the instance exhibits the behaviours of interest, and no number in it is a measurement of any real system.

Design, select fold. $m = 8$ candidates cover into $M = 2$ chains, $\delta_m = 0.05$, $n_{\min} = 59$; audit population 415 miss units and 630 budget units with a single membership digest shared by all candidates. Both chains are BOUNDARY_SPANNING (max joint slack +0.00335, min −0.00152 and −0.02491) and the pre-freeze checklist closes **9/9**, the ninth item being the inference contract: every declared construction resolves in the registry, its weighting is one the construction is proved for, the registered registry digest matches the executable one, and no multi-member chain carries a construction without (O).

Certification, certify fold. $r^* = 0.02651$, $g^* = 0.02349$. Both chains truncate:

chain	members certified	stopped at	p_{joint} at the stop
a_1	$b = 0$, 0.15, 0.30	$b = 0.45$	0.062
a_2	$b = 0$, 0.15	$b = 0.30$	0.0675

5 of 8 certified, no reason codes; the preregistered utility then selects $a_1|b{=}0.30$. Note that the sequence never evaluates $a_2|b{=}0.45$ at all — Theorem 1(ii) in action.

Ordering. The witness of M6, rendered in Figure 2.

Refusals. Four fail-closed paths, each returning a status and no number: declaring 12 audit units against $n_{\min} = 59$ gives C3_CERTIFIABILITY_AUDIT + NO_REJECTION and 0 certified; removing the payload semantics gives C7_FROZEN_PAYLOAD_SEMANTICS and 0 certified; presenting a stale inference registration — a registered inference_registry_id that no longer matches the executable table — gives I_INFERENCE_REGISTRATION + NO_REJECTION and 0 certified, before any p-value is computed; and removing the monotonicity lemma from the miss loss turns the cover from $[4, 4]$ into eight singletons, δ_m from 0.05 to 0.0125, and leaves no fail-safe path. All four appear in the machine-written report as well as here.

Diagnostics are declarations. The same family, same data, three declarations:

select bar	influence envelope S	Γ	h^*	status
r^*	full	0.09357	0.00000	MARGIN_CAPABLE (vacuously)
α	full	0.09357	0.03495	MARGIN_CAPABLE
α	restricted to a fifth	0.01871	0.03495	MARGIN_-INCAPABLE

and the two safest members alone are NOT_SPANNING (max = min = +0.00335: nothing for the sequence to reject). The status is a property of **the declared family evaluated on design/select data**; it is not a property of the certify-fold outcome. Both halves of that sentence matter: the capacity is computed from labelled select-fold data (M7.2), so it is not label-free, and it is precisely because it depends on data the family was built from that it may not enter the validity argument.

Reproduction, and which package reproduces what. The instance reads no dataset, no protocol and no detector output, so it reproduces from the package alone: python -m cascade.ace_toy (or STAGE=ace-toy). Every number in M6 and M9 — the chain cover id c5eae7965d30c72977afbb74eaf1c21c433d224ed4b5b8ee8fc778b8492e3465, the audit population 415/630 with membership digest 388adb8c929aa5c8..., r^* and g^*, the 5-of-8 certified set, the witness table of M6 and the capacity/spanning statuses above — was previously reproduced through cascade_v2_4_4.zip, including byte-for-byte on a second, independent machine (Ubuntu 20.04 server). **Re-measurement on the shipped cascade_v2_4_9b.zip on 28 September 2026 reproduced all enumerated M6/M9 numerical and structural outputs, with no reported value changed.** cascade.ace_toy sits outside ANALYSIS_CODE, so the re-measurement changed no digest.

The provenance mechanism, however, does **not** let the current package stand in for an

archived one, and this is worth stating precisely because auditable provenance is part of what the method claims (main text, §1).

object	pinned to	verifies under the current package?
the generic instance of M6/M9	no dataset, no protocol; `cascade.ace_toy` alone	**yes — re-measured at `cascade_v2_4_9b.zip`, 2026-09-28**, and its module is deliberately outside `ANALYSIS_CODE`
Study 1's certificate	`analysis_code_sha256 db3c5a76...` (`cascade_v2_19.zip`, spec v2.2)	**no**, by design
Study 2's certificate	`analysis_code_sha256 a7336412...` (`cascade_v2_3_2.zip`, spec v2.3)	**no**, by design
Study 3's certificate	`analysis_code_sha256 aa461f6b...` (`cascade_v2_4_9b.zip`)	**yes** — its registration pins the current release
the current method implementation	spec v2.4.3, `analysis_code_sha256 aa461f6b...` (`cascade_v2_4_9b.zip`)	—

Studies 1 and 2 are re-executable **only** from the packages their registrations pin, which we ship. Executing either against the current tree makes `check_registration` refuse on `analysis_code_sha256`, and that refusal is the mechanism working rather than a gap in it: the code that produced a certificate is part of what the certificate is about, so a historical certificate cannot be transplanted onto a newer package. Study 3 is the one study whose registration pins the release we ship as current, so its registration verifies against that package identity; the one-shot gate still forbids re-execution. The implementation releases between the three studies corrected enforcement defects and tightened contract checks; they changed no registered rule and no reported value.

References

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

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

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

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

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

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

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

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

[9] 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).

[10] 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.

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

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

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

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

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

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

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

[18] Madras, D., Pitassi, T. & Zemel, R. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, vol. 31 (2018).

[19] Mozannar, H. & Sontag, D. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, 7076–7087 (2020).

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

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

[22] Prinster, D. *et al.* Conformal policy control. In *Proceedings of the 43rd International Conference on Machine Learning* (2026). Preprint arXiv:2603.02196.

[23] Klivans, A., Stavropoulos, K. & Vasilyan, A. Testable learning with distribution shift. In *Proceedings of the Thirty Seventh Conference on Learning Theory*, vol. 247 of *Proceedings of Machine Learning Research*, 2887–2943 (2024).

[24] Goel, S., Klivans, A., Stavropoulos, K. & Vasilyan, A. Testing noise assumptions of learning algorithms. In *Proceedings of the Thirty Ninth Conference on Learning Theory*, vol. 336 of *Proceedings of Machine Learning Research*, 2804–2853 (2026).

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

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