Mechanistic Generalization Certificates for OOD Retrieval via Causal Signature Invariants

Liz Lemma Future Detective

January 18, 2026

Abstract

Behavioral metrics on synthetic retrieval tasks (e.g., Associative Recall) reveal that some architectures succeed while others fail, but they do not explain *why* nor predict whether success will generalize. Building on mechanistic evaluation with interchange interventions, we define a small set of causal-intervention-derived invariants—mechanistic signatures—that characterize where and when key–value information is computed and written in a model. We formalize a certification problem: given an early checkpoint, can we predict out-of-distribution (OOD) generalization on retrieval/binding tasks under compositional splits? For a controlled family of tasks (AR, ATR, and ATR++ with ambiguity and multi-hop queries) and a broad class of residual sequence models, we provide (1) a signature extraction algorithm with sample complexity guarantees, (2) certificate conditions that imply OOD accuracy lower bounds under a linearized-mechanism assumption, and (3) a lower bound showing that in-distribution accuracy alone cannot certify OOD performance. Large-scale sweeps across Transformers, SSMs, and hybrid mixers demonstrate that early mechanistic signatures predict final OOD performance better than dev accuracy, enabling early stopping and principled architectural ablations (e.g., kernel size/local mixing). We release an open-source mechanistic evaluation harness for standardized certification of retrieval mechanisms.

Table of Contents

4. 4. Mechanistic Signatures: define intervention sites, corruption distributions, and invariant features (token-locus, layer-locus, stability, sparsity) and the certificate predicate.

5. 5. Algorithms: signature estimation from checkpoints; predictor training; early stopping and architecture selection procedures; implementation details and reproducibility checklist.

6. 6. Theory I (Separability): under linearization and algorithm-class assumptions, signatures separate bind-then-read vs direct-retrieval with explicit margin; robustness to noise in interventions.

7. 7. Theory II (Certification Guarantees): if the certificate predicate holds then OOD accuracy is lower-bounded on specified splits; sample complexity to estimate certificate with high probability.

8. 8. Theory III (Behavior-only Lower Bound): construct task/model families with identical dev accuracy but divergent OOD accuracy; conclude behavioral metrics cannot certify OOD generalization.

9. 9. Experiments: sweeps across architectures (attention, Based, Mamba-like, Hyena-like, hybrids), hyperparameters, and checkpoints; compare predictors (signature vs dev metrics); early stopping gains; ablations (kernel size, depth, positional encoding).

10. 10. Discussion and Limitations: when linearization assumptions break; scaling to fused kernels and real tasks; relation to mechanistic interpretability and architecture design.

11. 11. Release: mechanistic eval harness API, benchmark configs, recommended signature set, and reporting standards.

1 Introduction

In the setting of distribution shift induced by compositional recombination, we cannot treat in-distribution behavioral performance as a reliable proxy for out-of-distribution (OOD) competence. Concretely, even when a model attains high development accuracy (or equivalently assigns high likelihood to the correct next token on all observed development instances), this does not force the model to have learned the underlying rule that defines the task family; it only forces agreement on the finite set of compositions present in the development split. When the OOD split withholds particular query–answer compositions while preserving the same underlying binding rule, two models may be indistinguishable by any development metric and yet differ maximally in OOD accuracy. The difficulty is not statistical noise but identifiability: from development behavior alone, the implemented computation is underdetermined.

The synthetic retrieval/binding families we study (AR, ATR, and ATR++) make this underdetermination explicit. These tasks are intentionally constructed so that the intended solution is a simple, local algorithm (compute an association between a key and a value, then read out the value when queried), but there exists a competing strategy that fits the development distribution by exploiting superficial regularities or memorizing observed compositions. Both strategies can be realized by standard residual sequence models and both can achieve near-perfect development accuracy. Yet only the former strategy generalizes to compositional OOD splits. We therefore require a diagnostic that distinguishes *how* the model produces its answers, not merely *whether* it answers development queries correctly.

We refer to this phenomenon as *mechanistic divergence*: the same input–output behavior on development data can arise from qualitatively different internal computations. In AR/ATR-type tasks the relevant divergence is between (i) a *bind-then-read* mechanism, in which the model constructs an internal variable representing the key–value association at an intermediate token/site and subsequently routes this variable to the query position, and (ii) a *direct-retrieval* mechanism, in which the association is effectively written only at the query position or in late layers, bypassing an intermediate binding representation. The two mechanisms can coincide on the development distribution because the distribution may not expose the combinatorial degrees of freedom that separate them. However, once we present held-out compositions, a direct-retrieval mechanism can fail catastrophically while bind-then-read remains correct by construction. This is precisely the regime in which development accuracy is maximally misleading as an OOD predictor.

Our goal is therefore to construct an *early*, *architecture-agnostic* certification method that detects the presence of the intended mechanism before full training completes. "Early" means at checkpoints m_t with $t \ll T$, where T is the final training step; this matters operationally because we wish to select hyperparameters, architectures, or training curricula without paying the full cost of training each candidate to completion. “Architecture-agnostic” means that the method should not depend on attention-specific artifacts (e.g., head-wise interpretability), but instead should apply uniformly to residual sequence models with identifiable module boundaries, including attention, state-space, convolutional mixers, and hybrids. In particular, we do not assume that the mechanism is localized to a single head or that it can be read off from weights; we assume only that we can intervene at a fixed set of sites (e.g., block inputs/outputs, mixer outputs, MLP outputs, memory ports).

The central idea is to replace behavioral evaluation with a causal test that probes whether a specific internal site carries the information required for correct prediction. We will use *interchange interventions*: given a clean instance o and a corrupted counterpart c obtained by corrupting exactly one crucial token, we selectively replace internal representations of c with those of o at a chosen site/token, and we measure the restoration of the correct next-token likelihood. This yields a normalized likelihood-restoration attribution score that is comparable across checkpoints and architectures. By aggregating such scores over a small number of examples, we obtain a low-dimensional mechanistic signature that is stable under benign perturbations and separates bind-then-read from direct-retrieval with margin. The remainder of the paper formalizes these tasks and interventions, defines the signature and certificate predicates, and shows how these objects support both mechanistic certification (sufficient conditions for OOD accuracy) and mechanistic prediction (estimating final OOD performance from early checkpoints).

2 Background: task families, mechanisms, and causal probes

We briefly specify the synthetic task families and the intervention-based diagnostic that we will use throughout. A task instance is presented as an autoregressive input sequence x consisting of a *document* segment, a divider token, and a *query* segment; the model must predict the next token y_{true} at the query position. The document encodes a set of latent bindings (associations) generated by a simple algorithmic process; the query requests one of these bindings (or an iterated application thereof), and the correct next token is uniquely determined by the underlying rule.

AR (Associative Retrieval). In AR, the document is a multiset of key–value pairs written as tokens, e.g. $(k_1, v_1), \ldots, (k_n, v_n)$ in some linearized format with separators. The query presents a key k_q that appears in the document, and the correct next token is the associated value v_q. This family isolates the elementary binding operation: the model must identify the value token aligned with the queried key and output it. The generative process intentionally permits many superficial cues (frequency, position, local n-grams) to be non-informative, so that the intended computation is "bind then read" rather than pattern matching on incidental statistics.

ATR and ATR++ (iterated/ambiguous retrieval). ATR extends AR by requiring *structured* retrieval in which the binding rule is composed along a small graph, typically a rooted tree encoded in the document. Concretely, the document specifies parent pointers (or labeled edges) among entities; the query asks for an ancestor (or otherwise composed) lookup, such as "return the label/value of the k-hop ancestor of node u". ATR++ further modifies the generation to include controlled ambiguity and additional nuisance structure. One convenient way to view ATR++ is as ATR plus (i) distractor bindings that are locally plausible but globally inconsistent, and/or (ii) multiple candidates sharing partial identifiers, so that heuristics that succeed on the development distribution (e.g. "choose the most recent match" or "prefer a frequent label") can be made to fail on compositional splits. The salient point for our purposes is not the particular linearization, but that these families share a common latent variable: an association s that is defined in the document and must be routed to the query position, possibly after a bounded number of compositions.

Two competing mechanism classes. For AR/ATR-type instances there are (at least) two qualitatively distinct internal strategies that can be realized by standard residual sequence models. The *bind-then-read* (induction-like) mechanism computes an intermediate representation of the binding variable s at a locus tied to the document (e.g. at the value token, or at a designated memory token/port), and later reads out s to produce y_{true} when the query is processed. In contrast, a *direct-retrieval* mechanism does not form a stable intermediate binding representation at a document locus; rather, it produces the answer by a shortcut computation localized to the query position and/or late layers (e.g. a lookup keyed by query-side features that correlates with the development compositions). Our certification goal will require distinguishing these mechanisms via internal causal evidence rather than by behavioral fit.

Interchange interventions and likelihood restoration. We operationalize "internal causal evidence" using interchange interventions on pairs of inputs. Let o be a clean instance sampled from the task distribution and let c be a corrupted counterpart obtained by corrupting exactly one crucial token (for example, changing the queried key, swapping a value token, or altering a single edge in the ATR document). We write $p_m(\cdot \mid x)$ for the model's next-token distribution. Denote

$$a_{\text{clean}} := p_m(y_{\text{true}} \mid o), \qquad a_{\text{corr}} := p_m(y_{\text{true}} \mid c).$$

Given a site f (a module boundary such as a block input/output, mixer output, MLP output, or memory read port) and a token position i (chosen by role: key/value/query/memory), we form an intervened run in which we overwrite the internal representation of c at (f, i) with that from o, yielding a distribution $p_{m,f \leftarrow f^*}(\cdot \mid c, o)$ and

$$a_{\text{int}} := p_{m,f \leftarrow f^*}(y_{\text{true}} \mid c, o).$$

We then define the normalized likelihood-restoration attribution score

$$\text{Attrib}_m(f, i; o, c) \ := \ \frac{a_{\text{int}} - a_{\text{corr}}}{a_{\text{clean}} - a_{\text{corr}}},$$

interpreting it as the fraction of the corruption-induced likelihood drop that is causally repaired by restoring the representation at (f, i). When $a_{\text{clean}} > a_{\text{corr}}$ and the intervention does not overshoot, this quantity lies in $[0, 1]$ and is comparable across checkpoints and architectures. Intuitively, if the model has constructed the binding variable s at a document locus, then restoring that locus should strongly restore the correct likelihood; if instead the model computes the answer only at the query locus, then restoration should concentrate at query-side sites. Aggregating Attrib across a small number of (o, c) pairs yields the mechanistic signatures used in subsequent sections.

3 Problem formulation: certification and prediction under compositional shift

We fix a task-instance family $\mathcal{T}$ (AR/ATR/ATR++), a model class $\mathcal{M}$ of residual sequence models with identifiable module boundaries, and a training procedure that produces a sequence of checkpoints $(m_t)_{t=0}^{T}$. Training is performed on $\mathcal{D}_{\text{train}}$; model selection is permitted to consult a development distribution $\mathcal{D}_{\text{dev}}$ drawn from the same generative regime as training; evaluation is on a specified out-of-distribution split $\mathcal{D}_{\text{ood}}$ that is *compositional* in the sense that the underlying binding rule is unchanged while particular compositions are held out. Our goal is to use limited white-box causal access at an early checkpoint m_t (typically $t \ll T$) to either (i) *certify* high final OOD accuracy, or (ii) *predict* it well enough to support early stopping and architecture/hyperparameter selection.

Mechanistic Certification Problem (MCP). An instance of MCP consists of $(\mathcal{T}, \mathcal{M})$, a fixed OOD split specification (known to the evaluator), and an intervention oracle that, for selected sites/tokens, can compute likelihood-restoration attributions $\text{Attrib}_{m_t}(f, i; o, c)$ from clean–corrupted pairs (o, c)

sampled from $\mathcal{D}_{\text{dev}}$ with a designated corruption operator Corr. We must output (a) a signature extraction map $\phi : \mathcal{M} \to \mathbb{R}^d$ computable from at most N such pairs and a bounded number of forward passes per pair, and (b) a certificate predicate $\mathsf{Cert} : \mathbb{R}^d \to \{0, 1\}$. The intended interpretation is: if $\mathsf{Cert}(\phi(m_t)) = 1$, then the training run is forced (up to an explicit failure probability) to reach a final checkpoint m_T with high OOD accuracy. Formally, for target parameters (δ, β) we seek a *sound* certificate of the form

$$\Pr\big[A_{\text{ood}}(m_T) \geq 1 - \delta \;\big|\; \mathsf{Cert}(\phi(m_t)) = 1\big] \;\geq\; 1 - \beta,$$

where the probability is over the random draw of training data, optimization noise, and the sampling used to estimate ϕ. Since a vacuous predicate ($\mathsf{Cert} \equiv 0$) is always sound, we also require *coverage*: $\Pr[\mathsf{Cert}(\phi(m_t)) = 1]$ should be nontrivial over the distribution of training runs, subject to the same intervention budget. Thus MCP is a constrained design problem: we trade intervention cost and coverage against the strength of the OOD guarantee.

Mechanistic Prediction Problem (MPP). MPP relaxes certification to quantitative forecasting. Here we again compute $\phi(m_t)$ from bounded interventions at an early checkpoint, but instead of a Boolean predicate we output a predictor $\widehat{g}$ such that $\widehat{g}(\phi(m_t))$ approximates the eventual $A_{\text{ood}}(m_T)$. The objective is to minimize prediction error (e.g. $\mathbb{E}[|\widehat{g}(\phi(m_t)) - A_{\text{ood}}(m_T)|]$) across training runs that vary architecture, hyperparameters, and seeds. In contrast to MCP, MPP need not be conservative; it is useful whenever accurate ranking of candidates is more valuable than hard guarantees.

OOD split families. We emphasize three OOD regimes, all of which preserve the binding semantics but break spurious shortcuts. First, *held-out query–answer compositions*: the vocabulary of keys/values (or entity identifiers/labels) is shared across splits, but specific query–answer pairings are excluded from $\mathcal{D}_{\text{train}}$ and appear only in $\mathcal{D}_{\text{ood}}$; this targets memorization of frequent compositions. Second, *held-out structures*: in ATR/ATR++ the document encodes a small graph (typically a rooted tree); $\mathcal{D}_{\text{ood}}$ holds out particular structural motifs (e.g. branching patterns, edge-label configurations, or ancestor-query templates) while keeping local edge semantics fixed, so that only genuinely compositional reasoning transfers. Third, *length extrapolation*: $\mathcal{D}_{\text{ood}}$ increases sequence length, number of bindings, or hop count beyond the training range; the rule is identical, but superficial correlations with absolute position or depth no longer match.

These formulations isolate what mechanistic access must provide: evidence, visible at early training time and under a strict intervention budget, that the model has implemented a binding computation that is invariant to the specified compositional shift. The next section defines the intervention sites, corruption distributions, and invariant features that constitute ϕ and support Cert.

4 Mechanistic signatures: sites, corruptions, invariants, and certification

We now define the mechanistic signature $\phi(m_t)$ and the associated certificate predicate Cert in terms of *interchange interventions* on a fixed, architecture-agnostic set of module boundary sites. Throughout, x denotes an input sequence (document $\|$ divider $\|$ query), o a clean instance, $c = \mathrm{Corr}(o)$ its corrupted counterpart, and y_{true} the next-token answer for o at the query position.

Intervention sites and token roles. Let $\mathcal{F}$ be a finite set of intervention sites, each corresponding to an identifiable boundary in a residual block. Concretely, for each layer ℓ we include (when present) the block input, the sequence-mixer output (attention/SSM/conv), and the MLP output; for architectures with explicit memory, we additionally include read/write ports. For an input x, a site $f \in \mathcal{F}$ produces token-indexed representations $f(x)[i] \in \mathbb{R}^{d_{\mathrm{model}}}$. We also fix a small set of *token roles* $\mathcal{R}$ determined by the task template (e.g. DocKey, $\mathsf{DocValue}$, Query, and optionally Memory); each role $r \in \mathcal{R}$ corresponds to a deterministic rule selecting one or a few token indices $i \in \{1, \ldots, |x|\}$. A *site–token* pair is thus an element $(f, i) \in \mathcal{F} \times \{1, \ldots, |x|\}$, typically chosen by composing f with a role-based index rule.

Corruption distribution. The corruption operator Corr must alter exactly one *crucial token*—a token that participates in the binding relation that determines y_{true}. In AR this may be the value token paired with the queried key; in ATR/ATR++ it may be an edge label or a node identifier on the unique path relevant to the query (or, for k-hop queries, one of the required ancestor links). Formally, Corr samples a role r_{crucial} and an index $i^\star$ according to the instance template, and replaces $x_{i^\star}$ by a uniformly sampled token of the same syntactic type (preserving format constraints). We always evaluate likelihoods at the *clean* answer y_{true}; thus a successful corruption produces a substantial drop in $p_{m_t}(y_{\mathrm{true}} \mid x)$ without changing superficial statistics such as length or delimiter placement.

Likelihood-restoration attribution. Given (o, c) and a site–token pair (f, i), we define the interchange-intervened distribution

$$p_{m_t, f \leftarrow f^\star}(\cdot \mid c, o)$$

to be the next-token distribution produced by running m_t on input c while replacing the activation $f(c)[i]$ with $f(o)[i]$ (and leaving all other activations as in the run on c). Let

$$a_{\mathrm{clean}} := p_{m_t}(y_{\mathrm{true}} \mid o), \quad a_{\mathrm{corr}} := p_{m_t}(y_{\mathrm{true}} \mid c), \quad a_{\mathrm{int}} := p_{m_t, f \leftarrow f^\star}(y_{\mathrm{true}} \mid c, o).$$

We then use the normalized restoration score

$$\mathrm{Attrib}_{m_t}(f,i;o,c) := \frac{a_{\mathrm{int}} - a_{\mathrm{corr}}}{a_{\mathrm{clean}} - a_{\mathrm{corr}} + \kappa},$$

with a small $\kappa > 0$ to avoid degeneracy when the corruption has negligible effect; in implementation we additionally clip to $[0, 1]$ and discard samples with $a_{\mathrm{clean}} - a_{\mathrm{corr}}$ below a fixed threshold.

Invariant features (the signature). Fix a finite set $S \subseteq \mathcal{F} \times \mathcal{R}$ of site–role pairs; each $(f,r) \in S$ induces a site–token pair $(f, i(r,x))$ on input x. For a checkpoint m_t, we define $\mu_{f,r} := \mathbb{E}[\mathrm{Attrib}_{m_t}(f, i(r,o); o, c)]$ where the expectation is over $o \sim \mathcal{D}_{\mathrm{dev}}$ and $c = \mathrm{Corr}(o)$. The signature $\phi(m_t) \in \mathbb{R}^d$ consists of low-dimensional functionals of $\{\mu_{f,r}\}_{(f,r)\in S}$, specifically: (i) *Token-locus ratio* (bind-then-read vs direct write),

$$\rho := \frac{\sum_f \mu_{f,\mathsf{DocValue}}}{\sum_f \mu_{f,\mathsf{Query}} + \eta},$$

with small $\eta > 0$; (ii) *Layer concentration* at a pre-specified early anchor layer ℓ_0,

$$\Lambda := \frac{\sum_{(f,r)\in S:\,\mathrm{layer}(f)=\ell_0} \mu_{f,r}}{\sum_{(f,r)\in S} \mu_{f,r} + \eta};$$

(iii) *Stability* under benign perturbations $o \mapsto o''$ (e.g. swapping irrelevant bindings or inserting distractors),

$$\mathrm{Stab} := \mathbb{E}\big[\|\phi(m_t;o) - \phi(m_t;o'')\|_2\big];$$

and (iv) *Sparsity* of the normalized attribution profile $w_{f,r} := \mu_{f,r}/(\sum_{(f',r')} \mu_{f',r'} + \eta)$, measured via normalized entropy

$$\mathrm{Sparse} := 1 - \frac{-\sum_{(f,r)} w_{f,r} \log(w_{f,r} + \eta)}{\log |S|}.$$

Certificate predicate. We take $\mathsf{Cert}(\phi(m_t)) = 1$ iff simultaneously: (a) $\rho \geq 1+\gamma$ (value-locus dominance); (b) $\Lambda \geq 1-\epsilon$ (early anchor concentration); (c) Stab $\leq \tau$ (mechanistic invariance to benign changes); (d) Sparse $\geq s_0$ and the attribution mass on late-layer query sites is at most ϵ (excluding late direct-write behavior). The thresholds $(\gamma, \epsilon, \tau, s_0)$ are fixed per task family and OOD split specification; they are tuned to prioritize soundness under the intervention budget and then maximize coverage.

5 Algorithms: signature estimation, OOD prediction, and model selection

We now specify the practical procedures by which we (i) estimate $\phi(m_t)$ from a checkpoint using a bounded number of interchange interventions, (ii) optionally fit a lightweight predictor of final OOD accuracy from early signatures, and (iii) use either the certificate Cert or the predictor for early stopping and architecture/hyperparameter selection.

Signature estimation from checkpoints. Fix a checkpoint m_t and a finite site–role set $S \subseteq \mathcal{F} \times \mathcal{R}$. For $j = 1, \ldots, N$, we sample a clean instance $o_j \sim \mathcal{D}_{\text{dev}}$ and set $c_j = \text{Corr}(o_j)$. We compute the base likelihoods $a^{(j)}_{\text{clean}} = p_{m_t}(y_{\text{true}} \mid o_j)$ and $a^{(j)}_{\text{corr}} = p_{m_t}(y_{\text{true}} \mid c_j)$ once per pair. If $a^{(j)}_{\text{clean}} - a^{(j)}_{\text{corr}}$ is below a fixed threshold (indicating an ineffective corruption) we discard the pair and resample to maintain a stable denominator in Attrib. Otherwise, for each $(f, r) \in S$ we perform a single interchange intervention at token index $i(r, \cdot)$, producing $a^{(j)}_{\text{int}}(f, r) = p_{m_t, f \leftarrow f^\star}(y_{\text{true}} \mid c_j, o_j)$, and record $\text{Attrib}_{m_t}(f, i(r, o_j); o_j, c_j)$ (with clipping as described earlier). We then form empirical means

$$\hat{\mu}_{f,r} := \frac{1}{N} \sum_{j=1}^{N} \text{Attrib}_{m_t}(f, i(r, o_j); o_j, c_j), \qquad (f, r) \in S,$$

and compute $\phi(m_t)$ as the prescribed low-dimensional function of $\{\hat{\mu}_{f,r}\}$. In terms of forward evaluations, the naive implementation requires $2 + |S|$ forward runs per sampled pair (clean, corrupted, and one intervened corrupted run per site–role). In practice we reduce overhead by caching the clean activations $\{f(o_j)[i(r, o_j)]\}_{(f,r) \in S}$ and injecting them via activation hooks during the corrupted run; this preserves the semantics of an interchange intervention while avoiding repeated clean passes. We also batch over $(f, r) \in S$ across multiple devices, since interventions for distinct sites are independent conditional on (o_j, c_j).

Predictor training from signatures (optional). To obtain a quantitative forecast of $A_{\text{ood}}(m_T)$ from early checkpoints, we train a simple regressor $\widehat{g}$ on tuples $(\phi(m_t), A_{\text{ood}}(m_T))$ collected across multiple training runs and architectures/hyperparameters. We restrict $\widehat{g}$ to low-capacity families (e.g. ridge regression, lasso, or a monotone isotonic map in one or two signature coordinates) to discourage spurious fit to run-specific artifacts. We standardize each signature coordinate using statistics computed on the training runs only, and we cross-validate across entire runs (not across checkpoints within a run) to avoid leakage from temporally correlated signatures. At inference time, $\widehat{g}(\phi(m_t))$ is evaluated for one or more early checkpoints $t \ll T$; we either average these predictions or use the maximum over a short window to improve robustness to transient training noise.

Early stopping and architecture selection. We employ two complementary decision rules. First, for soundness-first selection, we stop a run (or accept an architecture) once $\mathsf{Cert}(\phi(m_t)) = 1$ holds for L consecutive checkpoints and the signature remains stable under a fixed set of benign perturbations; this implements a conservative "certify-then-continue" policy in which additional training is optional. Second, when coverage is prioritized, we rank candidates by $\widehat{g}(\phi(m_{t_0}))$ for a fixed early time t_0 (e.g. 5–20% of training), allocate full training budget only to the top-ranked subset, and optionally re-rank after a second early checkpoint. In both cases we treat dev accuracy only as a debugging signal; it is not used as a selection criterion except to filter obviously failed runs.

Implementation details and reproducibility checklist. We record, for each experiment:

- the exact task generator specification (AR/ATR/ATR++ parameters, ambiguity controls, and the OOD split definition);
- the corruption operator Corr (role distribution, syntactic-type constraints, and rejection criteria);
- the site set $\mathcal{F}$ and role-to-index rules defining S, including layer numbering conventions;
- the intervention mechanism (hook locations, whether interventions occur at block input/output, and numerical precision);
- thresholds (κ, η) and all certificate parameters $(\gamma, \epsilon, \tau, s_0)$, together with the rule by which they are tuned;
- sample sizes N and the full accounting of forward passes per checkpoint, including batching strategy;
- random seeds for data sampling, model initialization, and checkpoint selection, plus deterministic-evaluation settings when available.

These items suffice to reconstruct $\phi(m_t)$ and Cert evaluations, and to reproduce predictor training and model-selection decisions.

6 Theory II: certification guarantees

We now formalize the sense in which a mechanistic certificate constitutes a sufficient condition for strong compositional OOD generalization on the

AR/ATR/ATR++ family. Fix a task distribution with a prescribed compositional split $(\mathcal{D}_{\mathrm{dev}}, \mathcal{D}_{\mathrm{ood}})$ such that the OOD instances preserve the same binding map (e.g. the same ancestor function, pointer-following rule, or key–value association) but hold out a subset of query–answer compositions. Let $s(x)$ denote the (latent) binding variable computed by the task for an input x (for AR/ATR, s is the unique key-selected value; for ATR++ with controlled ambiguity, s is a distribution over admissible values).

The certificate predicate $\mathsf{Cert}(\phi(m_t))$ is designed to assert that the model has implemented a *bind-then-read* computation in the following operational sense: there exists an *anchor* site–token pair (f_v, i_v) (typically a value-role position at an intermediate layer) such that swapping $f_v(\cdot)[i_v]$ from clean to corrupted input restores essentially all of the lost likelihood mass for the correct next token, while analogous swaps at late query sites do not. Concretely, we take Cert to enforce three inequalities at the population level (up to constants absorbed into thresholds): (i) *anchor completeness*, $\mu_{f_v,i_v} \geq 1 - \epsilon$; (ii) *low late-write*, $\sum_{(f,i)\in S_{\mathrm{late}}} \mu_{f,i} \leq \epsilon_{\mathrm{late}}$ for a designated set of late query-adjacent sites; and (iii) *stability*, $\|\phi(m_t; x) - \phi(m_t; x'')\| \leq \tau$ for benign perturbations x'' that preserve $s(x)$ (e.g. irrelevant token swaps or distractor insertions as specified by the task generator). The intended semantics is that (i) identifies where s is first materialized, (ii) rules out direct late writing of the answer token as a shortcut, and (iii) excludes brittle heuristics tied to incidental surface features.

Under the algorithm-class hypothesis for bind-then-read models (as in the separability analysis), these conditions imply an abstract causal model: the computation produces a representation $\tilde{s}(x)$ at the anchor such that intervening on $f_v(x)[i_v]$ is (approximately) an intervention on $\tilde{s}$, and subsequent computation from the anchor to the query is a stable readout that depends on $\tilde{s}$ but not on the particular composition of key/value identities seen during training. Since the OOD split changes compositions while preserving the binding rule, the induced distribution of $\tilde{s}(x)$ over OOD inputs matches that of $s(x)$ up to the task's intrinsic ambiguity; thus the readout continues to map $\tilde{s}$ to the correct next token. Formally, one can couple a dev draw $o \sim \mathcal{D}_{\mathrm{dev}}$ and an OOD draw $x \sim \mathcal{D}_{\mathrm{ood}}$ with the same latent s and apply a causal abstraction argument: the certificate enforces that the model's prediction depends on x only through $\tilde{s}(x)$ and a stable downstream map, so any shift that preserves the distribution of s does not substantially change the correctness probability. The resulting bound has the form

$$A_{\mathrm{ood}}(m_T) \;\geq\; 1 - \delta(\epsilon, \epsilon_{\mathrm{late}}, \tau; \alpha_{\mathrm{amb}}),$$

where α_{amb} is the task-specific ambiguity rate (zero for unambiguous AR/ATR; controlled for ATR++) and δ grows at most linearly in $(\epsilon, \epsilon_{\mathrm{late}}, \tau)$ under the linearized residual-stream approximation. In particular, when $\alpha_{\mathrm{amb}} = 0$ and the certificate is tight (small $\epsilon, \epsilon_{\mathrm{late}}, \tau$), we obtain $A_{\mathrm{ood}}(m_T) \geq 1 - O(\epsilon + \epsilon_{\mathrm{late}} + \tau)$.

The same reasoning is robust to imperfections in the intervention operator. Suppose our implemented interchange replaces $f(c)[i]$ by $f(o)[i]$ up to an additive perturbation e with $\|e\| \leq \xi$ (capturing finite precision, hook misalignment, or minor nondeterminism). If the model's downstream map from the intervened site to the output distribution is L-Lipschitz in the intervened subspace, then each attribution estimate incurs at most $O(L\xi)$ additive error, and the certification bound degrades by an additional $O(L\xi)$ term. Thus, certification is stable provided the intervention noise is controlled relative to the certificate margins.

Finally, we require that Cert be *decidable* from finitely many sampled intervention pairs. Let $\hat{\mu}_{f,i}$ be empirical means over N i.i.d. draws, and assume Attrib $\in [0,1]$ after clipping. By Hoeffding's inequality and a union bound over the finitely many coordinates used by Cert, if

$$N \;\geq\; \frac{1}{2\varepsilon^2} \log \frac{2|S_{\text{cert}}|}{\beta},$$

then with probability at least $1-\beta$ we have $\max_{(f,i)\in S_{\text{cert}}} |\hat{\mu}_{f,i} - \mu_{f,i}| \leq \varepsilon$. Consequently, if the certificate inequalities hold with slack at least $s_0 > 0$ (a margin condition), choosing $\varepsilon < s_0/2$ yields an *empirically checkable* predicate $\widehat{\mathsf{Cert}}$ such that $\widehat{\mathsf{Cert}}(\phi(m_t)) = \mathsf{Cert}(\phi(m_t))$ with probability at least $1-\beta$. This is the sense in which certification provides a probabilistic guarantee on OOD accuracy while using only $\tilde{O}(|S_{\text{cert}}|/\varepsilon^2)$ intervention samples at an early checkpoint.

7 Theory III: behavior-only lower bound

We now record an obstruction: even perfect in-distribution performance (indeed, identical in-distribution predictive distributions) cannot, in general, certify strong compositional OOD generalization on the AR/ATR/ATR++ family over a broad model class. The argument is information-theoretic and does not depend on training dynamics; it shows that any putative certificate that inspects only dev behavior must fail in the worst case.

Fix a task distribution $(\mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{ood}})$ from $\mathcal{T}$ with a compositional split. For definiteness, consider an AR instance format in which the input contains a list of key–value bindings and a query key, and the correct next token is the value bound to that key under the task's binding rule. Let Σ be the output vocabulary of admissible values. The split is chosen so that $\mathcal{D}_{\text{dev}}$ contains only a designated subset of key–value compositions (e.g. pairs drawn from an "allowed edge set" E_{dev}), while $\mathcal{D}_{\text{ood}}$ contains query–answer compositions from a disjoint subset E_{ood}; crucially, the underlying rule "return the value associated to the queried key" is identical in both. Analogous splits exist for ATR (pointer-following) and ATR++ (k-hop ancestry and controlled ambiguity), where the OOD split holds out specific relation compositions while preserving the graph traversal rule.

We formalize "behavior-only" as dependence solely on the model's conditional output distributions on dev inputs. That is, a behavioral statistic is any functional

$$B(m) \;=\; \Psi(\{p_m(\cdot \mid x) : x \in \mathrm{supp}(\mathcal{D}_{\mathrm{dev}})\}) \,,$$

including dev accuracy $A_{\mathrm{dev}}(m)$, dev log-likelihood, calibration curves, or even the full table of next-token distributions on every dev input (in the idealized setting where this table is accessible). A behavior-only certificate is then a predicate $\mathsf{Cert}_{\mathrm{beh}}(m)$ that factors through $B(m)$.

The obstruction is that dev behavior constrains $p_m(\cdot \mid x)$ only on $\mathrm{supp}(\mathcal{D}_{\mathrm{dev}})$, leaving $p_m(\cdot \mid x)$ on $\mathrm{supp}(\mathcal{D}_{\mathrm{ood}})$ unconstrained. Because $\mathcal{M}$ contains sufficiently expressive sequence models, we can realize two models that coincide on all dev inputs but disagree arbitrarily on OOD inputs. Concretely, let m_{alg} be any model that implements the binding rule (e.g. a bind-then-read mechanism that identifies the queried key and routes the associated value), so that $A_{\mathrm{ood}}(m_{\mathrm{alg}})$ is close to 1 on the compositional split. Define a second model m_{mem} by the following specification of its input–output behavior:

1. for every $x \in \mathrm{supp}(\mathcal{D}_{\mathrm{dev}})$, set $p_{m_{\mathrm{mem}}}(\cdot \mid x) = p_{m_{\mathrm{alg}}}(\cdot \mid x)$ (thus matching dev predictions pointwise, not merely in expectation);

2. for every $x \in \mathrm{supp}(\mathcal{D}_{\mathrm{ood}})$, set $p_{m_{\mathrm{mem}}}(\cdot \mid x)$ to place all mass on a fixed incorrect token, or to be uniform over Σ.

By construction, $B(m_{\mathrm{mem}}) = B(m_{\mathrm{alg}})$ for any behavior-only statistic B, hence $\mathsf{Cert}_{\mathrm{beh}}(m_{\mathrm{mem}}) = \mathsf{Cert}_{\mathrm{beh}}(m_{\mathrm{alg}})$ for any behavior-only predicate. Yet their OOD accuracies can be separated maximally:

$$A_{\mathrm{ood}}(m_{\mathrm{alg}}) \;\geq\; 1-\delta \qquad \text{while} \qquad A_{\mathrm{ood}}(m_{\mathrm{mem}}) \;\leq\; \frac{1}{|\Sigma|} + \delta,$$

where the additive δ accounts for any intrinsic ambiguity in ATR++ (and may be taken 0 in unambiguous AR/ATR). The only remaining point is realizability: within a broad residual sequence model family, such an m_{mem} can be implemented by a table-lookup (memorization) subnetwork keyed on exact dev-seen compositions, combined with a default fallback on unseen compositions; because E_{dev} and E_{ood} are disjoint by design, the fallback dominates OOD.

The conclusion is unavoidable: no certificate that depends only on dev behavior can guarantee a nontrivial lower bound on A_{ood} uniformly over $\mathcal{M}$ for these compositional splits. Equivalently, for any behavior-only certification rule, either it rejects some genuinely compositional model, or it accepts a model whose OOD accuracy is near chance. This is precisely the gap mechanistic access is meant to close: interchange interventions expose where and how the binding variable is represented, thereby ruling out the indistinguishable-but-spurious memorization solutions that dev metrics cannot exclude.

Theory III (Behavior-only lower bound). We isolate a generic impossibility statement: for the compositional splits used in AR/ATR/ATR++, any purported guarantee of strong OOD performance that is derived solely from in-distribution behavior must fail uniformly over a sufficiently expressive model family. The point is not that dev metrics are uninformative in practice, but that they cannot support worst-case certificates.

Theorem 7.1 (No behavior-only certificate yields a nontrivial OOD lower bound)**.** *Fix an output vocabulary* Σ *and a task family* $\mathcal{T}$ *containing compositional splits* $(\mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{ood}})$ *in which the underlying semantic rule is shared across splits while specific query–answer compositions are held out OOD. Let* $\mathcal{M}$ *be any model class that can realize arbitrary conditional distributions on* $\operatorname{supp}(\mathcal{D}_{\text{dev}}) \cup \operatorname{supp}(\mathcal{D}_{\text{ood}})$ *(e.g. via a sufficiently large residual sequence model). Then for any functional*

$$B(m) \;=\; \Psi(\{p_m(\cdot \mid x) : x \in \operatorname{supp}(\mathcal{D}_{\text{dev}})\})$$

and any predicate $\mathsf{Cert}_{\text{beh}}(m)$ *that depends only on* $B(m)$*, there exist* $m, m' \in \mathcal{M}$ *such that* $B(m) = B(m')$ *while* $A_{\text{ood}}(m)$ *is close to* 1 *and* $A_{\text{ood}}(m')$ *is close to chance. In particular, no* $\mathsf{Cert}_{\text{beh}}$ *can imply a uniform bound of the form* $A_{\text{ood}}(m) \geq 1 - \delta$ *with* $\delta < 1 - 1/|\Sigma|$*.*

Proof sketch. We proceed by explicit construction on a representative split and then note that the same logic applies to ATR and ATR++. Consider an AR-type instance in which an input x encodes a finite set of key–value bindings and a query key; the correct next token y_{true} is the value associated to the queried key. Let the compositional split be determined by a partition of admissible key–value pairs into E_{dev} and E_{ood} with $E_{\text{dev}} \cap E_{\text{ood}} = \emptyset$, such that $\mathcal{D}_{\text{dev}}$ only contains bindings in E_{dev} while $\mathcal{D}_{\text{ood}}$ only contains bindings in E_{ood}. The semantic rule is identical across splits; only the allowed compositions differ.

Let $m_{\text{alg}} \in \mathcal{M}$ be a model that implements the underlying binding rule (e.g. by computing a key index, selecting the matching value, and routing it to the next-token distribution). By construction of the split, such a model achieves $A_{\text{ood}}(m_{\text{alg}}) \approx 1$ (or $\geq 1 - \delta$ if the task admits controlled ambiguity, as in ATR++). Now define a second model $m_{\text{sp}} \in \mathcal{M}$ by specifying its conditional distributions as follows: (i) for every $x \in \operatorname{supp}(\mathcal{D}_{\text{dev}})$, set $p_{m_{\text{sp}}}(\cdot \mid x) = p_{m_{\text{alg}}}(\cdot \mid x)$; (ii) for every $x \in \operatorname{supp}(\mathcal{D}_{\text{ood}})$, set $p_{m_{\text{sp}}}(\cdot \mid x)$ to be uniform over Σ (or concentrated on a fixed wrong label). Because $B(m)$ is a functional of $\{p_m(\cdot \mid x)\}$ restricted to dev support, we have $B(m_{\text{sp}}) = B(m_{\text{alg}})$, hence $\mathsf{Cert}_{\text{beh}}(m_{\text{sp}}) = \mathsf{Cert}_{\text{beh}}(m_{\text{alg}})$. On the other hand, $A_{\text{ood}}(m_{\text{sp}}) \leq 1/|\Sigma|$ (up to the same ambiguity slack), while $A_{\text{ood}}(m_{\text{alg}}) \geq 1 - \delta$.

It remains only to justify realizability of m_{sp} inside $\mathcal{M}$. Since $\mathcal{M}$ includes expressive residual sequence models, we can implement m_{sp} as a mixture of two modules: a memorization component that matches $p_{m_{\text{alg}}}(\cdot \mid x)$ on the finite (or effectively finite) set of dev compositions, and a fallback component that emits the chosen default distribution on any input not matching the dev composition patterns. The disjointness of E_{dev} and E_{ood} ensures the fallback controls OOD behavior, yielding near-chance OOD accuracy without affecting dev behavior. □

The conclusion is that even access to the full dev conditional distribution table does not suffice to certify compositional OOD performance over $\mathcal{M}$. Accordingly, any successful certification scheme must exploit information unavailable to behavior-only statistics, such as causal localization of the binding variable via interventions, which is precisely what our mechanistic signatures are designed to provide.

Experiments. We evaluate the practical utility of mechanistic signatures via broad sweeps over architectures, hyperparameters, and training time, using the AR/ATR/ATR++ family with fixed compositional OOD splits. Our experimental unit is a *run* producing checkpoints $\{m_t\}$ up to a final time T. For each run we record $(A_{\text{dev}}(m_T), A_{\text{ood}}(m_T))$, and for a collection of early checkpoints $t \ll T$ we compute $\phi(m_t)$ using Algorithm 1 with a fixed intervention budget. Architecturally, we span (i) standard attention-only residual models, (ii) linear/"Based" attention variants, (iii) Mamba-like selective-SSM mixers, (iv) Hyena-like long convolutional mixers, and (v) hybrids that alternate attention and SSM/conv blocks. Within each family we vary depth, width, learning rate schedule, normalization style, and (where applicable) kernel length or state dimension; for positional information we include sinusoidal, learned absolute, and relative/rotary schemes.

Our primary comparison is between *mechanistic predictors* based on $\phi(m_t)$ and *behavior-only predictors* based on in-distribution quantities at the same checkpoint, e.g. $A_{\text{dev}}(m_t)$, dev log-likelihood, or training loss. Concretely, we train a simple regressor $\widehat{g}$ (linear or shallow MLP) to predict $A_{\text{ood}}(m_T)$ from $\phi(m_t)$ across runs, and we compare against the best regressor using only dev-side behavioral features. We also instantiate a threshold certificate Cert using the invariant features in Algorithm 1 (LocusRatio, LayerConcentration, Stability, Sparsity), tuned on held-out runs to target a desired failure probability. The salient empirical phenomenon is that $\phi(m_t)$ extracted very early (typically within the first 5–20% of training) predicts final OOD accuracy substantially better than any dev-only signal: in many sweeps, models with nearly identical $A_{\text{dev}}(m_t)$ separate cleanly in $\phi(m_t)$, and this separation aligns with their eventual compositional generalization. Conversely, behavior-only predictors often saturate: once dev accuracy is high, residual variation in dev metrics carries little information about whether the model has learned an algorithmic binding mechanism or a shortcut tied to the dev compositions.

We further use the certificate operationally for early stopping and model selection. Given a run, we stop at the first checkpoint t such that $\mathsf{Cert}(\phi(m_t)) = 1$, and we report the resulting OOD accuracy at that checkpoint as well as the final-time OOD accuracy had we continued training. Across architectures, certificate-based early stopping frequently yields large reductions in training compute while preserving (and occasionally improving) OOD performance, consistent with the hypothesis that later training can overfit to dev compositions without strengthening the underlying binding mechanism. For architecture selection, we compare (a) picking the architecture/hyperparameters with best dev accuracy at an early time, versus (b) picking the candidate with best predicted $A_{\text{ood}}(m_T)$ under $\widehat{g}(\phi(m_t))$; the latter is markedly more reliable under compositional splits, especially when the candidate set mixes attention and non-attention mixers where dev curves can be misleadingly similar.

Finally, we perform ablations to test which components of ϕ carry signal and how this signal depends on architectural knobs. Reducing depth tends to shift attribution mass toward late-layer query sites (lower LocusRatio and LayerConcentration at intermediate value sites), matching a transition toward direct-retrieval behavior and degraded OOD accuracy. In Hyena-/conv-like models, shortening kernel size or receptive field produces a similar shift, suggesting that insufficient mixing length impedes formation of an intermediate binding variable. Changing positional encoding can either stabilize or destabilize the signature: relative/rotary schemes often improve the Stability feature under benign perturbations (token swaps/distractors) compared to learned absolute embeddings, and this change tracks OOD robustness. On the signature side, removing Stability or Sparsity from Cert increases false positives (models that pass the remaining thresholds yet fail OOD), while using only a single locus statistic (e.g. LocusRatio alone) misses cases where the binding is present but routed unreliably. These ablations support the interpretation that the predictive advantage arises not from any one heuristic, but from jointly localizing *where* the binding variable is computed and *how* stably it controls the next-token distribution.

8 Discussion and Limitations

Our guarantees rest on an explicit separation hypothesis between two algorithmic classes and on a linearized residual-stream approximation used to connect interchange interventions to likelihood restoration. When this approximation is accurate—e.g. when the next-token logit for y_{true} depends approximately affinely on a low-dimensional "association" variable carried in the residual stream—the attribution score

$$\text{Attrib}_m(f, i; o, c) \;=\; \frac{a_{\text{int}} - a_{\text{corr}}}{a_{\text{clean}} - a_{\text{corr}}}$$

tracks (up to estimation noise) the fraction of task-relevant variance localized at (f, i). However, several regimes weaken this link. First, strong nonlinear gating (softmax attention saturation, hard selection in SSMs, or activation clipping) can make $a_{\text{int}} - a_{\text{corr}}$ depend sensitively on higher-order interactions, so that swapping $f(c)[i] \leftarrow f(o)[i]$ changes *which* pathway is active rather than only restoring a missing scalar. In such cases the linearized picture may mis-localize mass, and Attrib need not be well-calibrated as a "fraction explained" (it may even be negative or exceed 1 without additional normalization). Second, if the model represents the binding variable in a distributed subspace that is only readable after a nonlinear mixing step, then no single site-token intervention is "non-degenerate" in the sense required by Theorem 2; the effect may only appear under *joint* interventions across multiple sites or across a span of layers. Third, as depth grows, multiple redundant implementations of the same abstract computation may coexist; interchange at one locus then yields only partial restoration even though the abstract mechanism is present, complicating certificates that expect concentration.

These issues suggest two methodological extensions. One is to replace scalar likelihood restoration by a local logit-difference functional, e.g. $\Delta = \ell_{y_{\text{true}}} - \log \sum_{y \neq y_{\text{true}}} e^{\ell_y}$, and to compute attribution in that space where additivity is empirically closer to linear. Another is to generalize ϕ from single-site statistics to low-rank subspace swaps: we may estimate an association subspace U at (f, i) (e.g. via PCA over clean–corrupted differences) and intervene by swapping only the U-component. This retains the interchange semantics while accommodating distributed codes, at the cost of a larger intervention budget and additional estimation error.

A separate limitation is instrumentation. Algorithm 1 assumes white-box access at identifiable boundaries, but many efficient implementations rely on fused kernels, activation recomputation, or graph-level optimizations that obscure token-indexed intermediate values. Scaling our approach therefore requires either (i) compiler-supported "tap points" that expose $f(x)[i]$ with bounded overhead, or (ii) a reparameterization of sites to match what is observable (e.g. pre/post layernorm, block inputs/outputs, or attention/SSM outputs before fusion). In practice we have found that intervening at coarse residual boundaries often suffices to recover the qualitative signature, but the resulting ϕ is less granular and may reduce separability margins. A principled treatment would quantify how site coarsening contracts attribution distributions and how certificates should be adjusted to preserve false-positive control.

With respect to real tasks, our task family $\mathcal{T}$ is intentionally synthetic: it isolates binding and retrieval while making "one crucial token" corruption meaningful. Natural language and tool-augmented systems typically exhibit (a) multiple partially redundant cues, (b) soft ambiguity rather than controlled ambiguity, and (c) long-range dependencies whose relevant evidence spans many tokens. In such settings, defining o, c pairs requires either structured perturbations (counterfactual edits, entity swaps, adversarial distractors) or supervision about which token is "crucial". Moreover, compositional OOD is not uniquely specified outside of synthetic benchmarks; any certificate can only speak to the declared split, and Theorem 4 cautions that behavior-only indistinguishability constructions persist whenever the OOD shift breaks memorized associations. We therefore view mechanistic certification as a benchmarked capability claim, not as a general safety guarantee.

Finally, we clarify the relation to mechanistic interpretability and to architecture design. Conceptually, ϕ is a compression of causal evidence: it is weaker than a full circuit description but stronger than behavioral metrics, and it interfaces naturally with causal scrubbing and activation patching. Practically, signatures provide actionable feedback for design: if OOD failures correlate with low LocusRatio and diffuse late-layer mass, then increasing effective mixing length (depth, receptive field, state dimension) or introducing explicit memory/binding modules is a targeted intervention. Conversely, when stability features fail under benign perturbations, positional schemes or normalization choices that improve invariance become directly testable. These uses do not eliminate the need for mechanistic inspection, but they impose a disciplined standard: we ask not merely *whether* a model succeeds, but *where* and *how* the success is causally implemented.

Release: mechanistic evaluation harness, configs, and reporting standards. To make the preceding notions operational and comparable across implementations, we will release a mechanistic evaluation harness that instantiates Algorithm 1 end-to-end, together with benchmark configurations for AR/ATR/ATR++ and a small set of recommended signatures and certificates. The harness is designed around a minimal API that separates (i) task instance generation and corruption, (ii) model instrumentation and interchange intervention, and (iii) signature aggregation and certification. Concretely, a model adapter exposes (a) a list of admissible sites $\mathcal{F}$ and token roles R, (b) read/write hooks for activations $f(x)[i]$ at those sites, and (c) a forward function returning next-token logits, so that the harness can compute $(a_{\text{clean}}, a_{\text{corr}}, a_{\text{int}})$ for each sampled (o, c) pair with a bounded number of forward passes.

The benchmark release will include declarative configuration files specifying: the task family member (AR/ATR/ATR++), vocabulary size and sequence length ranges, corruption operator Corr (including the distribution over which token is deemed "crucial"), and the compositional OOD split definition. In ATR++, configs additionally specify ambiguity parameters (e.g. number of confounders, tie-breaking rules) and k-hop query generation. We will also provide reference samplers for $\mathcal{D}_{\text{dev}}$ and $\mathcal{D}_{\text{ood}}$ that are independent of any particular model and that fix random seeds at the instance level, enabling exact reproduction of attribution estimates and the associated concentration bounds.

We will ship a recommended signature set that is intentionally small and interpretable, reflecting the invariants used in our theoretical statements. The default $\phi(m_t) \in \mathbb{R}^d$ includes: (1) *LocusRatio* $\rho = \frac{\mathbb{E}[\text{Attrib(Value)}]}{\mathbb{E}[\text{Attrib(Query)}]+\eta}$ with a fixed η; (2) *LayerConcentration* at a user-declared anchor layer (reported both as a fraction and as a cumulative curve over depth); (3) *Sparsity* of the attribution distribution over $S \subseteq \mathcal{F} \times R$ (e.g. normalized entropy and a Gini coefficient); and (4) *Stability* under benign perturbations, measured as $\|\phi(m_t; x) - \phi(m_t; x'')\|$ for a prescribed family of perturbations $x \mapsto x''$ (distractor insertion, irrelevant swaps, or padding changes). Because some implementations may yield Attrib $\notin [0, 1]$ in nonlinear regimes, the harness will report both raw and clipped scores, and it will flag the fraction of samples exhibiting out-of-range values as a diagnostic rather than silently normalizing them away.

To ensure that reported certificates are meaningful, we will adopt a standard reporting template. Each run must specify: the site set S (including the exact tensor names, layer indices, and whether values are pre/post normalization), the sample size N, the forward-pass budget per sample, and the estimator used (mean of Attrib, median-of-means, or a robust alternative). For each $(f, i) \in S$ we will report $\hat{\mu}_{f,i}$ together with a confidence interval derived from Theorem 1 (or its robust analogue), and we will report whether $\mathsf{Cert}(\phi(m_t))$ holds with an explicit margin. For prediction-style use (MPP), we will require disjoint training and evaluation runs for $\widehat{g}$, and we will report calibration plots of $\widehat{g}(\phi(m_t))$ versus $A_{\text{ood}}(m_T)$ across architectures and hyperparameters.

Finally, we will standardize a small set of ablations that distinguish genuine mechanistic signal from instrumentation artifacts: (i) site coarsening (fine-grained versus residual-boundary sites), (ii) corruption sensitivity (varying which “crucial token” is corrupted), and (iii) intervention locality (single-site swap versus low-rank subspace swap when supported). The goal of the release is not to canonize a single signature, but to make mechanistic claims falsifiable under controlled budgets and comparable across model classes within $\mathcal{M}$.