CERTIFYING MULTI-STEP LOGICAL COMPUTATION IN FIXED-PARAMETER TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

A language model can answer a logical question correctly without necessarily executing the reasoning process that justifies the answer. We study conditions under which a fixed-parameter Transformer can be certified to perform finite-step Horn-rule deduction during inference. We first give sufficient conditions for consistent variable binding, showing that shared state reads together with attention-score separation ensure that repeated occurrences of a variable refer to the same entity across the premises of a rule. We then construct invariant neighborhoods of logical fact assignments using update errors at reference states and bounds on the corresponding Jacobians. These neighborhoods guarantee that derived facts remain available across successive reasoning steps, while local dependency structure can reduce reference-state verification to combinations of only the relevant facts. We further construct fixed-parameter Transformer templates using finite-temperature softmax, RMS normalization, and SiLU that internally enumerate substitutions, read both current and previously derived facts, and update the logical state. Rule templates are shared across substitutions, so the parameters and caches need not store an expanded list of ground rule instances. We give explicit bounds on representation size, parameter count, verification cost, and substitution-enumeration time, and show that the same parameters and invariant neighborhoods support every finite number of repeated scans. The guarantees yield soundness, bounded completeness under rule coverage, and implementation-independent intervention effects. Experiments sustain 10^5 correct macrosteps and separate storage from enumeration costs; exploratory Mistral interventions match 97.08% of eligible nonzero and 64.10% of zero effects.

1 INTRODUCTION

A Transformer can answer a logical question correctly without necessarily carrying out the reasoning process that would justify the answer (Clark et al., 2021). Generating an explicit proof provides stronger behavioral evidence than producing an answer alone, but it still does not establish that the model's internal states execute the corresponding reasoning steps (Saha et al., 2020). More generally, the ability to recover a structure from hidden representations does not by itself show that the same structure governs the model's computation, because a sufficiently expressive readout can perform part of the target computation (Hewitt & Liang, 2019). We therefore ask when a Transformer with fixed parameters can be proved to execute a specified multi-step logical computation during inference, rather than merely produce the correct final answer.

A standard way to relate neural computation to a higher-level process is to map low-level model states to abstract states and require the corresponding state transitions to agree (Geiger et al., 2021). A closely related idea appears in refinement mappings, which characterize when the executions of a lower-level transition system implement those of a higher-level system (Abadi & Lamport, 1991). Such a correspondence is not sufficient, however, if the interpretation of the neural state is allowed to perform arbitrary computation itself (Sutter et al., 2025). We therefore require the representation of logical facts to be shared across inference steps and require changes in those facts to result from the Transformer's own state updates rather than from changing the interpretation from one step to the next.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Earlier neural-symbolic systems compile propositional logic programs into neural networks for deductive computation (Avila Garcez & Zaverucha, 1999). Transformers have also been constructed to execute logical inference step by step. Zhang et al. (2023) give a fixed-parameter BERT construction in which successive layers perform forward chaining on propositional Horn rules and retain the derived facts for subsequent layers. Pan et al. (2025) construct a decoder-only Transformer for bounded-size SAT instances and prove that its chain-of-thought execution simulates an abstract DPLL procedure.

Program compilation also provides persistent intermediate representations. Tracr assigns program variables orthogonal residual subspaces, without layer normalization (Lindner et al., 2023). Looped Transformers implement addressed memory and control flow, correcting finite-softmax errors with ReLU (Giannou et al., 2023). Wei et al. (2022) establish correction-based all-layer margins, including a theorem allowing layerwise parameter sharing. In trained Transformers, Wu et al. (2025) identify variable binding through residual-stream memory and attention routing. Existing constructions therefore establish persistent discrete computation, but they do not, in their stated forms, certify relational Horn reasoning as a stable neural-to-symbolic transition system in which shared variable bindings remain consistent across premise reads, newly derived facts are reused by later steps, and each native update maps an entire reusable neighborhood of continuous states into the neighborhood of the correct symbolic successor. Appendix E.9 gives a detailed comparison with corrected discrete simulations under their respective interfaces.

Our results have four parts. **First**, we give sufficient conditions under which the premises of a rule use a consistent variable substitution in the Transformer's computation. **Second**, we give reusable state-neighborhood conditions for successive logical updates and show that, under a local dependency condition, reference-state verification depends on locally relevant facts rather than on the full fact-assignment space. **Third**, we construct a recurrent Transformer for finite Horn programs using finite-temperature softmax, RMS normalization, and SiLU, with internally generated substitutions and reuse of derived facts. We give explicit bounds on representation size, parameter magnitudes, perturbation radius, verification cost, and substitution-enumeration time, with the same parameters and certified neighborhoods valid for every finite number of scans. **Fourth**, these conditions imply soundness, bounded completeness under rule coverage, and verifiable decision effects of realizable logical interventions. Experiments test persistence, correction, storage, and logical edits, with partial non-oracle agreement between Mistral interventions and symbolic decision effects.

2 Logical states and the fixed decoder

Logical specification. Fix a finite nonempty set of constants, a finite relational vocabulary without function symbols, and the set $\mathcal{A}$ of ground atoms, with $n_{\mathrm{at}} = |\mathcal{A}|$. A finite definite Horn program $\mathcal{R}$ determines a finite set $\mathcal{G}$ of ground rule instances. Each $g \in \mathcal{G}$ records one substitution θ_g assigning a constant to every variable of its rule, a body $\mathsf{B}_g \subseteq \mathcal{A}$ containing its premises, and a head $\mathsf{h}_g \in \mathcal{A}$ giving its conclusion; empty bodies are allowed. Given input facts $\Gamma \subseteq \mathcal{A}$ and a ground query q, the task is to decide whether q belongs to the least deductive closure $\mathrm{Cl}_{\mathcal{R}}(\Gamma)$ (Van Emden & Kowalski, 1976). Absence from the closure is not a proof of a negative literal.

For fact bits $k \in \{0,1\}^{n_{\mathrm{at}}}$ write $\mathsf{K}(k) = \{a : k_a = 1\}$. For each $g \in \mathcal{G}$, define the single-rule update $\mathsf{U}_g : \{0,1\}^{n_{\mathrm{at}}} \to \{0,1\}^{n_{\mathrm{at}}}$ by

$$\mathsf{K}(\mathsf{U}_g(k)) = \begin{cases} \mathsf{K}(k) \cup \{\mathsf{h}_g\}, & \mathsf{B}_g \subseteq \mathsf{K}(k), \\ \mathsf{K}(k), & \text{otherwise.} \end{cases} \tag{1}$$

The idle update $\mathsf{U}_\circ$ is the identity. An executable abstract state is $z = (k, c_{\mathrm{ctrl}})$, where $c_{\mathrm{ctrl}} \in \mathcal{C}$ is a control state in a finite set, recording the information needed to select and execute a rule. A high-level transition is

$$H_t(k, c_{\mathrm{ctrl}}) = \big(\mathsf{U}_{\sigma_t(k,c_{\mathrm{ctrl}})}(k), \tau_t(k, c_{\mathrm{ctrl}})\big),\ \sigma_t(k, c_{\mathrm{ctrl}}) \in \mathcal{G} \cup \{\circ\}, \tag{2}$$

with supplied total functions σ_t (the rule selector) and τ_t (the control update). The selector, bindings, and control state are therefore included in the abstract computation.

Native execution and interpretation. A *native boundary* is a specified point in the original decoder execution at which its complete state is specified. Fix $T+1$ such boundaries, indexed by $t = 0, \ldots, T$.

The original operations between consecutive boundaries form a *block*. A fixed decoder induces a native transition map between each pair of consecutive boundaries: $F_t : X_t \longrightarrow X_{t+1},\ t = 0, \ldots, T-1$. Here X_t is the declared native-state domain at boundary t, and T is the number of blocks, called the *inference horizon*. The native state contains all residual states, cache entries, and control fields required to determine the subsequent native execution. The native boundaries and the horizon are fixed before any answer is inspected. We keep the original weights, normalization, masks, cache rules, and deterministic decoding, including a fixed rule for equal token scores. Whenever the decoder emits a token, the emitted token must be verified under the original decoding rule over the full vocabulary, rather than fixed to a reference continuation.

A register is a fixed collection of continuous state coordinates used to store a persistent representation. The linear selection map P_t reads the register vector $w = P_t x \in \mathbb{R}^{d_{\mathrm{reg}}}$ from native state x; d_{reg} is the number of selected coordinates. Each atom has a scalar *fact score* whose sign indicates whether it is represented as established. For each ground atom $a \in \mathcal{A}$, the fact score is affine in w and identical at every step: $\varphi_a(w) = v_a^{\mathsf{T}} w + b_a^{\mathrm{fact}},\ k_a(x) = \mathbf{1}\{\varphi_a(P_t x) > 0\},$ and $\alpha_t(x) = (k(x), c_t^{\mathrm{ctrl}}(x))$. Here $c_t^{\mathrm{ctrl}}(x)$ is the extracted control state, and α_t is the *extractor*, mapping the native state to its fact bits and control state. The assignment of atoms to score rows and the thresholds are shared across inputs and steps. A representation may move to other registers only through a verified transfer of the same represented contents, called *register transport*. Initialization must extract Γ itself, not its closure. The allowed class of extractors and its size limits are fixed in advance; the interpretation may not invoke the remaining decoder or recover additional information directly from the prompt.

Let $\mathrm{dec}(x)$ denote the greedy-decoding output at the terminal state and $D_t(z) = \{x \in X_t : \alpha_t(x) = z\}$ the set of native states with the same abstract state z, called a *cell*. Exact executable abstraction means

$$F_t(D_t(z)) \subseteq D_{t+1}(H_t(z)),\ \mathrm{dec}(D_T(k, c_{\mathrm{ctrl}})) \subseteq \{A(k)\}, \tag{3}$$

where $A(k) = y_1$ if $k_q = 1$ and $A(k) = y_0$ otherwise, for two tokens $y_1 \neq y_0$ fixed in advance. These inclusions are equivalent to the transition and decision identities (Appendix B). The certificates below establish transition containment on the declared domains; correctness of the final decoded answer must be verified separately by checking the terminal readout.

Sharing the interpretation across steps is essential. Proposition 8 in Appendix B shows that stage-dependent linear extractors can extract the successive facts of a derivation from an unchanged state, while a lookup table provides the output. Any fixed deterministic fact readout gives unchanged bits on unchanged registers. Using the same interpretation across steps rules out this artifact; affinity separately restricts readout complexity. Neither condition makes the interpretation unique or replaces verification of the native writes.

3 Consistent variable binding from shared state reads

Addresses and binding registers. An atom has an *address*, a vector representation of its predicate and ordered arguments, with each label in a separate block of coordinates. Label vectors have unit norm, and distinct labels within a block have inner product at most $1 - \delta_{\mathrm{addr}}$ with $\delta_{\mathrm{addr}} > 0$. Padding labels give every atom the same number h_{addr} of blocks. The concatenated address $\mathrm{addr}(a)$ has norm $\sqrt{h_{\mathrm{addr}}}$, and distinct atoms have address inner product at most $h_{\mathrm{addr}} - \delta_{\mathrm{addr}}$.

For each variable V of the selected rule, a declared register has contents $\beta_V = B_V^{\mathrm{bind}} x$, representing the entity assigned to V; B_V^{bind} is the linear map that reads the contents of this register. Every occurrence of V in the native queries must read this register. Let θ specify the assignment of variables to constants, and let $e_{\theta(V)}$ be the unit label vector for the entity assigned to V. A *binding domain* is a set of native states representing the same substitution. On this domain we assume $\|\beta_V - e_{\theta(V)}\|_2 \leq \epsilon_{\mathrm{bind}}$ with $2\epsilon_{\mathrm{bind}} < \sqrt{2\delta_{\mathrm{addr}}}$, so the represented entity is unique. For the ith body atom B_i of the selected rule, the native query must decompose exactly into the predicate code, the constant codes, copies of the relevant β_V, and a remainder bounded by $\epsilon_{\mathrm{qry}}^{\mathrm{rem}}$; this decomposition is verified directly from the original read expressions rather than fitted from data. With at most r_{arg} argument positions, the total query error is at most

$$\epsilon_{\mathrm{qry}} = \sqrt{r_{\mathrm{arg}}}\, \epsilon_{\mathrm{bind}} + \epsilon_{\mathrm{qry}}^{\mathrm{rem}}. \tag{4}$$

Normalization and positional transformations belong to the remainder unless their effect is represented exactly. The invariant domains of Section 4 are where these bounds can be reused.

Theorem 1 (Shared binding and attention score separation)**.** *Consider an original attention read with $N_{\mathrm{slot}} \geq 1$ positions included by the attention mask, one explicitly present target position for each selected ground premise, and scores $\ell_{ij} = \kappa_{\mathrm{att}}\langle \mathrm{qry}_i(x), \mathrm{key}_j(x)\rangle + b_{ij}^{\mathrm{pos}}$, and $\kappa_{\mathrm{att}} > 0$. Here qry_i is the original query map for premise i, key_j is the original key map at position j, a_j is the atom represented at that position, and b_{ij}^{pos} is the positional score bias. Assume the identities for shared register reads above, $\|\mathrm{qry}_i - \mathrm{addr}(B_i\theta)\|_2 \leq \epsilon_{\mathrm{qry}}$, $\|\mathrm{key}_j - \mathrm{addr}(a_j)\|_2 \leq \epsilon_{\mathrm{key}}$, and $|b_{ij}^{\mathrm{pos}}| \leq \epsilon_{\mathrm{pos}}$. Distinct included positions have distinct addresses. Put*

$$\Delta_{\mathrm{bind}} = \kappa_{\mathrm{att}}\big[\delta_{\mathrm{addr}} - 2\sqrt{h_{\mathrm{addr}}}(\epsilon_{\mathrm{qry}} + \epsilon_{\mathrm{key}}) - 2\epsilon_{\mathrm{qry}}\epsilon_{\mathrm{key}}\big] - 2\epsilon_{\mathrm{pos}}. \tag{5}$$

If $\Delta_{\mathrm{bind}} > 0$, every premise read has a score gap of at least Δ_{bind} between its target and every competing position under the same substitution θ. If the projected values for this read are u_j, with target position j_*, and satisfy $\|u_j - u_{j'}\| \leq V_{\mathrm{att}}$ for all included positions j, j', while the target projected value satisfies $\|u_{j_*} - u_i^{\mathrm{prem}}\| \leq \epsilon_{\mathrm{val}}$, where u_i^{prem} denotes the declared numerical representation of premise i, then the read error is at most

$$\epsilon_{\mathrm{val}} + V_{\mathrm{att}}\frac{(N_{\mathrm{slot}} - 1)e^{-\Delta_{\mathrm{bind}}}}{1 + (N_{\mathrm{slot}} - 1)e^{-\Delta_{\mathrm{bind}}}}. \tag{6}$$

The proof combines the separation of concatenated codes, a perturbation bound on the actual dot products, and a softmax mass bound (Appendix C). Reading the same register through B_V^{bind} at every occurrence makes repeated variables denote one entity; the attention score separation then selects the ground atoms induced by that substitution. Neither property implies the other. All included positions stay in the softmax denominator, and an absent premise must have an actual encoded value or a verified default position.

Binding persistence has a direct weight test. For a residual branch $x \mapsto x + O\,u(x) + b^{\mathrm{res}}$, the identities $B_V^{\mathrm{bind}}O = 0$ and $B_V^{\mathrm{bind}}b^{\mathrm{res}} = 0$ preserve the register exactly, regardless of the normalization and nonlinearities within u. They must hold for every intervening branch, with a verified register transport whenever the represented contents move to different fields. A binding that is meant to change needs its own certified update.

4 Invariant neighborhoods from native operations

A correct write at one step does not establish correct execution over subsequent steps. Negative activation values, attention mass assigned to non-target positions (*attention leakage*), and normalization can perturb a retained fact score until a later read no longer recovers the intended fact. We prove containment in the required successor domain from quantities evaluated once on a fixed verification domain. For the structural results in this section, the ground rule g_t associated with block t is fixed by the declared native boundary and is independent of the current abstract state. The theorem therefore considers the subclass with common continuous coordinates and fixed blockwise rule selection; general state-dependent control is covered by the exact criterion of Theorem 9 in Appendix B.

Reference states and complete state domains. Write the complete continuous native state as $x \in \mathbb{R}^{d_{\mathrm{nat}}}$, including every mutable field and all fixed cache coordinates, and use a common register-selection map P. For each fact assignment k choose a reference state $\chi(k)$, which will serve as the center of its verification box, with

$$\varphi_a(P\chi(k)) = 2k_a - 1. \tag{7}$$

These reference states are used only for verification; the extractor never consults them. For a nonnegative radius vector r define the box $\mathcal{Q}_r(k) = \{x : |x - \chi(k)| \leq r\}$, with componentwise absolute values and inequalities. A zero radius fixes a coordinate exactly, for instance an immutable cache entry or an entity code. Every other native field must either be assigned a radius or be represented by a separately declared discrete type, such as a control-flow branch or cache layout, so that no state component is omitted. A fixed outer radius vector R^{box} is chosen first, and the corresponding boxes are called *outer boxes*. All original operations must be defined and continuously differentiable on neighborhoods of these boxes. Conditions that determine discrete execution branches are checked separately.

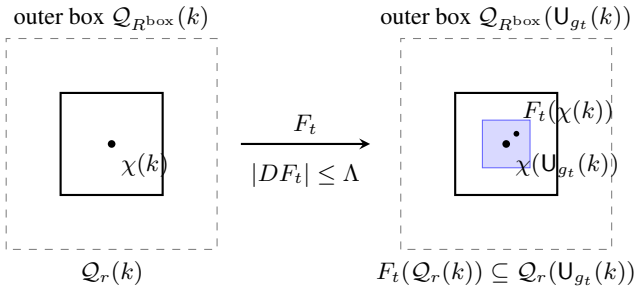


Figure 1: The invariant-domain argument. The block F_t moves the center $\chi(k)$ to within $\varepsilon^{\mathrm{anc}}$ of the successor center, while the Jacobian bound maps a source deviation bounded by r to a deviation bounded by Λr. Hence the image lies in the successor inner box whenever $\varepsilon^{\mathrm{anc}} + \Lambda r \le r$. All bounds are evaluated on the outer boxes, which never change.

Theorem 2 (Reusable native certificate)**.** *Suppose the original blocks F_t satisfy, for every fact code k and every $x \in \mathcal{Q}_{R^{\mathrm{box}}}(k)$, $|F_t(\chi(k)) - \chi(\mathsf{U}_{g_t}(k))| \le \varepsilon^{\mathrm{anc}}$, $|DF_t(x)| \le \Lambda$, $\Lambda \ge 0$, where $\varepsilon^{\mathrm{anc}} \ge 0$ bounds the coordinatewise error at each reference state relative to the required successor center, and $|DF_t|$ is the entrywise absolute Jacobian; both bounds include every branch of the complete native update. If a vector $0 \le r \le R^{\mathrm{box}}$ satisfies*

$$\varepsilon^{\mathrm{anc}} + \Lambda r \le r,\ |v_a^{\mathsf{T}} P| r \le 1 - \gamma_{\mathrm{fact}}\ (a \in \mathcal{A}),\ \gamma_{\mathrm{fact}} > 0, \tag{8}$$

then $F_t(\mathcal{Q}_r(k)) \subseteq \mathcal{Q}_r(\mathsf{U}_{g_t}(k))$. Every run initialized in these boxes follows the scheduled Horn updates with fact margin at least γ_{fact}. Whenever the binding and read hypotheses hold uniformly on the outer boxes, they hold at every later step without an additional successor-domain assumption.

Proof. The segment from $\chi(k)$ to x stays in its convex outer box. Integrating the Jacobian along that segment gives $|F_t(x) - F_t(\chi(k))| \le \Lambda |x - \chi(k)|$. The error bound at the reference state and (8) place $F_t(x)$ in the required successor box. Equation (7) and the bound on fact-score variation preserve the sign of every fact score throughout each box. Induction applies to the complete native state, not only to the fact signs. □

The inequality $\varepsilon^{\mathrm{anc}} + \Lambda r \le r$ ensures that the reference-state error together with the maximum propagated deviation remains within the same radius r (Figure 1). Strict contraction is one sufficient way to absorb a nonzero reference-state error, not a necessary condition for an invariant neighborhood: an exact identity update preserves its box without contraction. What is required here is the displayed componentwise inequality; without it or another invariant-domain argument, repeated errors need not remain bounded (Section 4.2). The estimate itself is a standard comparison argument; its role here is to turn weight-derived bounds into a reusable logical domain. Section 4.1 reduces the error condition at reference states to finitely many checks, and Appendix F.1 checks all conditions jointly.

After checking that zero-radius coordinates are preserved exactly, let $\mathcal{J}$ index the remaining coordinates. If the principal submatrix $\Lambda_{\mathcal{J}\mathcal{J}}$ has spectral radius less than one, the least admissible radius on those coordinates is $r_{*,\mathcal{J}} = (I_{\mathcal{J}} - \Lambda_{\mathcal{J}\mathcal{J}})^{-1} \varepsilon^{\mathrm{anc}}_{\mathcal{J}} = \sum_{j=0}^{\infty} (\Lambda_{\mathcal{J}\mathcal{J}})^j \varepsilon^{\mathrm{anc}}_{\mathcal{J}}$, extended by zero on the fixed coordinates (Proposition 10). It still has to satisfy $r_* \le R^{\mathrm{box}}$ and the fact-margin inequality, and it is least for the supplied comparison data rather than for every native invariant set. Blockwise pairs $(\varepsilon_t^{\mathrm{anc}}, \Lambda_t)$ can be checked as linear inequalities with one common r, and a finite horizon can be certified without contraction by propagating $r_{t+1} = \varepsilon_t^{\mathrm{anc}} + \Lambda_t r_t$ (Appendix D).

4.1 Local verification without exhaustive enumeration of fact assignments

Suppose coordinate j of block t has a local reference expression $\psi_{t,j}$ that depends only on the fact bits indexed by a specified set $S_{t,j} \subseteq \mathcal{A}$, with a certified bound on its difference from the original block output:

$$|F_{t,j}(\chi(k)) - \psi_{t,j}(k_{S_{t,j}})| \le \zeta_{t,j},\ [\chi(\mathsf{U}_{g_t}(k))]_j = \omega_{t,j}(k_{S_{t,j}}), \tag{9}$$

where $\omega_{t,j}$ is the target center coordinate. Then

$$\varepsilon_j^{\mathrm{anc}} = \max_t \left\{ \zeta_{t,j} + \max_{b \in \{0,1\}^{|S_{t,j}|}} |\psi_{t,j}(b) - \omega_{t,j}(b)| \right\} \tag{10}$$

is a valid error bound at the reference states. The bound $\zeta_{t,j}$ must hold uniformly over all fact assignments not represented explicitly in $S_{t,j}$ and over every relevant native branch. For an attention block, the target value may be used in the local reference expression only if $\zeta_{t,j}$ also bounds the contribution of all non-target positions under the original finite-temperature softmax. The reference expression is used only for the bound; the native computation still evaluates the full softmax.

Proposition 3 (Number of local reference evaluations)**.** *A* block template *specifies the same native operations and parameters wherever that block is repeated. Index the distinct block templates by* $\ell = 1, \ldots, N_{\mathrm{typ}}$*. The local calculation at reference states requires at most* $\sum_{\ell=1}^{N_{\mathrm{typ}}} \sum_j 2^{|S_{\ell,j}|}$ *local evaluations. If every set has size at most* d_{dep}*, this is at most* $N_{\mathrm{typ}} d_{\mathrm{nat}} 2^{d_{\mathrm{dep}}}$*. A head update depends on its previous bit and on at most* $m_{\max}$ *body bits, whereas a retained fact depends only on its own bit. Hence, for a representation with this locality,* $d_{\mathrm{dep}} \le m_{\max} + 1$*, independently of the total number of fact assignments.*

The count concerns evaluations at reference states only. The cost of the programs specifying the centers and local expressions must also be included, so an exponential table of centers does not become compact through this count, and the nonlinear remainder and Jacobian bounds have their own cost. Normalization preserves this locality at the reference states when $\chi(k) = x^{\mathrm{cen}} + E^{\mathrm{code}}(2k - \mathbf{1})$ and the projected columns of E^{code} are mutually orthogonal and orthogonal to the projected x^{cen}. Under these conditions, every reference state has the same normalization denominator (Appendix D). Away from the reference states, variation of the normalization factor is accounted for in the Jacobian and remainder bounds.

4.2 A SUFFICIENT NATIVE STRUCTURE FOR PERSISTENT FACTS

We call an operation that restores a perturbed signed fact value toward its assigned code value a *refresh operation*. Such an operation can induce contraction while retaining the original activation function. For the SiLU activation $f_{\mathrm{S}}(u) = u/(1 + e^{-u})$ define $C_{\kappa_{\mathrm{act}}}(u) = \frac{f_{\mathrm{S}}(\kappa_{\mathrm{act}}(2u+1)) - f_{\mathrm{S}}(\kappa_{\mathrm{act}}(2u-1))}{\kappa_{\mathrm{act}}} - 1$. When $2\kappa_{\mathrm{act}}\tau_{\mathrm{act}} \ge 1$, for $|u| \ge 1/2 + \tau_{\mathrm{act}}$ this expression lies within $4\tau_{\mathrm{act}} e^{-2\kappa_{\mathrm{act}}\tau_{\mathrm{act}}}$ of $\mathrm{sign}(u)$, so it maps a perturbed signed value close to its target sign. For a rule with m premises, let s_{h_g} and p_i denote the signed head and premise values, respectively. With signed inputs, the affine test $m s_{\mathsf{h}_g} + \sum_{i=1}^m p_i + 1$ is positive exactly when the old head is true or all m premises are true. Preserving an already established head is essential, because overwriting an established head with the current body value would erase a valid earlier deduction whenever the current rule body is false.

In this structure, *fact coordinates* store signed values representing the fact bits, while *scratch coordinates* temporarily store intermediate values. Suppose a block's attention leaves the fact coordinates unchanged and writes the premise reads into the scratch coordinates, and let $\widetilde{s}, \widetilde{p}$ be their actual normalized values. The native fact update has the form $s'_a = s_a - \widetilde{s}_a + C_{\kappa_{\mathrm{act}}}(u_a) + e_a^{\mathrm{wr}}$, $u_{\mathsf{h}_g} = m\widetilde{s}_{\mathsf{h}_g} + \sum_{i=1}^m \widetilde{p}_i + 1 + \xi^{\mathrm{ref}}$, with $u_a = \widetilde{s}_a$ for non-head facts. Scratch coordinates receive their negative normalized values, so only normalization error and the bounded remaining write are left. All remaining native contributions are included in e^{wr} or in the preactivation remainder ξ^{ref}. *Protected coordinates* are state coordinates required to remain unchanged; their preservation must follow from verified identities for the native operations.

Proposition 4 (Uniform preservation by smooth refresh)**.** *Let every mutable coordinate be one of these fact or scratch coordinates. On source boxes of radius* $0 < r_{\mathrm{uni}} < 1$*, suppose the normalized-coordinate error is at most* δ_{norm}*, a scratch premise before normalization differs from its signed fact by at most* $r_{\mathrm{uni}} + \epsilon_{\mathrm{ret}}$*,* $|\xi^{\mathrm{ref}}| \le \epsilon_{\mathrm{gate}}$*, and* $\|e^{\mathrm{wr}}\|_\infty \le \epsilon_{\mathrm{write}}$*. Suppose all nonempty bodies have size at most* $m_{\max}$*. If* $0 < \tau_{\mathrm{act}} < 1/2$*,* $2\kappa_{\mathrm{act}}\tau_{\mathrm{act}} \ge 1$*, and*

$$\begin{aligned} r_{\mathrm{uni}} + \delta_{\mathrm{norm}} &\le 1/2 - \tau_{\mathrm{act}}, \\ 2m_{\max}(r_{\mathrm{uni}} + \delta_{\mathrm{norm}}) + m_{\max}\epsilon_{\mathrm{ret}} + \epsilon_{\mathrm{gate}} &\le 1/2 - \tau_{\mathrm{act}}, \\ \delta_{\mathrm{norm}} + 4\tau_{\mathrm{act}} e^{-2\kappa_{\mathrm{act}}\tau_{\mathrm{act}}} + \epsilon_{\mathrm{write}} &\le r_{\mathrm{uni}}, \end{aligned} \tag{11}$$

then every original block maps its source box into the correct Horn-successor box. The same conditions certify arbitrarily many repetitions of the block templates, or any fixed finite prefix realized by a decoder. An empty-body head uses $u_{\mathsf{h}_g} = 1$ and needs no premise test.

The first two inequalities keep every refresh input at least τ_{act} away from the transition interval $[-1/2, 1/2]$; the third absorbs normalization and all remaining writes, including negative activation values. The proof is in Appendix D.6. Verification does not modify the gain or cancellation weights; the required affine structure must already be present in the original block.

5 Internally generated substitutions and shared rule templates

We share parameters across substitutions, generating candidate substitutions internally and checking them against the current facts. The weights encode the rule program without an expanded substitution table.

Let $R = |\mathcal{R}| \geq 1$, let $c \geq 1$ be the number of constants, and write $n = |\mathcal{A}|$. Rule ρ has m_ρ premises and v_ρ distinct variables, including variables occurring only in its head. Put $m = \max_\rho m_\rho$, $v = \max_\rho v_\rho$, and let $a_{\max}$ be the maximum predicate arity. A macrostep consists of seven native blocks, detailed in Appendix E, and applies one rule instance; T_{mac} macrosteps comprise $7T_{\mathrm{mac}}$ native blocks. Its abstract state is $z = (k, \rho, d_1, \ldots, d_v)$, where $d_j \in \{0, \ldots, c-1\}$ and $d_j = 0$ for $j > v_\rho$. The first v_ρ digits supply the current substitution θ. After applying $\mathsf{U}_{\rho\theta}$, the network increments these digits in base c; overflow resets them and advances ρ cyclically. Denote this complete abstract update by H. Neither θ nor the next control state is supplied externally.

The live residual stores fact signs, a one-hot rule phase, shared one-hot entity digits, and temporary bits. For each premise–atom pair, an attention head tests agreement with the current binding and reads the atom's live fact coordinate; a second included position has fixed zero key and value. Summing the heads retrieves the premise under that substitution, including previously derived facts. SiLU blocks then perform the guarded write and internal counter update.

Theorem 5 (Shared templates with internal substitution enumeration)**.** *For every such program, seven fixed residual–attention–SiLU templates implement H using finite-temperature softmax and RMS normalization. Set $B = R + v + m + a_{\max} + 3$, $r_{\mathrm{sh}} = \frac{1}{64(B+1)}$. There are complete-state boxes $\mathcal{Q}_{r_{\mathrm{sh}}}(z)$, with all temporary bits initially false, such that their native composition F satisfies*

$$F(\mathcal{Q}_{r_{\mathrm{sh}}}(z)) \subseteq \mathcal{Q}_{r_{\mathrm{sh}}}(H(z)). \tag{12}$$

Every intermediate block also maps its declared source box into the correct successor box, with fact margin at least $1 - r_{\mathrm{sh}}$. The same parameters and radii work for every finite repetition. Initialization uses the input facts, the first rule, and zero entity digits. If $G_{\mathrm{sub}} = \sum_{\rho=1}^{R} c^{v_\rho}$, $T_{\mathrm{mac}} \geq G_{\mathrm{sub}}(n - |\Gamma|)$, then after T_{mac} macrosteps the facts equal $\mathrm{Cl}_{\mathcal{R}}(\Gamma)$, and a fixed full-vocabulary readout answers any fixed ground query with margin at least one. The residual width d, number of read heads H_{att}, and nonzero parameter count W obey $d = O(R(n+m+1)+cv+v)$, $H_{\mathrm{att}} \leq \max\{1, Rmn\}$, and $W = O(Rn(m+1)(a_{\max}+1) + cv + v^2 + Rv)$. Sparse execution costs $O(T_{\mathrm{mac}}W)$ real-arithmetic operations. Structure and analytic verification cost $O(W + d^2 + H_{\mathrm{att}}d + Vd)$ for vocabulary size V, independently of T_{mac} and G_{sub} apart from their program-size parameters.

Appendix E specifies every stage, the attention projections, the SiLU matrices, and the normalization and score bounds. The proof controls complete intermediate states by direct box inclusion; it does not assume the fixed rule selection used in Theorem 2. An internal counter visits every labeled rule–substitution pair once per G_{sub} macrosteps. Parameter sharing therefore avoids an explicit substitution table while preserving its enumeration cost. The weights may depend on the program and constant domain, but not on the initial facts or number of scans. The atom vocabulary is explicit, and the guarantee concerns the declared internal-state input and real arithmetic.

For example, the rules $E(x, y) \to P(x, y)$ and $P(x, y) \wedge E(y, z) \to P(x, z)$ have $n = 2c^2$ and $G_{\mathrm{sub}} = c^2 + c^3$. The construction uses $O(c^2)$ mutable coordinates and nonzero parameters, and subsequent reads use the newly established P facts. Appendix E.8 gives the count. This construction gives a storage upper bound for the stated architecture. Appendix E.9 uses a matched ReLU construction to isolate the approximation error introduced by SiLU.

6 LOGICAL AND CAUSAL CONSEQUENCES

Theorem 6 (Soundness and bounded completeness)**.** *Assume correct initialization and cell inclusions corresponding to Horn-rule updates, obtained from Theorem 2 or another valid native certificate. Then every run that remains within the certified domains satisfies*

$$\Gamma \subseteq \mathsf{K}(k_t) \subseteq \mathrm{Cl}_{\mathcal{R}}(\Gamma),\ \alpha_t(x_t) = H_{t-1} \cdots H_0(\alpha_0(x_0)). \tag{13}$$

Every fact added for the first time has a ground-rule proof whose premises were available earlier. If every ground rule is selected at least once in every $B_{\mathrm{cov}} \geq 1$ consecutive steps, then $T \geq B_{\mathrm{cov}}(n_{\mathrm{at}} - |\Gamma|) \implies \mathsf{K}(k_T) = \mathrm{Cl}_{\mathcal{R}}(\Gamma)$. If the terminal decision condition in (3) *also holds, the original greedy output answers the declared entailment task correctly.*

Soundness follows by induction over legally added new facts. Within each coverage window, monotonicity keeps every body that was enabled at the start of the window enabled until its rule is tested, so each window performs one round of simultaneous consequences, and at most $n_{\mathrm{at}} - |\Gamma|$ rounds reach the least closure. The full proof, the empty-program case, and early positive termination are in Appendix G. A legal but idle execution shows why soundness alone cannot justify a negative answer.

Original output. Let $\boldsymbol{\pi}(x)$ be the terminal vocabulary distribution. A sufficient certificate for the greedy decision is $\log \boldsymbol{\pi}_{A(k)}(x) - \log \boldsymbol{\pi}_y(x) \geq \gamma_{\mathrm{out}} > 0$ $(y \neq A(k))$, where the inequality is checked against every competing vocabulary token. Correct fact transitions do not prove this readout condition, and an output margin does not imply equal probability vectors within a logical cell.

6.1 REALIZABLE LOGICAL EDITS

A *complete logical edit* υ specifies the entire retained abstract state, including control. A *partial edit* leaves some components of that state unspecified. A permitted native operation I implements it at $x \in X_t$ if

$$I(x) \in X_t,\ \alpha_t(I(x)) = \upsilon(\alpha_t(x)), \tag{14}$$

and its continuation stays in the certified domains. The state immediately after the intervention is its *endpoint*. Let $J_t(z) \in \{0, 1\}$ denote the indicator that the remaining abstract execution from z terminates with y_1, so $J_T(k, c_{\mathrm{ctrl}}) = k_q$.

Theorem 7 (Implementation-independent decision effects)**.** *Under* (3)*, every permitted implementation of the same complete edit has the same terminal decision, and*

$$\begin{aligned} &\mathbf{1}\{\mathrm{dec}(F_{T-1} \cdots F_t(I(x))) = y_1\} - \mathbf{1}\{\mathrm{dec}(F_{T-1} \cdots F_t(x)) = y_1\} \\ &\quad = J_t(\upsilon(\alpha_t(x))) - J_t(\alpha_t(x)). \end{aligned} \tag{15}$$

For a partial edit with at least one permitted native implementation, the decision effect is uniquely determined if and only if J_t is constant on the abstract states extracted from all permitted intervention endpoints.

The result follows directly from the abstraction. A nonzero effect shows that the edit is causally relevant to the terminal decision on the intervention domain. Deleting one premise may leave another derivation intact, and deleting it after its conclusion has been stored differs from removing it at initialization. Appendix H characterizes which subsets of the initial facts still entail the query and gives an affine-control feasibility test; neither determines changes in confidence without additional real-valued output conditions.

The program predicts which initial deletions can change the answer. Let $\mathcal{U}_q(\Gamma)$ be the family of inclusion-minimal subsets of Γ that entail q, called *minimal premise supports*. Under complete deduction from both initializations, a deletion $\Delta_\Gamma \subseteq \Gamma$ changes an originally positive decision exactly when $S \cap \Delta_\Gamma \neq \varnothing$ for every $S \in \mathcal{U}_q(\Gamma)$. A single derivation provides only one sufficient support, not the full family of minimal supports. For propositional atoms with rules $\mathsf{P} \to \mathsf{Q}$ and $\mathsf{R} \to \mathsf{Q}$ and both inputs present, either single deletion preserves the answer and deleting both changes it. An empty support makes every input deletion ineffective. Proposition 16 proves the characterization and shows that finding a deletion within a cardinality bound is NP-complete.

7 EXPERIMENTS AND DISCUSSION

Logical execution and perturbations. We evaluate the seven-stage update equations for the reachability program in Appendix E.8 using double precision. For $c = 2$, SiLU and matched ReLU each achieved 100% logical-state agreement on all $3{,}072$ reference states and $30{,}720$ states sampled uniformly from their radius-r_{sh} boxes across ten seeds. For $c \in \{4, 8\}$, chains, cycles, disconnected graphs, and graphs with multiple paths yielded 80 runs per activation across ten seeds, each lasting 10^5 macrosteps, with no fact or controller errors. The largest recorded macrostep-boundary error for SiLU was 1.18×10^{-6}, below $r_{\mathrm{sh}} = 1/832$. Replacing correction by the identity caused failure at the first macrostep in every long run. This ablation changes Boolean gate realization as well as error correction, so it does not isolate accumulated numerical drift.

Storage and substitution enumeration. For $c \in \{2, 4, 8, 16, 32\}$, shared metadata and explicit substitution-table storage had log–log slopes of 1.975 and 2.868, respectively. At $c = 32$, their sizes were $254{,}032$ and $1{,}351{,}680$ bytes. Symbolic evaluators generating substitutions or reading the explicit table produced identical fact sets after one complete scan, with timing slopes of 2.884 and 2.856 across ten seeds and ten timing repetitions. These measurements illustrate reduced representation storage without removing enumeration cost; they do not measure total model storage, native Transformer runtime, or complete certificate verification. The execution results are consistent with the invariant-neighborhood analysis, but finite numerical tests do not establish a floating-point certificate for arbitrary horizons.

Logical-state interventions. For $c \in \{4, 8\}$ and ten seeds, we tested four logical edits with three continuous implementations each. All 240 edited endpoints lay within their prescribed boxes, every subsequent macrostep matched the symbolic continuation, and implementations of the same edit produced identical terminal decisions and decision effects. Initial deletions changed positive answers to negative exactly when no premise support remained. After $P(0, 1)$ had been stored, deleting $E(0, 1)$ still allowed $P(0, 2)$ to be derived, unlike deletion at initialization. These results illustrate how intervention effects depend on both alternative supports and retained facts, consistently with Theorem 7 and Proposition 16. Details appear in Appendix K.

Hidden-state interventions in a pretrained model. We evaluate learned state edits in frozen 4-bit Mistral-Small-24B-Instruct-2501. A predictor uses the current layer-17 state and edit type, without matched test counterfactual states. Unedited test accuracy was 99.14%. Among 318 conditions whose original and counterfactual inputs were both answered correctly, three endpoint variants jointly matched symbolic effects in $233/240$ nonzero-effect conditions (97.08%) and $50/78$ zero-effect conditions (64.10%); their greedy decisions always agreed. Agreement across endpoints therefore did not ensure preservation of answers supported by alternative derivations. Only one of ten seeds met all development criteria, so the pooled results remain exploratory rather than a checkpoint certificate. Appendix L gives the protocol and detailed results.

8 VERIFICATION SCOPE AND CONCLUSION

A certificate is a universal statement over declared state domains, so finitely many successful prompts cannot establish it. Local verification reduces reference-state checks under the stated dependency structure but does not eliminate the need to bound the original nonlinear operations or verify exact structural identities.

The shared-template construction combines internal substitution enumeration and derived-fact reuse under finite-temperature softmax, RMS normalization, and SiLU. Its explicit resource bounds separate stored parameters from enumeration time, with a perturbation radius independent of the number of scans. The twelve-atom instance gives a complementary concrete realization of the binding, invariance, and readout conditions. Applying these guarantees to a pretrained checkpoint would require verifying its representations, structural identities, and numerical bounds; no such certification is claimed.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of this paper, to support the retrieval and discovery of related work, and to assist in checking and revising portions of the mathematical arguments and proofs, and to support the verification of portions of the experimental code. All AI-assisted writing was independently verified by the authors against the stated assumptions, derivations, and reported results, and all references identified with AI assistance were checked against the original publications. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Complete proofs of the theoretical results are provided in Appendices A–H. Appendices I–L describe the experimental protocols, implementation details, numerical settings, random seeds, evaluation criteria, and statistical procedures, together with detailed results. The supplementary material contains the source code used to reproduce the experiments and numerical evaluations reported in the paper.

REFERENCES

Martín Abadi and Leslie Lamport. The existence of refinement mappings. *Theoretical Computer Science*, 82(2):253–284, 1991. doi: 10.1016/0304-3975(91)90224-P. URL https://www.sciencedirect.com/science/article/pii/030439759190224P.

Artur S. Avila Garcez and Gerson Zaverucha. The connectionist inductive learning and logic programming system. *Applied Intelligence*, 11(1):59–77, July 1999. ISSN 0924-669X. doi: 10.1023/A:1008328630915. URL https://doi.org/10.1023/A:1008328630915.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 9574–9586. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4f5c422f4d49a5a807eda27434231040-Paper.pdf.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pp. 2733–2743, 2019.

Richard M. Karp. *Reducibility among Combinatorial Problems*, pp. 85–103. Springer US, Boston, MA, 1972. ISBN 978-1-4684-2001-2. doi: 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. Tracr: compiled transformers as a laboratory for interpretability. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Leyan Pan, Vijay Ganesh, Jacob Abernethy, Chris Esposo, and Wenke Lee. Can transformers reason logically? a study in SAT solving. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47632–47671. PMLR, 2025. URL https://proceedings.mlr.press/v267/pan25d.html.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3363–3377, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295/.

Eigil F. Rischel and Sebastian Weichwald. Compositional abstraction error and a category of causal models. In Cassio de Campos and Marloes H. Maathuis (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1013–1023. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/rischel21a.html.

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. PRover: Proof generation for interpretable reasoning over rules. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 122–136, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.9. URL https://aclanthology.org/2020.emnlp-main.9/.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. The non-linear representation dilemma: Is causal abstraction enough for mechanistic interpretability? *ArXiv*, abs/2507.08802, 2025. URL https://api.semanticscholar.org/CorpusID:280298297.

M. H. Van Emden and R. A. Kowalski. The semantics of predicate logic as a programming language. *J. ACM*, 23(4):733–742, October 1976. ISSN 0004-5411. doi: 10.1145/321978.321991. URL https://doi.org/10.1145/321978.321991.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Yiwei Wu, Atticus Geiger, and Raphaël Millière. How do transformers learn variable binding in symbolic programs? In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van Den Broeck. On the paradox of learning to reason from data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/375. URL https://doi.org/10.24963/ijcai.2023/375.

A Formal scope, certificate data, and notation

A.1 The specification carried by a certificate

A claim in this paper refers to one fixed native executable, one bounded inference protocol, one logical specification, and one reported domain. The weights, normalization constants, tokenizer, positional transformations, cache policy, rule for equal token scores, and arithmetic interpretation are part of that executable. A real-arithmetic certificate and a certificate for its floating-point implementation are different objects. At every boundary, the state contains every register required to determine its successor. Temporary variables used to accumulate attention values can be omitted only at boundaries where their contents are no longer needed. A separately verified conversion to an alternative state representation may be used through its exact state-transition interface; the present proofs do not require one.

The logical input is a finite relational signature, a finite nonempty constant set, a definite program, and a ground query. Grounding is over that constant set. The mathematical results do not require every head variable to occur in the body, because even variables occurring only in the head have finitely many ground assignments. Such assignments must be recorded. Distinct constant symbols are

interpreted as the distinct constants of the declared Herbrand domain; no built-in equality reasoning is assumed. An equality predicate, if used, requires its own specified interpretation and rules. Grounding can be expensive and is not free preprocessing when an execution-cost claim is made.

A certificate contains: the prompt-to-fact interpretation; the source domains and initialization map; the grouping of original operations between the specified boundaries; register reads and transports; the shared fact-score matrix and thresholds; controller extraction and coding; the selector and controller program; every variable substitution used to instantiate a rule; cell coverage and native successor evidence; and the unchanged terminal-output evidence. A causal certificate additionally contains the permitted low-level intervention, its complete abstract edit, and its endpoint and continuation-domain checks. All of these are mathematical data, not observations that the paper asserts have been measured on a pretrained model.

Formally, $\sigma_t : \{0,1\}^{n_{\mathrm{at}}} \times \mathcal{C} \to \mathcal{G} \cup \{\circ\}$ and $\tau_t : \{0,1\}^{n_{\mathrm{at}}} \times \mathcal{C} \to \mathcal{C}$ are supplied total functions. Their formulas determine H_t on the full finite abstract space, although agreement with the native execution is required only on nonempty cells and the claimed extensions. Initial fact annotations are effectively supplied whenever a decision procedure is asserted.

The main native-operation theorem uses affine fact scores but does not assume that all latent semantics are linear. It addresses the supplied representation class. An alternative expression class can be analyzed through Theorem 9, provided its program and verification costs are explicit. No claim is made that a bounded affine code is complete for all ways in which a network might implement a logical algorithm. Likewise, a Horn program does not represent every possible account of reasoning.

A.2 Persistent registers and legitimate movement of a representation

A fact-register assignment identifies physical fields in the native state, rather than supplying a new interpretation of the whole state at each time. At boundaries where the fields persist, P_t and P_{t+1} select corresponding fields. Their change is the actual native write to those fields. If a block only transports contents through a known map T_t^{reg}, its interpretation must satisfy a checked identity

$$\varphi^{\mathrm{dst}}(T_t^{\mathrm{reg}}(w)) = \varphi^{\mathrm{src}}(w) \tag{16}$$

on the declared transport domain. In the common-coordinate form of the main text, both sides use the same fact-score map after transporting back to the declared common register coordinates. The native register transport itself, the coordinate conversions, and their costs are supplied. A change in a fact bit cannot be attributed solely to changing the atom-to-row assignment or its threshold.

These requirements distinguish a representation move from an inference step. They do not require a fact representation to remain in the same neuron throughout execution. A cache insertion, a new token's residual initialization, or a write into a different buffer can be included, but it has to be part of the actual F_t and of the checked interpretation identity. A proposed transport defined by running the remaining model to determine the desired future fact is not a register transport in this specification. Without a verified restricted transport, the extraction requirement is not established.

A.3 Decisions, probabilities, and logical truth

The native decoder maps its terminal state to a full probability vector, and the specified greedy rule then picks one vocabulary token. The high-level answer map uses only k_q. Equality of these decisions is weaker than factorization of the probability vector or of a real-valued contrast. Consequently Theorem 7 identifies decision changes only. A single logical state may contain native states with different confidence, normalization statistics, or probability mass on tokens not selected by greedy decoding. These differences are not suppressed by an unproved real-valued readout factorization. Quantitative response guarantees require separate abstraction and error conditions (Rischel & Weichwald, 2021).

A correct decision under definite-program semantics means membership in the least closure of the supplied facts. It does not assert the truth of the supplied facts in the world. An input-level annotation that maps a natural-language sentence to a predicate can itself be wrong; none of the subsequent algebra repairs that error. The interpretation map is therefore reported separately from the native execution proof. The same qualification applies to negative answers: a query absent from the saturated closure is not a proved negative literal unless a different logical specification explicitly says so.

A.4 NOTATION

The following table lists the principal symbols. Superscripts and subscripts that distinguish native numerical quantities are part of their notation.

Symbol	Meaning
$\mathcal{A}, n_{\mathrm{at}}, q$	Ground atoms, their number, and the queried atom.
$\mathcal{R}, \mathcal{G}, g, \theta_g$	Rule schemata, their ground instances, one instance, and its common substitution.
$\mathsf{B}_g, \mathsf{h}_g, \Gamma$	Ground body, ground head, and initial facts.
$\mathrm{Cl}_{\mathcal{R}}, \mathsf{U}_g$	Least deductive closure and one scheduled rule test.
$k, \mathsf{K}(k)$	Fact bits and the represented set of established facts.
$\mathcal{C}, c_{\mathrm{ctrl}}, z = (k, c_{\mathrm{ctrl}})$	Finite control set, its current value, and the complete abstract state.
σ_t, τ_t, H_t	Rule selector, controller update, and Horn-form transition.
X_t, x, F_t, T	Native domain, state, native block, and number of blocks.
P_t, w, d_{reg}	Register-selection map, selected state vector, and number of selected coordinates.
$\varphi_a, v_a, b_a^{\mathrm{fact}}$	Fixed score of atom a, its row vector, and its offset.
$\alpha_t, D_t(z)$	Complete extractor and the corresponding set of native states.
$\boldsymbol{\pi}, \mathrm{dec}, A$	Native probabilities, greedy decision, and logical answer token.
$y_1, y_0, \gamma_{\mathrm{out}}$	Positive and negative answer tokens, and full-vocabulary margin.
$N_{\mathrm{slot}}, \ell_{ij}, u_j$	Included attention positions, attention scores, and projected values.
$\Delta_{\mathrm{bind}}, V_{\mathrm{att}}$	Target score gap and projected-value diameter bound.
$\gamma_{\mathrm{fact}}, B_{\mathrm{cov}}$	Fact margin and rule-coverage window length.
I, υ, J_t	Low-level edit, complete abstract edit, and decision indicator for the remaining abstract execution.
$L_{\mathrm{edit}}, h_{\mathrm{edit}}$	Intervention matrix and control vector.
$M_{\mathrm{code}}, b^{\mathrm{code}}, b^{\mathrm{tgt}}$	Stacked score matrix, offsets, and target bit code.
$\Delta_\Gamma, \mathcal{U}_q(\Gamma)$	Removed input facts and inclusion-minimal query supports.

The selector's idle label $\circ$ is not a ground rule. An empty sequence of native or abstract transitions means the identity map.

A.5 ADDITIONAL STRUCTURAL NOTATION

Symbol	Meaning
$h_{\mathrm{addr}}, \delta_{\mathrm{addr}}$	Address-block count and within-block label separation.
$\beta_V, B_V^{\mathrm{bind}}$	Current entity register and its shared native read map.
$\mathrm{qry}_i, \mathrm{key}_j$	Original query and key maps in the stated address coordinates.
$\epsilon_{\mathrm{qry}}, \epsilon_{\mathrm{key}}, \epsilon_{\mathrm{pos}}$	Query, key, and positional-score error bounds.
$\kappa_{\mathrm{att}}, \Delta_{\mathrm{bind}}$	Score scale and derived attention score gap under the shared substitution.
$\chi(k), d_{\mathrm{nat}}, P$	Reference state in the complete native domain, native coordinate count, and common register read.
$\mathcal{Q}_r(k), R^{\mathrm{box}}, r$	Box centered at a reference state, fixed outer radii, and candidate invariant radii.
$\varepsilon^{\mathrm{anc}}, \Lambda$	Error at reference states for the complete native update and nonnegative Jacobian comparison matrix.
$\mathcal{J}, r_*$	Non-fixed coordinate indices and least comparison radius.
$S_{t,j}, \psi_{t,j}, \omega_{t,j}, \zeta_{t,j}$	Local dependency set, reference expression, target expression, and native remainder.
$N_{\mathrm{typ}}, d_{\mathrm{dep}}$	Number of block templates and maximum local dependency size.
$x^{\mathrm{cen}}, E^{\mathrm{code}}, L^{\mathrm{norm}}$	Affine center offset, code directions, and effective normalizer input map.
$C_{\mathrm{norm}}, \lambda, C_{\mathrm{act}}$	Fixed normalization scale, actual variable scale, and native SiLU refresh expression.
$e_{\mathrm{uni}}, L_{\mathrm{uni}}, r_{\mathrm{uni}}$	Scalar bounds on reference-state error, the Lipschitz constant, and the radius in the norm-based certificate.

In the shared construction, R counts rules, c counts constants, $N_{\mathrm{prem}} = \sum_\rho m_\rho$ counts premise occurrences, D is the mutable dimension, $d = D + 1$ is the residual width, and T_{mac} counts

seven-block macrosteps. The digit vector is $\mathbf{d}$, and c_{ctrl} denotes an abstract control state. Protected coordinates have radius zero, not an omitted error budget.

B Extraction restrictions and the exact characterization

Proposition 8 (Step-dependent linearity does not exclude a spurious logical trace)**.** *For any finite collection of fact inputs, there is a stationary native state system with a correct lookup readout and stage-dependent linear fact-score maps that reproduce a common Horn evaluation schedule exactly, including its initial and terminal facts. If the same fact-score map instead reads unchanged persistent registers at consecutive boundaries, no extracted fact can change.*

B.1 Proof of Proposition 8

Proof. Let the input collection be indexed by $p \in \{1, \dots, N_{\mathrm{inp}}\}$. Fix the same atom set, rule program, and query for all inputs in the collection. Choose any fixed schedule of ground-rule tests long enough to reach closure on all these fact inputs, for example repeated complete rule scans. Let $k_t^{(p)}$ be the fact bits after t tests, starting from the actual input $\Gamma(p)$.

The native continuous state for input p is $x = e_p \in \mathbb{R}^{N_{\mathrm{inp}}}$, and every native transition is the identity on these states. Define a matrix $V_t^{\mathrm{dec}} \in \mathbb{R}^{n_{\mathrm{at}} \times N_{\mathrm{inp}}}$ by

$$[V_t^{\mathrm{dec}}]_{a,p} = 2[k_t^{(p)}]_a - 1. \tag{17}$$

Reading the signs of $V_t^{\mathrm{dec}} x$ gives precisely $k_t^{(p)}$ at $x = e_p$. Every fact-score map is linear. The high-level map H_t applies the preselected legal ground-rule test and requires no additional controller state. Therefore $\alpha_{t+1}(x) = H_t(\alpha_t(x))$ at every declared native state even though x never changes. Initial facts match by construction.

An affine vocabulary readout can assign a larger logit to y_1 exactly when $[k_T^{(p)}]_q = 1$ and to y_0 otherwise, with every other token given a smaller logit. The logits can be affine in e_p. The native output is therefore correct for every input in the collection. A fixed positive-stabilizer normalization on these unit basis vectors only multiplies the nonconstant linear read by a fixed positive scalar and therefore does not alter the construction. The construction is a transition-system counterexample, not an empirical Transformer result.

For the converse statement, unchanged persistent registers give the same w at both boundaries. Any fixed deterministic fact readout therefore gives the same bits; in particular, the shared affine scores $\varphi_a(w)$ and their thresholded bits are identical. A verified transport satisfying (16) gives the same conclusion. Thus a newly derived fact cannot be produced solely by the stage-dependent read used in the construction. □

The counterexample does not show that linear probing is never informative. It is a formal version of the concern that a probe can perform the computation itself (Hewitt & Liang, 2019; Ravichander et al., 2021), and it shows exactly what step-dependent linearity alone fails to exclude. A size bound may exclude a lookup matrix for a large input family, but the argument also applies to a small collection on which that matrix is inexpensive. Shared interpretation and native-write verification are therefore separate requirements, not consequences of a favorable parameter count.

Theorem 9 (Exact cell criterion)**.** *For a supplied extractor and an explicit program of the form* (2)*, exact native execution and the specified terminal decision are preserved if and only if* (3) *holds for all nonempty cells. If H_t is not specified in advance, a deterministic high-level update exists if and only if $\alpha_{t+1} \circ F_t$ is constant on every $D_t(z)$. Such an update has Horn form only if its fact component agrees with* (1) *for some declared ground rule or with the idle update. If the selector and control update are additionally required to belong to a bounded grammar, this requirement must be checked separately.*

B.2 Proof of Theorem 9

Proof. For a supplied H_t, take $x \in D_t(z)$. If the transition identity holds, then $\alpha_{t+1}(F_t(x)) = H_t(z)$, which is exactly membership of $F_t(x)$ in $D_{t+1}(H_t(z))$. This proves necessity of the first inclusion.

Conversely the same membership statement implies the identity for every x in every nonempty cell, and these cells cover X_t. The terminal inclusion is exactly the stated decision identity, so both directions hold there as well.

If H_t is not supplied, necessity follows because any deterministic high-level update must assign the same successor to native states with the same extracted abstract state. Conversely, suppose $\alpha_{t+1} \circ F_t$ is constant on every nonempty cell $D_t(z)$. Define the successor of a represented abstract state z by $H_t(z) = \alpha_{t+1}(F_t(x))$ for any $x \in D_t(z)$. Because the value is constant on the cell, this definition does not depend on the choice of x. It defines a unique deterministic update on represented abstract states. No regularity or computational bound is implied by this factorization.

For a Horn-form update, the fact component of the successor must belong to the finite set $\{\mathsf{U}_g(k) : g \in \mathcal{G} \cup \{\circ\}\}$, as required by (2). Conversely, suppose this condition holds and no syntactic restriction is imposed. Fix an ordering of the matching rule labels and choose the first one, and set the control update to the corresponding determined successor control state. This defines a Horn-form update on all represented abstract states. The update may be defined arbitrarily on abstract states not represented by any native state, provided that no additional invariant-domain condition is imposed on those states. A restriction to a particular expression grammar must still be verified; finite-table definability is not automatically definability at the requested cost. This distinction also matters for counterexample-guided discovery in a restricted candidate class. □

C Shared binding from the original query and key maps

C.1 Address conventions and the scope of the certificate

An address has h_{addr} orthogonal coordinate blocks. Its first block represents a predicate, and the remaining blocks represent ordered arguments and padding. Arity is determined by the predicate; a padding symbol is distinct from every entity label. Each block uses unit vectors with within-block separation $\langle e_u, e_v \rangle \le 1 - \delta_{\mathrm{addr}}$ for $u \neq v$. Distinct atoms differ in at least one block. The scalar δ_{addr} may be chosen as any common positive lower bound on these finitely many separations.

The selected rule and its current binding are part of the abstract control where they vary with the input. In Theorem 1, all states of one binding domain assign the same entity to each variable. Different binding domains may represent different substitutions. The actual maps B_V^{bind}, their allowable register transport, the query decompositions, and the association of each included key with an atom are certificate data. The theorem does not infer predicate labels from vector geometry. It verifies consistency of native operations after the meanings of those labels and registers have been specified independently of the verification.

A query decomposition copies the same vector β_V at each occurrence of variable V. It is not enough that separate fitted vectors happen to be close on a tested collection. On a source subset, the original effective query rows must equal the declared linear maps that copy register values plus a remainder bounded uniformly there. If s_i variable occurrences appear in the ith query, orthogonality of the address blocks makes the squared contribution of binding errors at most $s_i \epsilon_{\mathrm{bind}}^2$. Repeated occurrences of one variable are counted repeatedly, but do not become independent bindings. Constants and predicate labels are represented exactly in this calculation; any errors in them enter $\epsilon_{\mathrm{qry}}^{\mathrm{rem}}$. This proves (4) by the triangle inequality.

C.2 Proof of Theorem 1

Proof. For distinct entity labels, $\|e_u - e_v\|_2^2 = 2 - 2\langle e_u, e_v \rangle \ge 2\delta_{\mathrm{addr}}$. If a binding-register value were within ϵ_{bind} of two such labels, the distance between those labels would be at most $2\epsilon_{\mathrm{bind}}$, contradicting the strict inequality above. Hence the value of each variable-binding register corresponds to a unique declared entity. Since each query copies the same register for each occurrence of that variable, the substitution θ is common to all premises.

Write $a_* = B_i\theta$ for the target atom of one read. Every address has squared norm h_{addr}, while a distinct address has inner product with $\mathrm{addr}(a_*)$ at most $h_{\mathrm{addr}} - \delta_{\mathrm{addr}}$. Thus the ideal target-minus-competitor dot-product gap is at least δ_{addr}.

If u_0, v_0 are ideal query and key addresses, $u = u_0 + e_u$, and $v = v_0 + e_v$, then

$$|\langle u, v\rangle - \langle u_0, v_0\rangle| \le \sqrt{h_{\mathrm{addr}}}(\epsilon_{\mathrm{qry}} + \epsilon_{\mathrm{key}}) + \epsilon_{\mathrm{qry}}\epsilon_{\mathrm{key}}. \tag{18}$$

Apply this bound separately to the target and competitor, multiply by κ_{att}, and subtract their possible positional-bias difference of at most $2\epsilon_{\mathrm{pos}}$. This gives (5). The bound does not require independent query errors across competitors.

Let j_* be the unique included target position. Put $u_{\mathrm{leak}} = \sum_{j \neq j_*} \exp(\ell_{ij} - \ell_{ij_*})$. Then $u_{\mathrm{leak}} \le (N_{\mathrm{slot}} - 1)e^{-\Delta_{\mathrm{bind}}}$, and the total non-target mass is $u_{\mathrm{leak}}/(1 + u_{\mathrm{leak}})$. Subtract the target projected value from the attention-weighted value average and apply the value-diameter bound. Finally, add the error between the target projected value and its declared numerical premise representation. This proves (6). With one included position, the leakage sum is empty and the same formula gives zero leakage. □

The assumption of a unique target position simplifies the statement. Multiple positions representing the same atom may instead be treated as a target set, provided that their projected values have the same verified premise representation; in that case, the attention mass assigned to the entire target set must be bounded. Another position with the same address cannot be treated as a distinct-address competitor. If every possible premise has an explicitly represented position, absence may be encoded by the declared false value, such as zero or minus one. If only present premises are stored, a separate verified mechanism is required to represent absence. These cases require different attention-read conditions and therefore cannot be interchanged without verifying the corresponding native attention computation.

C.3 Persistence through normalization and residual branches

For one actual branch $x' = x + O\,u(x) + b^{\mathrm{res}}$, multiplying by B_V^{bind} gives

$$B_V^{\mathrm{bind}}x' = B_V^{\mathrm{bind}}x + B_V^{\mathrm{bind}}O\,u(x) + B_V^{\mathrm{bind}}b^{\mathrm{res}}. \tag{19}$$

The two zero-product identities in the main text make this equal to β_V at every source state. No property of u is needed beyond being the original defined expression. Consequently the argument applies to normalized attention, ordinary feed-forward blocks, and gated feed-forward blocks. Every output column and bias must be included. For a sequence of such branches, induction preserves the binding registers. A cache write or register relocation is covered only when its own exact read-after-write or transport identity is supplied. If the native update intentionally changes the binding, its new binding domain and its control meaning must instead be verified.

The numerical identity certifies consistency of the native shared read, not the correctness of the external entity interpretation; conversely, external entity labels do not establish the native shared-read identity. Both conditions are required to derive a common substitution.

D Complete-state invariant domains and local certificates

D.1 Domains, fixed coordinates, and initialization

The structural theorem uses one common continuous coordinate layout across the certified segment. A boundary's program position, mask, and cache type are discrete data. The schedule g_t is fixed by these boundaries. Any change in the program position, mask, or cache type must either be represented in the state description at the next boundary or be determined by a verified native branch condition. In particular, the fixed schedule cannot substitute for a state-dependent rule selector implemented by the native computation.

The vector x contains the complete native continuous state needed by the continuation. A coordinate with radius zero is fixed at its center on that box; this does not remove its dependence on the logical input. For example, an immutable cache value can equal $2k_a - 1$ for an input-only atom a. Its center differs between initial fact assignments, but the ground-rule updates leave that atom unchanged, so its center is identical at the required successor. Such a coordinate is valid precisely when its native write is the corresponding exact identity. Any state coordinate with nonzero variation must therefore be assigned a nonzero radius or represented separately.

The outer boxes and the original functions are fixed before the radius inequalities are solved. Every operation must remain well defined on these boxes; in particular, square-root arguments must remain positive and attention must be taken over nonempty sets. Continuous differentiability is required on an open neighborhood of each outer box, including when some coordinates have zero radius. The line segment from a center to any point in its box remains in that box, so the argument never interpolates between reference states associated with different fact codes.

The theorem applies to initialized states within the certified inner boxes. Applying it to prompts additionally requires verifying that the native initialization map enters those boxes with the prescribed fact bits; the instance below instead specifies its initial residual states and cache contents directly.

D.2 Full proof of Theorem 2

Proof. Fix a code k, a block F_t, and $x \in \mathcal{Q}_r(k)$. Since $r \le R^{\mathrm{box}}$, every point $x(u) = \chi(k) + u(x - \chi(k))$, $0 \le u \le 1$, belongs to the outer box. The fundamental theorem of calculus, applied coordinatewise to the original differentiable map, gives

$$F_t(x) - F_t(\chi(k)) = \int_0^1 DF_t(x(u))\,[x - \chi(k)]\,du. \tag{20}$$

Taking componentwise absolute values and using the nonnegative comparison matrix yields

$$|F_t(x) - \chi(\mathsf{U}_{g_t}(k))| \le \varepsilon^{\mathrm{anc}} + \Lambda|x - \chi(k)| \le \varepsilon^{\mathrm{anc}} + \Lambda r \le r. \tag{21}$$

This proves the asserted inclusion for the complete state vector. In particular, every zero-radius successor coordinate has its specified value, so no unaccounted cache write is permitted.

For every $x \in \mathcal{Q}_r(k)$, affinity and (7) imply

$$|\varphi_a(Px) - (2k_a - 1)| = |v_a^{\mathsf{T}} P(x - \chi(k))| \le |v_a^{\mathsf{T}} P| r \le 1 - \gamma_{\mathrm{fact}}. \tag{22}$$

Thus true scores are at least γ_{fact}, and false scores are at most $-\gamma_{\mathrm{fact}}$. Each inner box has the declared fact code. Distinct fact codes cannot have intersecting inner boxes. Since the control state is fixed by the declared boundary, the complete extracted successor is $H_t(k, c_t^{\mathrm{ctrl}})$. Taking the union of these boxes as the declared native domain gives the transition part of (3).

Inductively, every initialized state stays in the appropriate box at every boundary. All premise-representation, shared-binding, normalization, and derivative bounds already verified uniformly on the outer boxes therefore apply to the actual subsequent states. No additional successor-domain assumption is required in this induction. The theorem does not by itself provide a terminal output margin; that requires a separate calculation for the original readout. □

D.3 Least comparison radii and their limitations

Suppose a set of coordinates is fixed with zero radius and preserved exactly by all original blocks. Let $\mathcal{J}$ index the remaining coordinates. For each exactly preserved coordinate, the corresponding update row has no dependence on the mutable coordinates. Their radius-dependent contributions vanish because these coordinates have zero radius, while exact preservation gives zero reference-state error in the corresponding coordinates. Write $\Lambda_{\mathcal{J}\mathcal{J}}$ for the remaining principal comparison matrix and $\varepsilon^{\mathrm{anc}}_{\mathcal{J}}$ for the corresponding vector of errors at reference states.

Proposition 10 (Least radius for a common comparison matrix)**.** *If $\Lambda_{\mathcal{J}\mathcal{J}} \ge 0$ has spectral radius less than one, then*

$$r_{*,\mathcal{J}} = (I_{\mathcal{J}} - \Lambda_{\mathcal{J}\mathcal{J}})^{-1}\varepsilon^{\mathrm{anc}}_{\mathcal{J}} = \sum_{j=0}^{\infty}(\Lambda_{\mathcal{J}\mathcal{J}})^j \varepsilon^{\mathrm{anc}}_{\mathcal{J}} \tag{23}$$

is the least nonnegative vector satisfying $\varepsilon^{\mathrm{anc}}_{\mathcal{J}} + \Lambda_{\mathcal{J}\mathcal{J}} r_{\mathcal{J}} \le r_{\mathcal{J}}$. This radius is admissible only if the outer-domain and fact-margin bounds are also satisfied.

Proof. Spectral radius less than one implies convergence of the matrix Neumann series. Every term is nonnegative, so the stated vector is nonnegative and satisfies the comparison equation with equality. If $r_{\mathcal{J}} \ge \varepsilon^{\mathrm{anc}}_{\mathcal{J}} + \Lambda_{\mathcal{J}\mathcal{J}} r_{\mathcal{J}}$, repeated substitution gives

$$r_{\mathcal{J}} \ge \sum_{j=0}^{m}(\Lambda_{\mathcal{J}\mathcal{J}})^j \varepsilon^{\mathrm{anc}}_{\mathcal{J}} + (\Lambda_{\mathcal{J}\mathcal{J}})^{m+1} r_{\mathcal{J}} \ge \sum_{j=0}^{m}(\Lambda_{\mathcal{J}\mathcal{J}})^j \varepsilon^{\mathrm{anc}}_{\mathcal{J}}. \tag{24}$$

Taking the limit proves minimality. Extending by zero on the fixed coordinates gives the full radius, provided their exact preservation has been checked. □

The spectral condition is sufficient, not necessary for existence of a comparison radius. A preserved coordinate has derivative one and zero error at its reference states, so a full comparison matrix can have spectral radius one while admissible radii exist. Likewise, separate blockwise conditions $\varepsilon_t^{\mathrm{anc}} + \Lambda_t r \le r$ form a finite linear feasibility problem for a common box without needing a single dominating contraction matrix. When exact rational bounds are used, these radius constraints are ordinary rational linear inequalities.

For a scalar maximum-norm bound $\|F_t(\chi(k)) - \chi(\mathsf{U}_{g_t}(k))\|_\infty \le e_{\mathrm{uni}}$ and a uniform Jacobian operator bound $\|DF_t\|_{\infty\to\infty} \le L_{\mathrm{uni}}$ in mutable coordinates, the same line-segment argument yields the radius condition $e_{\mathrm{uni}} + L_{\mathrm{uni}} r_{\mathrm{uni}} \le r_{\mathrm{uni}}$. This is an operator-norm version of the theorem, not a claim that each Jacobian entry is bounded by $L_{\mathrm{uni}}/d_{\mathrm{nat}}$. The scalar form is used in the instance below.

For a finite horizon one can instead propagate nonnegative radii by $r_{t+1} = \varepsilon_t^{\mathrm{anc}} + \Lambda_t r_t$, checking each against its outer domain and score margins. This provides a valid finite-horizon certificate without contraction. A constant radius has the stronger advantage that the same verification domain can be reused for every repetition of the checked block templates.

D.4 Local verification at reference states

Proof of Proposition 3 and (10). Fix t, j, k. The triangle inequality and (9) give

$$|F_{t,j}(\chi(k)) - [\chi(\mathsf{U}_{g_t}(k))]_j| \le \zeta_{t,j} + |\psi_{t,j}(k_{S_{t,j}}) - \omega_{t,j}(k_{S_{t,j}})|. \tag{25}$$

The second term is one of at most $2^{|S_{t,j}|}$ local assignments, so its maximum is bounded by (10). Maximizing over the finite block templates yields a common error bound at the reference states. If some local assignments cannot occur globally, including them only makes the bound more conservative.

The explicit enumeration uses at most $\sum_{t,j} 2^{|S_{t,j}|}$ evaluations of the scalar local expressions. Bounding each support size gives the claimed count. For a Horn head, the successor bit is the disjunction of its old bit and the conjunction of at most $m_{\max}$ body bits; it therefore depends on at most $m_{\max} + 1$ distinct atoms. A non-head bit is retained and depends on one atom. Auxiliary or cached output coordinates must have their own checked dependency sets; they do not inherit this bound merely because the fact update is local. The last assertion of the proposition therefore applies only when this locality condition holds for all relevant output coordinates. □

A certified remainder is essential. Finite softmax normally mixes all included value positions. Even when its intended target is local, the exact output at a reference state can depend weakly on every fact encoded in those values. A uniform bound on the contributions from non-target positions accounts for this dependence through $\zeta_{t,j}$. Declaring the tail to be zero would change the native map and invalidate the certificate. The local expression is used only for bounding; the actual inference still evaluates its original softmax.

The count above does not include the cost of range analysis or numerical precision requirements. For example, a dense normalization derivative can require a dense comparison matrix even when the calculations at reference states are local. The number of ground rule templates can itself be exponential in the length of a compact description of rules before grounding. These costs must be reported separately from the local assignment count.

D.5 Reference states with equal squared norms and the actual normalizer

Consider one native normalization applied to a selected residual vector. Let L^{norm} denote the fixed linear map from complete state coordinates to the centered or uncentered vector in its denominator. For RMS normalization it includes the residual selection; for centered layer normalization it also includes the centering projection. Suppose the centers are affine,

$$\chi(k) = x^{\mathrm{cen}} + E^{\mathrm{code}}(2k - \mathbf{1}), \tag{26}$$

and the vectors $L^{\mathrm{norm}}E^{\mathrm{code}}_{:a}$ are pairwise orthogonal and each is orthogonal to $L^{\mathrm{norm}}x^{\mathrm{cen}}$. Then

$$\|L^{\mathrm{norm}}\chi(k)\|_2^2 = \|L^{\mathrm{norm}}x^{\mathrm{cen}}\|_2^2 + \sum_{a\in\mathcal{A}} \|L^{\mathrm{norm}}E^{\mathrm{code}}_{:a}\|_2^2. \tag{27}$$

Indeed, the cross terms vanish and every sign coefficient has square one. The original positive stabilizer and dimension divisor therefore yield the same exact normalization denominator at all reference centers. This equality holds at the reference centers only and does not imply that the normalization denominator is constant on a neighborhood of those centers.

To obtain local reference expressions, the remaining affine rows, write columns, and any intervening operations must still have the claimed support, or a uniformly bounded nonlocal remainder. In a multi-sublayer block, the statement must be checked at the reference inputs to each normalizer, not only at the block's initial center. The instance below uses signed premise reference values, so its post-attention reference norm is also independent of the fact assignment. The deviation caused by the actual finite-temperature softmax is bounded explicitly rather than assumed to vanish.

D.6 Proof of the smooth-refresh certificate

Proof of Proposition 4. Fix a source fact assignment and one actual block. For a fact stored in the current residual, the signed center is $2k_a - 1$. The source-radius and normalization bounds place its normalized coordinate within $r_{\mathrm{uni}} + \delta_{\mathrm{norm}}$ of that center. A normalized scratch premise is within $r_{\mathrm{uni}} + \epsilon_{\mathrm{ret}} + \delta_{\mathrm{norm}}$ of its own signed fact. A premise read directly from the current residual satisfies the same bound with a smaller retrieval contribution.

For a rule with $m \geq 1$ premises, define the exact signs $\sigma_{\mathrm{head}} = 2k_{\mathsf{h}_g} - 1$ and $\sigma_i = 2k_{B_i} - 1$ for the proof. If the head is already true, $m\sigma_{\mathrm{head}} + \sum_i \sigma_i + 1 \geq 1$, even when every premise is false. If the head is false and all premises are true, the same expression equals one. If the head is false and at least one premise is false, it is at most minus one. Thus its sign is the Horn successor. This conclusion remains true with repeated body atoms; the bound is conservative when their errors are correlated.

The actual head preactivation differs from this ideal expression by at most $2m(r_{\mathrm{uni}} + \delta_{\mathrm{norm}}) + m\epsilon_{\mathrm{ret}} + \epsilon_{\mathrm{gate}}$. The second condition in (11) therefore ensures that its magnitude is at least $1/2 + \tau_{\mathrm{act}}$ and that its sign agrees with the Horn successor. The first condition gives the same conclusion for every non-head input. For an empty body, $u_{\mathsf{h}_g} = 1$ has the always-true successor sign and directly satisfies the condition that the input lies outside the transition interval.

For every u with $|u| \geq 1/2+\tau_{\mathrm{act}}$, the two arguments $2u+1$ and $2u-1$ have the same sign and absolute values at least $2\tau_{\mathrm{act}}$. Replacing SiLU by ReLU in the corresponding comparison expression gives exactly $\mathrm{sign}(u)$ on this domain. Using the actual SiLU bound $|f_{\mathrm{S}}(v) - \max\{v, 0\}| \leq |v|e^{-|v|}$, the difference of the native expression from that sign is at most the sum of two terms $|2u \pm 1|e^{-\kappa_{\mathrm{act}}|2u\pm1|}$. The function $ve^{-\kappa_{\mathrm{act}}v}$ is decreasing for $v \geq 2\tau_{\mathrm{act}}$ because $2\kappa_{\mathrm{act}}\tau_{\mathrm{act}} \geq 1$. Hence

$$|C_{\kappa_{\mathrm{act}}}(u) - \mathrm{sign}(u)| \leq 4\tau_{\mathrm{act}}e^{-2\kappa_{\mathrm{act}}\tau_{\mathrm{act}}}. \tag{28}$$

This reference is used only to prove an error bound; every actual nonlinear evaluation is still SiLU.

In the exact native residual formula, $s_a - \widetilde{s}_a$ has magnitude at most δ_{norm}. Adding the tail bound and $\epsilon_{\mathrm{write}}$ therefore places every successor fact within r_{uni} of its prescribed signed successor. A scratch successor is its post-attention value minus its normalized value plus a remainder, so its distance from its zero center is at most $\delta_{\mathrm{norm}} + \epsilon_{\mathrm{write}} \leq r_{\mathrm{uni}}$. All other coordinates are protected and preserved exactly by hypothesis. This proves complete successor-box inclusion. Induction establishes repetition, with no claim that a finite-depth decoder has more blocks than its declared horizon. □

In the constructed instance, take $\tau_{\mathrm{act}} = 1/4$, $\kappa_{\mathrm{act}} = 32$, $m_{\max} = 2$, $r_{\mathrm{uni}} = 1/32$, $\delta_{\mathrm{norm}} = 1/2048$, $\epsilon_{\mathrm{ret}} = 1/1024$, $\epsilon_{\mathrm{gate}} = 0$, and $\epsilon_{\mathrm{write}} = 0$. The first two inequalities are strict, and the last follows from $e^{-16} < 1/65536$. Thus the direct smooth-refresh result and the Jacobian comparison result certify the same repeated native execution by two compatible arguments.

The negative tail of SiLU alone does not imply cumulative drift, because other residual contributions may compensate for it. Both a drift counterexample and a positive certificate must concern the complete native update.

E SHARED TEMPLATES WITH INTERNALLY GENERATED SUBSTITUTIONS

We prove Theorem 5. The construction retains the declared internal-state input interface: one live residual and fixed null key–value caches. The seven stages repeat in a fixed cycle, while the active rule and its substitution are represented in mutable coordinates and updated by the original network operations.

E.1 INTERNAL ENUMERATION OF RULE SUBSTITUTIONS

Let the nonempty constant set be $\{a_0, \dots, a_{c-1}\}$, with $c \geq 1$, and let the nonempty list of rule schemata be $\rho = 1, \dots, R$. Schema ρ has an ordered list of all its distinct variables $V_{\rho,1}, \dots, V_{\rho,v_\rho}$, a body with m_ρ occurrences, and one head. Variables occurring only in the head are included in this list. Put

$$v = \max_\rho v_\rho,\ N_{\text{prem}} = \sum_{\rho=1}^{R} m_\rho,\ G_{\text{sub}} = \sum_{\rho=1}^{R} c^{v_\rho}. \tag{29}$$

The number G_{sub} counts pairs consisting of a schema and a substitution; different pairs may produce the same ground clause. Neither G_{sub} nor the list of these pairs is stored in the network.

The persistent logical state is

$$z = (k, \rho, \mathbf{d}),\ k \in \{0,1\}^n,\ \mathbf{d} \in \{0, \dots, c-1\}^v,\ d_j = 0 \text{ for } j > v_\rho. \tag{30}$$

We call the last condition the controller validity condition. Its substitution is $\theta_{\rho,\mathbf{d}}(V_{\rho,j}) = a_{d_j}$. The active schema and its binding are read from mutable state coordinates. For an inactive schema, its premise queries also use the first v_ρ digits of the same current vector $\mathbf{d}$; these reads have no logical effect because their subsequent action gates include the schema's phase bit.

One abstract step first applies the ground Horn update $\mathsf{U}_{\rho\theta_{\rho,\mathbf{d}}}$ to k. It then increments the first v_ρ digits in base c, with digit one least significant. On overflow it resets those digits to zero and advances ρ cyclically. Digits beyond v_ρ remain zero. We denote this complete update by H. The counter update is independent of whether the rule body is true.

For each schema ρ, let $\mathcal{H}_\rho \subseteq \mathcal{A}$ contain the ground atoms having its head predicate and its specified constant arguments. This set is constructed from the head syntax and the existing atom addresses. It does not enumerate body substitutions. Let

$$A_{\text{head}} = \sum_{\rho=1}^{R} |\mathcal{H}_\rho| \leq Rn. \tag{31}$$

For $a \in \mathcal{H}_\rho$, let $\mathcal{J}_{\rho,a}$ list the pairs (j, u) contributed by the variable occurrences in the head when the corresponding argument of a is a_u. Repeated occurrences may remain in this list. All their tests read the same binding register.

E.2 REGISTERS AND THE SEVEN NATIVE STAGES

Every mutable coordinate has a signed Boolean reference value in $\{-1, 1\}$. The fact coordinate s_a represents k_a by $2k_a - 1$. The binding coordinates $b_{j,u}$ and phase coordinates p_ρ use one-hot codes in this signed convention: exactly one coordinate in each group has value 1, and all the others have value -1. The binding group j represents d_j, and the phase group represents the active schema. All working coordinates have reference value -1 at a macro boundary. The complete mutable register list is

Registers	Number	Contents at a macro boundary
s_a	n	Fact signs
$b_{j,u}$	vc	Binding one-hot codes
p_ρ	R	Schema one-hot code
$r_{\rho,i}$	N_{prem}	Premise working coordinates, all -1
$z_{\rho,a}$	A_{head}	Action working coordinates, all -1
ι_j	v	Increment bits, all -1
o_ρ	R	Schema overflow bits, all -1
o	1	Global overflow bit, -1
$t^{\text{inc}}_{j,u}, t^{\text{keep}}_{j,u}$	$2vc$	Binding multiplexer coordinates, all -1
$h^{\text{adv}}_\rho, h^{\text{keep}}_\rho$	$2R$	Phase multiplexer coordinates, all -1

Thus the mutable dimension is

$$D = n+vc+R+N_{\text{prem}}+A_{\text{head}}+v+R+1+2vc+2R = n+3vc+4R+N_{\text{prem}}+A_{\text{head}}+v+1. \quad (32)$$

Every other native coordinate, including the fixed null-cache keys and values and the protected normalization coordinate, is separately included in the complete state and preserved exactly. Atom identities determine fixed parameter indices rather than additional mutable state fields. The reference map $\chi(z)$ concatenates the three persistent codes and the working values just specified. The shared extractor reads the signs of the fact, binding, and phase registers.

We describe the target of each native stage on these reference codes. The following targets define the reference code maps implemented by the native affine–SiLU–affine operations. For signed inputs $x_1, \ldots, x_b$, conjunction and disjunction use the affine preactivations

$$u_\wedge(x_1, \ldots, x_b) = \sum_{i=1}^{b} x_i - (b-1),\ u_\vee(x_1, \ldots, x_b) = \sum_{i=1}^{b} x_i + (b-1). \quad (33)$$

Their signs have the required truth tables, with absolute preactivation at least one. An empty conjunction is the constant 1, and an empty disjunction is the constant -1. A preserved coordinate has target equal to its current sign; a cleared coordinate has target -1. The smooth refresh construction implements these signs and preserves their margins on the declared neighborhoods. In Stages 2–7, every attention output row is zero, so the attention residual is the identity and the feed-forward residual performs the specified update. The original normalization operations remain present.

Stage 1: premise reads. For every schema and every body occurrence, use its fixed syntactic query map with the current binding registers to address that ground body atom in the current fact memory. The reference target is

$$r'_{\rho,i} = 2k_{B_{\rho,i}\theta_{\rho,\mathbf{d}}} - 1. \quad (34)$$

In this formula $\theta_{\rho,\mathbf{d}}$ denotes the substitution obtained from the current digits for schema ρ, whether or not ρ is active. All occurrences of $V_{\rho,j}$ in every query use the same group $b_{j,:}$. The attention reads include newly derived facts because their values come from the current fact registers. A premise working coordinate starts at -1; when its attention residual receives a signed value w, the refresh input uses the resulting coordinate plus one, recovering w at the ideal read. The finite-softmax and normalization errors in this operation are included in the native implementation bounds. All other coordinates retain their reference values.

Stage 2: schema actions. For each ρ and $a \in \mathcal{H}_\rho$, set $z_{\rho,a}$ to the signed conjunction of the phase bit, all the schema's premise bits, and all its head-variable occurrence tests. Its affine preactivation is

$$u^{\text{act}}_{\rho,a} = p_\rho + \sum_{i=1}^{m_\rho} r_{\rho,i} + \sum_{(j,u)\in\mathcal{J}_{\rho,a}} b_{j,u} - (m_\rho + |\mathcal{J}_{\rho,a}|). \quad (35)$$

The same binding coordinates therefore determine both the premise addresses and the head selected for the write. Repeated head-variable occurrences contribute repeated reads, not separate bindings. All other coordinates retain their reference values.

Stage 3: fact writes. For every atom a, take the disjunction of its old fact sign and all action bits with that atom as target. Its affine preactivation is

$$u_a^{\text{fact}} = s_a + \sum_{\rho: a \in \mathcal{H}_\rho} z_{\rho,a} + |\{\rho : a \in \mathcal{H}_\rho\}|. \tag{36}$$

At the same stage, clear all premise and action working coordinates. All outputs of the feed-forward operation are computed from the incoming state, so clearing an action coordinate does not alter its use in the fact output of this stage. Binding and phase coordinates are preserved.

Stage 4: increment and schema overflow. Write $\mathcal{E}_j = \{\rho : v_\rho \geq j\}$ and $e_j = |\mathcal{E}_j|$. The increment coordinate ι_j uses the single affine preactivation

$$u_j^{\text{inc}} = \sum_{\rho \in \mathcal{E}_j} p_\rho + \sum_{\ell < j} b_{\ell,c-1} + e_j - j. \tag{37}$$

This gate does not require an additional stage to compute whether the current schema uses digit j. If the active schema lies in $\mathcal{E}_j$, its first sum is $2 - e_j$; otherwise it is $-e_j$. If f_j of the lower digits are not maximal, the second sum is $j - 1 - 2f_j$. Hence the preactivation is $1 - 2f_j$ in the first case and $-1 - 2f_j$ in the second. Its positive sign is exactly the condition that digit j is active and every lower digit is maximal. The schema overflow coordinate uses

$$u_\rho^{\text{over}} = p_\rho + \sum_{j=1}^{v_\rho} b_{j,c-1} - v_\rho. \tag{38}$$

It is positive exactly when this schema is active and all its digits are maximal. Every persistent coordinate is preserved.

Stage 5: global overflow and binding multiplexer. The global overflow preactivation is

$$u^{\text{over}} = \sum_{\rho=1}^{R} o_\rho + R - 1. \tag{39}$$

For every binding coordinate, compute the two conjunctions with preactivations

$$u_{j,u}^{\text{inc}} = \iota_j + b_{j,(u-1) \bmod c} - 1, \; u_{j,u}^{\text{keep}} = -\iota_j + b_{j,u} - 1. \tag{40}$$

Their signed outputs are stored in $t_{j,u}^{\text{inc}}$ and $t_{j,u}^{\text{keep}}$. The binding and phase coordinates are still preserved at this stage.

Stage 6: binding update and phase multiplexer. Update the binding coordinate $b_{j,u}$ using the disjunction preactivation

$$u_{j,u}^{\text{bind}} = t_{j,u}^{\text{inc}} + t_{j,u}^{\text{keep}} + 1. \tag{41}$$

Let $\text{prev}(1) = R$ and $\text{prev}(\rho) = \rho - 1$ for $\rho > 1$. At this same stage compute

$$u_\rho^{\text{adv}} = o + p_{\text{prev}(\rho)} - 1, \; u_\rho^{\text{keep}} = -o + p_\rho - 1, \tag{42}$$

and store their signs in h_ρ^{adv} and h_ρ^{keep}. These two expressions depend only on the incoming phase code and overflow bit. They do not depend on binding coordinates, so the simultaneous binding update cannot change the phase selection. The old phase and the global overflow coordinate are preserved until the next stage.

Stage 7: phase update and working-coordinate reset. Update p_ρ using

$$u_\rho^{\text{phase}} = h_\rho^{\text{adv}} + h_\rho^{\text{keep}} + 1. \tag{43}$$

Clear ι, all schema overflow coordinates, the global overflow coordinate, and both multiplexer groups. Preserve the updated facts and bindings. The premise and action coordinates, already cleared in Stage 3, remain at -1. The macro boundary therefore has exactly the register form required by $\chi(z)$.

E.3 EXACT LOGICAL AND CONTROLLER SEMANTICS

Lemma 11 (One macro step)**.** *On every valid reference state, the seven stage targets above produce $\chi(H(z))$. The fact update uses exactly the substitution represented by the incoming binding registers, and the resulting controller remains valid.*

Proof. Let the incoming active schema be ρ_*. Stage 1 gives the signed truth value of each instantiated premise before any fact is changed. For $\rho \neq \rho_*$, every action gate contains a false phase bit. For $\rho = \rho_*$, the one-hot bindings satisfy all head-variable tests for exactly the atom $a_* = \mathsf{h}_{\rho_*}\theta_{\rho_*,\mathbf{d}}$. This atom belongs to $\mathcal{H}_{\rho_*}$: its predicate and constant arguments agree with the head syntax. Conversely, any different compatible atom differs at a variable argument, and its corresponding binding test is false. If the head has no variables, the compatible set has exactly its fixed ground atom, and the same conclusion holds with no variable tests. Thus the only action that can be true is the one for a_*, and it is true exactly when every current body fact is true. Stage 3 takes the disjunction with each old fact. Its result is precisely $\mathsf{U}_{\rho_*\theta_{\rho_*,\mathbf{d}}}(k)$, including when the head was already established or occurs in its own body.

By (37), digit j rotates its one-hot code exactly when $j \leq v_{\rho_*}$ and all lower digits are maximal. The two binding multiplexer branches are mutually exclusive; their disjunction therefore either retains the code or advances its represented value by one modulo c. This is the usual base-c increment on the first v_{ρ_*} digits. Digits above that index do not rotate.

The overflow expressions in (38) and (39) make o true exactly when all active digits are maximal. The phase multiplexer then retains the phase or advances it cyclically. In the overflow case every active digit rotates from its maximal value to zero. Every previously inactive digit is already zero by controller validity. All digits are consequently zero when the phase changes, so validity holds even when consecutive schemata have different numbers of variables. In the nonoverflow case the phase does not change and inactive digits stay zero. Stage 7 restores every working coordinate to its boundary value, completing the claim. □

Lemma 12 (Internal coverage)**.** *The valid controller states form one cycle of length $G_{\mathrm{sub}} = \sum_\rho c^{v_\rho}$ under the specified native controller circuit. Every G_{sub} consecutive macro steps select each schema–substitution pair exactly once.*

Proof. For a valid controller state define the proof-only index

$$L(\rho, \mathbf{d}) = \sum_{\sigma<\rho} c^{v_\sigma} + \sum_{j=1}^{v_\rho} d_j c^{j-1}. \tag{44}$$

Within phase ρ, the second sum ranges bijectively over $\{0, \ldots, c^{v_\rho} - 1\}$, because the remaining digits are fixed at zero. The phase intervals are consecutive and disjoint, so L is a bijection from valid controller states to $\{0, \ldots, G_{\mathrm{sub}} - 1\}$. The increment circuit advances the second sum by one until overflow. At overflow, the zeroed counter in the next phase has the next index; the last phase returns to index zero. Thus every update satisfies

$$L(\rho', \mathbf{d}') = (L(\rho, \mathbf{d}) + 1) \bmod G_{\mathrm{sub}}. \tag{45}$$

The index L is used only in the proof; the implemented controller operates directly on the one-hot state. Equation (45) therefore establishes that one full cycle visits every variable assignment for every schema, independently of the facts. □

E.4 BOUNDARY CASES

If $c = 1$, every binding group has one true coordinate. The global overflow is true at every step and the phase advances each time, giving $G_{\mathrm{sub}} = R$; the binding multiplexer still returns its unique code. If $v_\rho = 0$, (38) reduces to the phase bit, so that schema has one test per cycle. If $v = 0$, all binding and increment groups are absent; the remaining stages are well defined. An empty body contributes no premise coordinates and an empty conjunction to the action gate, so its currently instantiated head is always established. Variables occurring only in a head are counted and bound exactly as the other variables. Repeated variables, self dependencies, and recursive programs need no change to the update or coverage proof. If there are no rules, the controller and deduction stages can be omitted and the initial fact set is already closed.

E.5 Uniform implementation of the Boolean stages

Every mutable coordinate has a signed Boolean center in $\{-1, 1\}$; a false temporary value has center -1. Let D be the number of mutable coordinates and let $d = D + 1$. The remaining residual coordinate is protected and has value C_{fix}. Every immutable cache entry also has radius zero. For R rule schemas, at most v variables per schema, at most m premises per schema, and at most $a_{\max}$ variable occurrences in an atom, put $B = R + v + m + a_{\max} + 3$. The algebraic requirements used below are that every specified threshold row satisfies $\|a_j\|_1 \le B$ and $|b_j| \le B + 1$; these bounds are checked against the stated Boolean stages. Set

$$r_{\mathrm{sh}} = \frac{1}{64(B+1)},\ C_{\mathrm{fix}} = \sqrt{\frac{512D}{r_{\mathrm{sh}}}},\ C_{\mathrm{norm}} = \sqrt{C_{\mathrm{fix}}^2 + d},\ \kappa_{\mathrm{act}} = 2\log\frac{32}{r_{\mathrm{sh}}}. \tag{46}$$

RMS normalization has stabilizer one and scalar gain $C_{\mathrm{norm}}/\sqrt{d}$. On the mutable coordinates it is $\mathcal{N}(y) = \lambda(y)y$, where

$$\lambda(y) = \frac{C_{\mathrm{norm}}}{\sqrt{C_{\mathrm{norm}}^2 + \|y\|_2^2}}. \tag{47}$$

For any $z \in \{-1, 1\}^D$, write $Q_{r_{\mathrm{sh}}}(z) = \{x : \|x - z\|_\infty \le r_{\mathrm{sh}}\}$, with the protected coordinate and caches understood to be fixed. Only explicitly specified subsets of these discrete codes are admissible; in particular, the entity and phase coordinates must encode valid one-hot vectors.

Lemma 13 (Smooth implementation of a Boolean stage)**.** *Let $\mathcal{Z}_{\mathrm{leg}} \subseteq \{-1, 1\}^D$ be the legal source codes of one stage. Suppose its required code map Ψ satisfies, for every output coordinate j,*

$$\Psi_j(z) \in \{-1, 1\},\ \Psi_j(z)(a_j^\top z + b_j) \ge 1\ (z \in \mathcal{Z}_{\mathrm{leg}}). \tag{48}$$

With the constants in (46)*, one residual affine–SiLU–affine sublayer implements a map F satisfying*

$$F(Q_{r_{\mathrm{sh}}}(z)) \subseteq Q_{r_{\mathrm{sh}}/16}(\Psi(z))\ (z \in \mathcal{Z}_{\mathrm{leg}}). \tag{49}$$

It uses $4D$ SiLU units. If S is the total number of nonzero coefficients in its threshold rows, its affine maps have $O(D + S)$ nonzero coefficients. Coordinates representing retained Boolean values are refreshed through the rows $a_j = e_j$, $b_j = 0$; a temporary coordinate is reset to false through $a_j = 0$, $b_j = -1$. Every protected output row is zero.

Proof. For $\|y\|_\infty < 3$, the inequality $1 - (1 + t)^{-1/2} \le t/2$ gives

$$\|y - \lambda(y)y\|_\infty \le \frac{D\|y\|_\infty^3}{2C_{\mathrm{norm}}^2} < \frac{27 r_{\mathrm{sh}}}{1024} < \frac{r_{\mathrm{sh}}}{32},\ \lambda(y)^2 > \frac{3}{4}. \tag{50}$$

The second statement follows from $\|y\|_2^2/C_{\mathrm{norm}}^2 < 9r_{\mathrm{sh}}/512 < 1/3$. A source box satisfies $\|x\|_\infty \le 1 + r_{\mathrm{sh}} < 3$. Consequently

$$|a_j^\top \mathcal{N}(x) + b_j - (a_j^\top z + b_j)| \le B\left(r_{\mathrm{sh}} + \frac{r_{\mathrm{sh}}}{32}\right) < \frac{1}{4}. \tag{51}$$

Every actual threshold argument therefore has its required sign and absolute value greater than $3/4$.

Let $f_{\mathrm{S}}(u) = u/(1 + e^{-u})$ and use the actual SiLU expression

$$C_\kappa(u) = \frac{f_{\mathrm{S}}(\kappa(2u + 1)) - f_{\mathrm{S}}(\kappa(2u - 1))}{\kappa} - 1. \tag{52}$$

For $|u| \ge 3/4$ and $\kappa/2 \ge 1$,

$$|C_\kappa(u) - \mathrm{sign}(u)| \le e^{-\kappa/2}. \tag{53}$$

Indeed, replacing SiLU by ReLU in the comparison expression gives exactly $\mathrm{sign}(u)$. The bound $|f_{\mathrm{S}}(v) - \max\{v, 0\}| \le |v|e^{-|v|}$ and monotonicity of $te^{-\kappa t}$ for $t \ge 1/2$ bound the sum of the two errors by $e^{-\kappa/2}$. This comparison does not change the implemented activation. Our choice of κ_{act} makes this error at most $r_{\mathrm{sh}}/32$.

The complete residual formula is

$$F_j(x) = x_j - \mathcal{N}_j(x) + C_{\kappa_{\mathrm{act}}}(a_j^\top \mathcal{N}(x) + b_j). \tag{54}$$

By (50) and (53), its error relative to $\Psi(z)$ is at most $r_{\mathrm{sh}}/16$ in every coordinate. For each bit output, two hidden units implement $C_{\kappa_{\mathrm{act}}}$. For every mutable coordinate, two additional hidden units implement its negative normalized value exactly, using $f_{\mathrm{S}}(u) - f_{\mathrm{S}}(-u) = u$. For output j, the two refresh rows have weights $2\kappa_{\mathrm{act}} a_j^\top$ and biases $\kappa_{\mathrm{act}}(2b_j + 1)$ and $\kappa_{\mathrm{act}}(2b_j - 1)$, with output coefficients $1/\kappa_{\mathrm{act}}$ and $-1/\kappa_{\mathrm{act}}$ and output bias -1. The two cancellation rows have weights $e_j^\top$ and $-e_j^\top$, zero biases, and output coefficients -1 and 1. All other entries are zero. Thus the hidden width is $4D$, with the asserted sparsity. The protected coordinate and fixed caches are unchanged by construction. □

E.6 Attention reads under a shared substitution

Within schema ρ, variable $V_{\rho,j}$ uses shared digit group $b_{j,:}$. In the following formulas, $\beta_{V,e}$ denotes the corresponding coordinate of that group; it does not allocate a separate group to each schema. Let the constant domain have size $c \geq 1$. Each group has c signed coordinates encoding one entity: exactly one coordinate is 1 and all others are -1. The substitution represented by the exact register code is denoted by θ. This notation is used only in the specification and proof; the implemented queries read the register coordinates directly.

Consider a premise occurrence L_i. Let ℓ_i be its number of variable occurrences, counting repetitions. Let A_i be the set of ground atoms with the same predicate and with the constants in L_i in the required positions. For $a \in A_i$, let $a[p]$ denote its entity at argument position p, and define

$$t_{i,a}(\beta) = 1 - \ell_i + \sum_{p:\, L_i[p] \text{ is a variable}} \beta_{L_i[p],a[p]}. \tag{55}$$

At a valid signed one-hot code, $t_{i,a} = 1$ precisely for $a = L_i\theta$, and $t_{i,a} \leq -1$ for every other $a \in A_i$. This also holds when a variable occurs more than once: an inconsistent tuple cannot make all of its corresponding register coordinates equal to one. All occurrences of a variable use the same register.

Allocate one scalar attention head to each pair (i, a) and one temporary coordinate to each premise occurrence i, initialized at signed value -1. Each head includes exactly two positions: the live residual and one fixed null cache entry with key and value zero. Let n be the number of ground atoms, and choose

$$\kappa_{\mathrm{att}} = 2 \log \frac{64n}{r_{\mathrm{sh}}}. \tag{56}$$

Its live key is λ. Its query is $\kappa_{\mathrm{att}} \lambda t_{i,a}(\beta)$: each variable occurrence contributes a query coefficient κ_{att} on its normalized register coordinate, and the protected coordinate has coefficient $\kappa_{\mathrm{att}}(1 - \ell_i)/C_{\mathrm{fix}}$. Its live value is λs_a, where s_a is the current fact coordinate. The output projection sums these heads into temporary coordinate i. Thus the attention residual leaves all other mutable coordinates unchanged and adds

$$p_i = \sum_{a \in A_i} \alpha_{i,a} \lambda s_a, \;\; \alpha_{i,a} = \frac{1}{1 + \exp(-\kappa_{\mathrm{att}} \lambda^2 t_{i,a}(\beta))} \tag{57}$$

to that coordinate. Both positions remain in each original finite-temperature softmax. The live keys and values are recomputed in each block and are not appended to the fixed null caches. No fact is obtained from an immutable value cache: all values, including derived facts, are read from the current residual.

Lemma 14 (Uniform attention read)**.** *Assume that all premise temporary coordinates have center -1 at entry, that each fact and entity coordinate has its prescribed signed center, and that $\ell_i \leq B$. Follow the attention residual* (57) *by a feed-forward residual that assigns premise coordinate i the complete output $y_i - \mathcal{N}_i(y) + C_{\kappa_{\mathrm{act}}}(\mathcal{N}_i(y) + 1)$, refreshes all other retained Boolean coordinates, and resets the remaining temporary coordinates to false. This complete block maps every source box of radius r_{sh} into the radius-$r_{\mathrm{sh}}/16$ box whose premise coordinate i is the signed truth value of $L_i\theta$ and whose other retained Boolean coordinates are unchanged. A false temporary value is implemented by the constant threshold argument -1.*

Proof. The source normalization satisfies (50). Each address test changes from its signed reference value by at most $\ell_i r_{\mathrm{sh}} \leq B r_{\mathrm{sh}} < 1/64$. Hence the target score is greater than $\kappa_{\mathrm{att}}/2$, while every

non-target score is less than $-\kappa_{\mathrm{att}}/2$. With $\epsilon = \exp(-\kappa_{\mathrm{att}}/2) = r_{\mathrm{sh}}/(64n)$, the target head satisfies $1 - \alpha_{i,L_i\theta} \le \epsilon$, and every other head satisfies $\alpha_{i,a} \le \epsilon$.

Let $\sigma_i \in \{-1, 1\}$ be the signed truth value of $L_i\theta$ at the reference state. Since $|s_a| < 2$, the exact attention sum obeys

$$|p_i - \sigma_i| \le r_{\mathrm{sh}} + \frac{r_{\mathrm{sh}}}{32} + 2n\epsilon = \frac{17 r_{\mathrm{sh}}}{16}. \tag{58}$$

The first two terms bound the source fact and normalization errors; the final term bounds both the missing target mass and all non-target contributions. Since the incoming temporary coordinate is within r_{sh} of -1, its post-attention value satisfies $|y_i - (\sigma_i - 1)| \le 33 r_{\mathrm{sh}}/16$. All other coordinates retain magnitude at most $1 + r_{\mathrm{sh}}$. Thus the entire post-attention mutable vector has maximum norm less than three, and its actual feed-forward normalization also satisfies (50). The shifted normalized argument $\mathcal{N}_i(y) + 1$ is within $67 r_{\mathrm{sh}}/32 < 3 r_{\mathrm{sh}}$ of σ_i, so it has the correct sign and absolute value greater than $3/4$. All other retained Boolean coordinates have the same margin. Applying (53) and retaining the residual normalization error proves the radius-$r_{\mathrm{sh}}/16$ conclusion. Constant signed outputs obey the same bound. The proof covers every legal substitution and every fact assignment with the same constants. □

E.7 Proof of Theorem 5 and resource bounds

The seven stages in Appendix E.1 specify all mutable output coordinates. Stage 1 uses Lemma 14; the other six stages use Lemma 13. An inactive attention branch has zero output projection. Each Boolean conjunction of $t \ge 1$ signed inputs uses the affine threshold $\sum_{i=1}^{t} x_i - t + 1$, and each disjunction uses $\sum_{i=1}^{t} x_i + t - 1$. Empty conjunctions and disjunctions use the constant thresholds 1 and -1, respectively. Negated inputs are represented by negative coefficients. Repeated inputs are counted with their multiplicity. These thresholds have absolute margin at least one at every legal signed code.

In Stage 2 a threshold has at most $1 + m + a_{\max}$ inputs. Stage 3 has at most $R + 1$ inputs. In Stage 4 the increment rows have at most $R + v$ inputs and the overflow rows have at most $v + 1$. The remaining stages use disjunctions of at most R inputs or two-input conjunctions and disjunctions. Retained coordinates use the identity threshold and reset coordinates use the constant threshold -1. Thus $B = R + v + m + a_{\max} + 3$ bounds every threshold's absolute coefficient sum, its bias magnitude, and the number of variable occurrences in every premise. The choices in (46) and (56) therefore give one common set of parameters for all stages.

Let $\chi_1(z)$ be the signed code of a canonical macrostep state, with every temporary bit false. Let $\chi_{j+1}(z)$ be the result of the specified Boolean operation at Stage j, retaining all other coordinates. These reference codes describe the proof and are never consulted by the network. The stage lemmas give

$$F_j(Q_{r_{\mathrm{sh}}}(\chi_j(z))) \subseteq Q_{r_{\mathrm{sh}}/16}(\chi_{j+1}(z)) \subseteq Q_{r_{\mathrm{sh}}}(\chi_{j+1}(z)). \tag{59}$$

At the last stage $\chi_8(z) = \chi_1(H(z))$. This proves (12) and every finite repetition by induction. The same fact coordinates and sign thresholds are used at every boundary. Only Stage 3 changes a represented fact, and it performs exactly the selected Horn update; the other stages preserve the represented fact assignment. The phase and binding coordinates are updated by the network, and their decoded one-hot constraints hold on the whole successor boxes. The protected residual coordinate and all null-cache coordinates are fixed exactly. This accounts for the complete native state, with the position in the seven-block cycle a declared discrete execution type.

The counter proof in Appendix E.1 gives coverage of every labeled rule–substitution pair in every G_{sub} macrosteps. Theorem 6 then gives closure after $G_{\mathrm{sub}}(n - |\Gamma|)$ macrosteps, or $7G_{\mathrm{sub}}(n - |\Gamma|)$ native blocks. The bound also covers query absence after complete deduction. For the original terminal projection, apply the same RMS normalization and assign logits $2\lambda s_q$ and $-2\lambda s_q$ to the two answer tokens and zero to every other token. Since $|s_q| \ge 1 - r_{\mathrm{sh}}$ and $\lambda^2 > 3/4$, the winning token exceeds every competing logit by more than one. This includes the zero-horizon case $\Gamma = \mathcal{A}$.

For the resource count, put $N_{\mathrm{prem}} = \sum_\rho m_\rho$. Let $\mathcal{A}_{\mathrm{comp}}(L)$ be the ground atoms compatible with the predicate and constant positions of an atom pattern L, so that $\mathcal{H}_\rho = \mathcal{A}_{\mathrm{comp}}(\mathsf{h}_\rho)$, and set

$$A_{\mathrm{head}} = \sum_\rho |\mathcal{A}_{\mathrm{comp}}(\mathsf{h}_\rho)| \le Rn, \;\; H_0 = \sum_{\rho,i} |\mathcal{A}_{\mathrm{comp}}(B_{\rho,i})| \le nN_{\mathrm{prem}}. \tag{60}$$

The full register list has

$$D = n + 3cv + N_{\text{prem}} + A_{\text{head}} + 4R + v + 1 \tag{61}$$

mutable coordinates and one protected coordinate, so $d = D + 1$. There are H_0 scalar heads in the read block; if $H_0 = 0$, one zero-output head can be used. The concatenated head outputs are mapped to the residual by a rectangular output projection. Zero-output attention branches in the remaining blocks may reuse these head dimensions and cache layouts. At most $14 \max\{1, H_0\}$ scalar null-cache entries suffice for all seven blocks. None contains a fact value or a substitution table.

Each read query has at most $a_{\max}$ variable coefficients and one protected-coordinate coefficient. The key, value, and output projections have bounded support per head. The read block therefore uses $O(H_0(a_{\max} + 1) + D)$ nonzero coefficients. The rule-activation stage uses $O(A_{\text{head}}(m + a_{\max} + 1) + D)$; the fact-write stage uses $O(A_{\text{head}} + D)$. Across the four controller stages, the nontrivial threshold rows use $O(v^2 + Rv + cv + R)$ coefficients, in addition to $O(D)$ coefficients for refreshing retained or resetting temporary coordinates. The SiLU realization adds two nonlinear units per signed output and two units for its exact negative linear term, hence at most $4D$ units per stage. Combining these counts gives

$$W = O\big(Rn(m+1)(a_{\max}+1) + cv + v^2 + Rv\big). \tag{62}$$

The notation W counts independently stored nonzero scalar parameters across the seven shared templates. Indices and the explicit rule and atom descriptions must also be encoded. Their reading cost is included in the displayed upper bound. In particular, the atom set may itself grow exponentially with predicate arity; the saving concerns expansion over rule substitutions, not the storage of all possible facts.

Every internal coefficient or bias has magnitude

$$O\big(B \log(B+1) + B \log(n(B+1)) + \sqrt{B+1}\big), \tag{63}$$

and the protected initialization has magnitude $O(\sqrt{DB})$. To see this, the two SiLU rows for a threshold have coefficients bounded by $2\kappa_{\text{act}}B$ and biases bounded by $\kappa_{\text{act}}(2B+3)$. The cancellation rows have coefficients of magnitude one. The attention rows are bounded by $B\kappa_{\text{att}}$ and $\kappa_{\text{att}}(B + 1)/C_{\text{fix}}$, and the normalization gain is at most $\sqrt{512/r_{\text{sh}} + 1}$. All parameters are independent of the macrostep horizon and are represented by the stated square roots and logarithms.

A verifier checks the sparse rule–atom incidence lists and threshold rows, the exact zero cache entries, the normalization parameters, and the vocabulary projection. Given dense matrices, this requires $O(W + d^2 + H_{\text{att}}d + Vd)$ entry and real-arithmetic checks, where $H_{\text{att}} = \max\{1, H_0\}$. The uniform inequalities in the two implementation lemmas apply to every legal code, so neither fact assignments nor substitutions need to be enumerated for this structural certificate. This verification cost applies to the specified family with its declared coordinate interpretation; discovering such an interpretation in an arbitrary checkpoint is outside its scope. The count treats the displayed elementary functions as real-arithmetic primitives.

Sparse native execution costs $O(W)$ per macrostep, including normalization and both softmax positions in every read head, and hence $O(T_{\text{mac}}W)$ for T_{mac} macrosteps. Describing a horizon takes $O(\log(T_{\text{mac}} + 1))$ bits. An independently stored expansion into $7T_{\text{mac}}$ native blocks has the corresponding linear storage and inspection cost; the stated savings require the shared-template representation. A program with no rules needs only the fact initialization and the final readout.

E.8 A RECURSIVE REACHABILITY PROGRAM

Consider two binary predicates and the rules

$$E(x, y) \longrightarrow P(x, y),\ P(x, y) \wedge E(y, z) \longrightarrow P(x, z). \tag{64}$$

On c constants, $n = 2c^2$, $R = 2$, $N_{\text{prem}} = 3$, $m = 2$, $v = 3$, $a_{\max} = 2$, $A_{\text{head}} = 2c^2$, and $H_0 = 3c^2$. Thus $D = 4c^2 + 9c + 15$, $G_{\text{sub}} = c^2 + c^3$, and $W = O(c^2)$. The radius is $1/832$ for every c, although width and parameter values may change with c. The model tests the second rule using the current P coordinates, so it can use a P fact established by either rule at an earlier macrostep. For example, with distinct constants u, w, z and initial facts $E(u, w), E(w, z)$, the first rule establishes $P(u, w)$ and the second can subsequently establish $P(u, z)$. No initial cache entry contains either derived P fact. This family exhibits sharing across c^3 substitutions of the recursive rule with only quadratically many stored parameters. It is an explicit upper bound and does not assert a lower bound for other encodings or architectures.

E.9 Comparison with corrected discrete simulations

Correcting intermediate representations is already central to the constructions of Giannou et al. (2023) and Wei et al. (2022). The former implement a SUBLEQ interpreter with ten layers and two heads (their Lemma 4), encode instructions and working memory in the input sequence, and correct finite-softmax read and write errors with ReLU (their Section 5.1). Their architecture does not include RMS normalization. The latter prove an all-layer margin result for corrected computations, including layerwise parameter sharing (their Theorem D.6). These results already establish repeated control and robust discrete simulation. They do not, in their stated forms, give the present relational register layout or its RMS–SiLU constants. Conversely, our theorem does not provide their input-programmable universality or their statistical guarantees.

A standard perturbation argument explains why smooth correction alone is not the main distinction. Let F_0 be a discrete-state implementation and F_1 another implementation on the same complete-state domain, with fixed coordinates preserved exactly. If, for every represented abstract state z,

$$F_0(\mathcal{Q}_r(z)) \subseteq \mathcal{Q}_{r-\eta_{\mathrm{tr}}}(H(z)),\ 0 < \eta_{\mathrm{tr}} < r, \quad \sup_{x \in \bigcup_z \mathcal{Q}_r(z)} \|F_1(x) - F_0(x)\|_\infty \le \eta_{\mathrm{tr}}, \tag{65}$$

where the norm is on mutable coordinates, the triangle inequality gives $F_1(\mathcal{Q}_r(z)) \subseteq \mathcal{Q}_r(H(z))$. A fixed approximation bound therefore suffices for every finite repetition. Applying this observation to a published simulator would still require a common input interface, a verified strict margin, and resource bounds for the chosen normalized smooth implementation; the general argument supplies none of those quantities by itself.

For a direct comparison, we consider a matched ReLU variant of the seven-stage architecture with identical caches, threshold rows, normalization, and parameter values, replacing only SiLU by ReLU. Let F_j^{R} and F_j^{S} denote the corresponding ReLU and SiLU maps at Stage j, respectively. With the source and target centers of (59), each stage satisfies

$$\begin{aligned} F_j^{\mathrm{R}}(Q_{r_{\mathrm{sh}}}(\chi_j(z))) &\subseteq Q_{r_{\mathrm{sh}}/32}(\chi_{j+1}(z)), \\ \sup_{x \in Q_{r_{\mathrm{sh}}}(\chi_j(z))} \|F_j^{\mathrm{S}}(x) - F_j^{\mathrm{R}}(x)\|_\infty &\le r_{\mathrm{sh}}/32, \\ F_j^{\mathrm{S}}(Q_{r_{\mathrm{sh}}}(\chi_j(z))) &\subseteq Q_{r_{\mathrm{sh}}/16}(\chi_{j+1}(z)). \end{aligned} \tag{66}$$

Indeed, both activations satisfy $f(u) - f(-u) = u$, so the paired cancellation units implement exactly the same linear term. On every admitted threshold input, the ReLU correction equals the target sign exactly. The only remaining ReLU error is the normalization residual, less than $r_{\mathrm{sh}}/32$ by (50). The SiLU correction adds at most $e^{-\kappa_{\mathrm{act}}/2} = r_{\mathrm{sh}}/32$ by (53). The attention operation is identical in both constructions, and Lemma 14 establishes the required shifted threshold margin after that operation. This proves all three bounds, including Stage 1, without comparing only unperturbed codes.

Both constructions therefore use seven stages, $4D$ nonlinear units per stage, the same head count, and the same nonzero-parameter order in (62). With $\kappa_{\mathrm{act}} = 2\log(32/r_{\mathrm{sh}})$, $\kappa_{\mathrm{att}} = 2\log(64n/r_{\mathrm{sh}})$, and $C_{\mathrm{fix}} = \sqrt{512D/r_{\mathrm{sh}}}$, neither the radius nor the selected parameters depend on the number of scans. The SiLU inclusion also leaves an output-error budget: an additional perturbation of maximum norm at most $15r_{\mathrm{sh}}/16$ in the mutable coordinates after each completed native block still leaves its state in the required radius-r_{sh} box. This follows by another triangle inequality and induction, provided protected coordinates and caches remain exact. It is not a floating-point certificate unless all numerical errors of that implementation have independently been bounded by this budget.

For the reachability program, the shared representation has $D = 4c^2 + 9c + 15$ mutable coordinates, $H_0 = 3c^2$ read heads, $W = O(c^2)$ stored nonzero parameters, and $r_{\mathrm{sh}} = 1/832$, while one complete substitution scan still requires $G_{\mathrm{sub}} = c^2 + c^3$ macrosteps. Thus the construction avoids explicitly storing the G_{sub} labeled ground instances while retaining the corresponding enumeration cost. This comparison concerns the stated relational interface and does not imply a cubic storage lower bound for alternative Transformer or logic-program implementations; sequence memory in the SUBLEQ construction and residual width here measure different resources.

F A FULL RESIDUAL-AND-CACHE REALIZATION

F.1 A TWELVE-ATOM INSTANCE

We now show that the conditions are jointly satisfiable by a fully specified family of residual and cache operations. The instance is specified directly on a domain of internal activations. Its input interface consists of declared initial residuals and fixed *prefix caches* containing key–value entries available before the scan. All subsequent operations through the final vocabulary decision are verified. It is not a trained model, and it makes no claim about how a prompt would produce these states. Appendix F specifies every matrix through its coordinate action and proves the bounds below.

Use constants $\mathtt{a}, \mathtt{b}$ and the program

$$\mathsf{P}(X) \wedge \mathsf{R}(X,Y) \to \mathsf{Q}(Y),\ \mathsf{Q}(Y) \wedge \mathsf{S}(Y) \to \mathsf{Z}(Y). \tag{67}$$

There are twelve ground atoms and six ground rules. A fixed scan tests the four instances of the first rule and the two of the second. The current residual holds twelve fact coordinates, two scratch coordinates, protected entity registers, and one protected normalization coordinate. The prefix cache holds the eight ground input atoms of $\mathsf{P}, \mathsf{R}, \mathsf{S}$, including false ones with signed value -1. Derived facts are stored in the residual. Every cached input bit is fixed throughout the scan and agrees with the corresponding input-fact bit in the current residual.

All layers use RMS normalization with a positive additive constant, called the *stabilizer*, in its mean-square term. With its scalar gain, each mutable coordinate normalizes as $y_i \mapsto \lambda(y) y_i$ with

$$\lambda(y) = \frac{C_{\mathrm{norm}}}{\sqrt{C_{\mathrm{norm}}^2 + \|y\|_2^2}},\ C_{\mathrm{norm}} \geq 128. \tag{68}$$

This is the exact normalization factor for the declared residual. Two finite-temperature attention heads write premise values into the scratch coordinates; shared entity reads determine their addresses. With attention score scaling 32 and all eight cache positions included, the target score gap is at least 31 on the certified domain.

With f_{S} the original SiLU activation, define

$$C_{\mathrm{act}}(u) = \frac{f_{\mathrm{S}}(32(2u+1)) - f_{\mathrm{S}}(32(2u-1))}{32} - 1. \tag{69}$$

A feed-forward block refreshes every non-head fact through C_{act}. The head is updated by applying the same function to a test that preserves an already established head. For the first rule the preactivation is $2\widetilde{s}_{\mathsf{Q}(Y)} + \widetilde{p}_1 + \widetilde{p}_2 + 1$; for the second it is $2\widetilde{s}_{\mathsf{Z}(Y)} + \widetilde{s}_{\mathsf{Q}(Y)} + \widetilde{p}_1 + 1$, where tildes denote normalized coordinates after attention. With signed Boolean inputs the test is at least one exactly when the old head or the complete body is true, and at most minus one otherwise. The residual correction for fact a is $-\widetilde{s}_a + C_{\mathrm{act}}(u_a)$, and each scratch correction is its negative normalized coordinate. The linear terms are implemented exactly by the identity $f_{\mathrm{S}}(u) - f_{\mathrm{S}}(-u) = u$, so no nonlinear unit is replaced by a Boolean gate. Protected rows have zero writes by construction, and every remaining branch is included.

Theorem 15 (Verified repeated execution)**.** *For every one of the 2^{12} initial fact assignments, initialize the fact centers in the current residual at $2k_a - 1$, the scratch centers at zero, and the cache consistently. On mutable boxes of radius $1/32$, the blocks above satisfy*

$$\|F_t(\chi(k)) - \chi(\mathsf{U}_{g_t}(k))\|_\infty \leq \frac{1}{1024},\ \|DF_t\|_{\infty\to\infty} \leq \frac{1}{32}, \tag{70}$$

where differentiation is in the mutable coordinates and all fixed coordinates are preserved exactly. These boxes are therefore invariant under the logical successors. The first six-step scan reaches closure for every input, and all 72 updates preserve the required logical state. The native final vocabulary projection gives the correct query decision with margin at least one against every competing token.

The maximum required proof depth of this program is two: all first-rule instances precede all second-rule instances, so one six-step scan reaches closure and the remaining 66 updates certify its persistence. The construction does not rely on unbounded accumulation of positive writes. Refreshing

Table 1: Uniform bounds for the instance. Every bound holds on all declared source boxes, not only at the reference states.

Quantity	Certified bound
Included positions and target gap	$N_{\mathrm{slot}} = 8$, gap at least 31
Signed premise-read error	$14e^{-31}/(1 + 7e^{-31}) < 1/1024$
Actual normalized-coordinate error	$\delta_{\mathrm{norm}} < 1/2048$
Refresh error for $\lvert u\rvert \geq 3/4$	$\lvert C_{\mathrm{act}}(u) - \mathrm{sign}(u)\rvert < 1/65536$
Error at reference states, all mutable coordinates	$e_{\mathrm{uni}} \leq 1/1024$
Complete mutable Jacobian norm	$L_{\mathrm{uni}} \leq 1/32$
Invariant radius and fact margin	$r_{\mathrm{uni}} = 1/32$, margin $31/32$
Original full-vocabulary margin	$\gamma_{\mathrm{out}} \geq 1$

an established fact keeps it near its reference state, and the head test preserves a head that is already true. The invariant estimate controls normalized reads, finite attention leakage, negative SiLU outputs, and scratch contents at once. For the initialization $\{\mathsf{P}(\mathtt{a}), \mathsf{R}(\mathtt{a}, \mathtt{b}), \mathsf{S}(\mathtt{b})\}$ the scan first establishes $\mathsf{Q}(\mathtt{b})$, and a later block reads that new fact to establish $\mathsf{Z}(\mathtt{b})$. Removing $\mathsf{R}(\mathtt{a}, \mathtt{b})$ from the initialization changes the final decision.

Table 1 summarizes the calculation. The radius check is $1/1024 + (1/32)(1/32) = 1/512 < 1/32$. It covers the entire mutable state after each attention and feed-forward block, while exact zero-write identities cover the protected fields and the fixed caches. The strict margin in this inequality is not obtained by resetting the state between blocks; the actual residual evolution produces the next box of the same family.

The reference-state calculation depends on only a bounded number of fact bits. For each of the six templates the head depends on three bits, eleven retained facts depend on one bit each, and two scratch coordinates depend on one bit each, which gives $6(8 + 22 + 4) = 204$ scalar local evaluations. The softmax contribution from non-target positions is propagated through the downstream operations as a certified remainder. The normalization and derivative bounds are separate analytic checks, so this count is not a running time.

F.2 Signature, schedule, and complete initialization

Use the constants $\{\mathtt{a}, \mathtt{b}\}$ and predicates $\mathsf{P}, \mathsf{Q}, \mathsf{S}, \mathsf{Z}$ of arity one and R of arity two. Thus there are twelve atoms. The six ground instances of (67) are scanned in this order: the first rule with $(X, Y) = (\mathtt{a}, \mathtt{a}), (\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{b})$, followed by the second rule with $Y = \mathtt{a}, \mathtt{b}$. Repeating this list twelve times gives 72 fixed blocks. A block is one attention residual sublayer and one feed-forward residual sublayer, each applying normalization before its attention or feed-forward operation.

The current residual consists of $s \in \mathbb{R}^{12}$, scratch coordinates z_1^{scr}, z_2^{scr}, two protected two-dimensional entity registers fixed at $(1, 0)$ and $(0, 1)$, and one protected scalar fixed at 128. Its width is $d_{\mathrm{live}} = 19$. The fact score for atom a is exactly s_a with threshold zero. The binding reads select the same protected entity vector at all occurrences of a variable in a given ground instance. The query maps change with the declared ground-rule template, not with a hidden choice based on the answer.

Each attention head has all eight ground input atoms of $\mathsf{P}, \mathsf{R}, \mathsf{S}$ in its fixed prefix cache. Predicate, first argument, and second argument occupy three mutually orthogonal three-dimensional address blocks. Predicate labels are the standard basis for $\mathsf{P}, \mathsf{R}, \mathsf{S}$. Argument labels are the standard basis for $\mathtt{a}, \mathtt{b}$, and padding. Every key is its nine-dimensional address. Every value has first coordinate $2k_a - 1$ and eight zero coordinates. The prefix cache is part of the declared input interface. All other cached fields are fixed explicitly, and the prefix-only mask includes exactly these eight positions.

For an initial fact assignment k, set $s = 2k - \mathbf{1}$ and both scratch coordinates to zero. Store its eight input-predicate signs in the prefix values. The cache center is unchanged at every successor because no rule has an input predicate as its head. The $\mathsf{P}, \mathsf{R}, \mathsf{S}$ coordinates in the current residual may move within their intervals, but their signs remain equal to their unchanged cache values. Initial Q and Z facts are allowed, as in the general Horn specification. The initialization interface is a residual-and-cache state with these contents.

F.3 Exact normalization and attention operations

Use RMS normalization with stabilizer $\varepsilon_{\mathrm{norm}} = 1$ and scalar gain

$$G_{\mathrm{scalar}} = C_{\mathrm{norm}}/\sqrt{19},\ C_{\mathrm{norm}}^2 = 128^2 + 2 + 19 = 16405. \tag{71}$$

For a current residual whose fourteen mutable coordinates are y, the squared norm of its protected part is $128^2 + 2$. Consequently its actual normalized mutable coordinates are exactly $\lambda(y)y$, with λ given by (68). The protected coordinates are also multiplied by λ during a read, but every attention and feed-forward output row into them is zero. The original residual state therefore preserves them, rather than resetting them after each block.

The query projection reads the protected entity registers and the scalar coordinate to produce $96\lambda\,\mathrm{addr}(a_*)$ for the intended target atom. The scalar read divided by 128 supplies any predicate or padding constant, with the same factor λ. Repeated occurrences of a variable use the same protected entity read. Keys equal $\mathrm{addr}(a)$, and standard dot-product scaling by $\sqrt{9} = 3$ gives scores

$$\ell_a = 32\lambda\langle \mathrm{addr}(a_*), \mathrm{addr}(a)\rangle. \tag{72}$$

There are no positional-score terms in this specified family. The two attention output projections write their scalar value averages into separate scratch coordinates and into no fact or protected coordinate. This defines the attention weights and output matrices completely through their coordinate actions; unmentioned entries are zero.

For a first-rule block, the target positions are $\mathsf{P}(X)$ and $\mathsf{R}(X, Y)$. For a second-rule block, the first target is $\mathsf{S}(Y)$ and the second is $\mathsf{P}(\mathtt{a})$. The latter head remains a genuine nonzero native computation; its scratch value is cleared and not used in the rule gate. The other premise $\mathsf{Q}(Y)$ is read from its fact coordinate in the current residual. Its fixed row and the $\mathsf{S}(Y)$ address use the same ground instance. Thus a derived fact is actually reused and is not expected to appear in an unchanged prefix cache.

The attention residual maps the mutable state $x^{\mathrm{mut}} = (s, z_1^{\mathrm{scr}}, z_2^{\mathrm{scr}})$ to y, leaving s unchanged and setting $y_{12+i} = z_i^{\mathrm{scr}} + p_i^{\mathrm{att}}$ for the two softmax averages. Denote this map by $\mathcal{A}_t^{\mathrm{res}}$ solely within this construction. Each average is a convex combination of signs, so $|p_i^{\mathrm{att}}| \le 1$ for all input fact assignments.

F.4 Exact feed-forward matrices and head updates

Let $\widetilde{y} = \lambda(y)y$. The feed-forward preactivation for its head is

$$u_{\mathsf{h}_g} = \begin{cases} 2\widetilde{y}_{\mathsf{Q}(Y)} + \widetilde{y}_{13} + \widetilde{y}_{14} + 1, & g \text{ is a first-rule instance,} \\ 2\widetilde{y}_{\mathsf{Z}(Y)} + \widetilde{y}_{\mathsf{Q}(Y)} + \widetilde{y}_{13} + 1, & g \text{ is a second-rule instance.} \end{cases} \tag{73}$$

For a non-head fact, put $u_a = \widetilde{y}_a$. The actual feed-forward output into fact coordinate a is

$$-\widetilde{y}_a + \frac{f_{\mathrm{S}}(32(2u_a + 1)) - f_{\mathrm{S}}(32(2u_a - 1))}{32} - 1,\ f_{\mathrm{S}}(v) = \frac{v}{1 + e^{-v}}. \tag{74}$$

Its output into each scratch coordinate j is $-\widetilde{y}_j$. The output into protected coordinates is zero. The residual adds these outputs to y.

These formulas specify an ordinary affine–SiLU–affine feed-forward map. Each $C_{\mathrm{act}}(u_a)$ uses two hidden rows with preactivations $32(2u_a + 1)$ and $32(2u_a - 1)$ and output coefficients $1/32$ and $-1/32$ into row a, with output bias -1. For the negative linear term, two additional hidden rows have preactivations $\widetilde{y}_j$ and $-\widetilde{y}_j$ with output coefficients -1 and 1 into row j. Their sum is exactly $-\widetilde{y}_j$, because

$$f_{\mathrm{S}}(v) - f_{\mathrm{S}}(-v) = v\left(\frac{1}{1 + e^{-v}} + \frac{1}{1 + e^{v}}\right) = v. \tag{75}$$

These fixed affine–SiLU–affine layers therefore realize the construction entirely within the declared native state, with all biases and nonzero connections specified above.

With exact signed inputs $\sigma_{\mathrm{head}}, \sigma_1, \sigma_2 \in \{-1, 1\}$, the quantity $2\sigma_{\mathrm{head}} + \sigma_1 + \sigma_2 + 1$ is at least one when the old head is true or both premises are true. If neither condition holds, $\sigma_{\mathrm{head}} = -1$ and at least one premise is -1, so it is at most minus one. This covers a head that is already true and avoids repeated loss from a disabled smooth gate. The desired successor is the Horn update, not an overwrite by the body truth value alone.

F.5 UNIFORM QUANTITATIVE BOUNDS

All estimates below hold for every fact assignment and every mutable source coordinate within $1/32$ of its center. Protected coordinates and prefix caches have radius zero. At the source, fact magnitudes are at most $33/32$ and scratch magnitudes at most $1/32$. Immediately after attention every mutable magnitude is at most $33/32$. Hence

$$\|y\|_2^2 \le 14(33/32)^2 < 15,\ 1-\lambda(y) \le \frac{\|y\|_2^2}{2C_{\mathrm{norm}}^2}. \tag{76}$$

The second inequality follows from $(1+t)^{-1/2} \ge 1 - t/2$ for $t \ge 0$, proved by differentiation or convexity. Therefore

$$\|y - \lambda(y)y\|_\infty < \frac{1}{2048},\ \lambda(y) > \frac{31}{32}. \tag{77}$$

The same bounds apply before attention. Both the original normalization divisor and its positive stabilizer remain in C_{norm}.

Every target address has squared norm three, and every distinct cache address differs in at least one orthogonal block. The target score gap is therefore at least $32\lambda > 31$. The value diameter is two. Thus every attention premise read obeys

$$|p_i^{\mathrm{att}} - \sigma_{a_*}| \le \frac{14e^{-31}}{1+7e^{-31}} < \frac{1}{1024}. \tag{78}$$

The last strict inequality already follows from $e > 2$. This bound includes every false and unrelated position in the denominator. It is compatible with Theorem 1; here the exact common query scaling also gives the gap directly.

The normalized coordinate of a fact in the current residual differs from its signed bit by at most $1/32 + 1/2048$. A normalized scratch premise differs from its intended signed bit by at most $1/32 + 1/1024 + 1/2048$. Each head test consequently differs from its ideal signed test by at most

$$\frac{4}{32} + \frac{2}{1024} + \frac{4}{2048} < \frac{1}{4}. \tag{79}$$

This common bound covers both templates. Every head preactivation thus has magnitude at least $3/4$ and the correct successor sign. Non-head preactivations also have the correct sign and magnitude at least $3/4$.

For SiLU and any real v,

$$|f_{\mathrm{S}}(v) - \max\{v, 0\}| = \frac{|v|}{1+e^{|v|}} \le |v|e^{-|v|}. \tag{80}$$

The ReLU expression corresponding to (69) equals the sign of u when $|u| \ge 1/2$. When $|u| \ge 3/4$, both $|2u+1|$ and $|2u-1|$ are at least $1/2$. Since ve^{-32v} is decreasing for $v \ge 1/2$, the actual SiLU expression satisfies

$$|C_{\mathrm{act}}(u) - \mathrm{sign}(u)| \le e^{-16} < \frac{1}{65536}. \tag{81}$$

This comparison is a proof bound only; all native activations in the construction remain SiLU. Equations (77) and (81) place every successor fact within $33/65536$ of its correct signed center and every successor scratch coordinate within $1/2048$ of zero. This already proves direct successor inclusion on the entire box and bounds the error at reference states by $1/1024$.

F.6 A DERIVATIVE CERTIFICATE ON THE SAME BOXES

For completeness, we also verify the derivative bound used by Theorem 15, thereby establishing the stated comparison bounds independently. Write $\mathcal{N}_{\mathrm{eff}}(y) = \lambda(y)y$ for the actual normalized mutable coordinates. On the post-attention domain,

$$\|D\lambda(y)\|_1 \le \frac{\|y\|_1}{C_{\mathrm{norm}}^2} < \frac{1}{1024},\ \|D\mathcal{N}_{\mathrm{eff}}(y) - I_{14}\|_{\infty\to\infty} < \frac{1}{512}. \tag{82}$$

Indeed, $\partial_j \lambda = -\lambda^3 y_j / C_{\mathrm{norm}}^2$. The second bound is at most $(1-\lambda) + \|y\|_\infty \|D\lambda\|_1$, which is less than $45/32768 < 1/512$ by the common squared-norm bound. In particular $\|D\mathcal{N}_{\mathrm{eff}}\|_{\infty\to\infty} < 2$.

For a softmax average with scores $32\lambda d_j$, where $0 \le d_j \le 3$ and scalar values have magnitude at most one, differentiating with respect to λ gives $32 \sum_j a_j v_j (d_j - \sum_i a_i d_i)$. Its magnitude is at most 192; this loose bound is sufficient. Combining it with (82) gives $\|Dp_i^{\mathrm{att}}\|_1 < 1/4$ with respect to source mutable coordinates. Therefore

$$\|D\mathcal{A}_t^{\mathrm{res}}\|_{\infty\to\infty} \le \frac{5}{4}. \tag{83}$$

All cache values and keys are fixed on the source box, as specified by their zero radii.

For $v \neq 0$, direct differentiation of SiLU gives

$$|f_{\mathrm{S}}'(v) - \mathbf{1}\{v > 0\}| \le (1 + |v|)e^{-|v|}. \tag{84}$$

When $|u| \ge 3/4$, the ideal derivatives of the two terms in C_{act} cancel and both scaled arguments have magnitude at least sixteen. Consequently

$$|C_{\mathrm{act}}'(u)| \le 68e^{-16} < \frac{1}{512}. \tag{85}$$

The affine input to each head test has coefficient absolute sum four; a retained fact has coefficient sum one. For the post-attention feed-forward residual map $\mathcal{G}_t^{\mathrm{res}}$, equations (82) and (85) yield

$$\|D\mathcal{G}_t^{\mathrm{res}}\|_{\infty\to\infty} \le \frac{1}{512} + \frac{4 \cdot 2}{512} = \frac{9}{512}. \tag{86}$$

Scratch rows satisfy the smaller bound $1/512$. The margin calculation above holds throughout each source box, so the activation derivative bound applies at every relevant post-attention point. Composing the two original sublayers gives

$$\|DF_t\|_{\infty\to\infty} \le \frac{5}{4}\frac{9}{512} = \frac{45}{2048} < \frac{1}{32}. \tag{87}$$

Differentiation here is in the fourteen mutable coordinates. Protected registers in the current residual and all immutable caches are preserved exactly, as required for the zero-radius reduction.

F.7 Proof of Theorem 15 and the terminal decision

Proof. The preceding calculations give an error bound at reference states of $e_{\mathrm{uni}} = 1/1024$ and derivative bound $L_{\mathrm{uni}} = 1/32$ on every outer mutable box of radius $r_{\mathrm{uni}} = 1/32$. Their scalar radius condition is

$$\frac{1}{1024} + \frac{1}{32}\frac{1}{32} = \frac{1}{512} < \frac{1}{32}. \tag{88}$$

All fact scores are the corresponding coordinates in the current residual, so the boxes have fact margin $31/32$. The protected rows and immutable cache fields remain exactly at their prescribed centers. The scalar comparison proof in Appendix D therefore applies to the complete state. Shared entity identities and uniform attention gaps persist, scratch values stay in the declared intervals, and each step performs exactly its Horn update.

Every six consecutive blocks contain every ground rule. Since there are twelve ground atoms, Theorem 6 makes 72 blocks sufficient for every initialization, including arbitrary initial derived facts. In fact, no rule writes to $\mathsf{P}, \mathsf{R}, \mathsf{S}$. The first four tests therefore establish all possible Q facts, and the final two establish all possible Z facts. Thus the first scan is already closed for every initialization; subsequent tests preserve this closure. The fixed schedule is an actual sequence of layer templates; it is not an external replacement for a hidden controller.

Use the same final RMS normalizer and a vocabulary projection with logits λs_q for y_1, $-\lambda s_q$ for y_0, and bias -2 with zero weight for every other vocabulary token. On the invariant domain, $|s_q| \ge 31/32$ and $\lambda > 31/32$. The correct token exceeds the opposite answer by at least $2(31/32)^2 > 1$ and every remaining competitor by more than two. Log-probability differences equal these logit differences. Thus the original greedy answer has full-vocabulary margin at least one. □

For the stated nontrivial initialization, the ground instance $(X, Y) = (\mathtt{a}, \mathtt{b})$ first establishes $\mathsf{Q}(\mathtt{b})$. The final instance of the same scan then reads its coordinate in the current residual with the cached $\mathsf{S}(\mathtt{b})$ value and establishes $\mathsf{Z}(\mathtt{b})$. The query is not initially present. Deleting $\mathsf{R}(\mathtt{a}, \mathtt{b})$ from that initialization prevents every derivation of this query, because all other input atoms are absent. Under an interface allowing replacement of the complete initial residual and the corresponding cache value, this deletion is realizable and changes the decision. An interface forbidding that cache replacement would not implement the same edit; the construction makes the intervention permission explicit.

G Logical soundness, progress, and output correctness

G.1 Finite closure and its rule semantics

For $S \subseteq \mathcal{A}$, define the simultaneous consequence operator

$$\mathcal{T}_{\mathcal{R}}(S) = S \cup \{\mathsf{h}_g : g \in \mathcal{G},\ \mathsf{B}_g \subseteq S\}. \tag{89}$$

It is monotone and extensive. Its iterates from Γ form an increasing sequence of subsets of a finite set. Every strictly increasing iteration adds at least one new atom, so after at most $n_{\mathrm{at}} - |\Gamma|$ iterations it is fixed. The fixed set is closed under every ground rule. Every closed set containing Γ contains all these iterates by induction. The fixed set is therefore exactly $\mathrm{Cl}_{\mathcal{R}}(\Gamma)$.

Equivalently, a Herbrand interpretation of the finite relational vocabulary is the set of ground atoms it makes true. It satisfies all facts and rules when it contains Γ and contains h_g whenever it contains B_g. Every such interpretation contains the least closure. Conversely the closure itself is such an interpretation. Hence a ground atom is true in every interpretation satisfying the facts and rules exactly when it belongs to the closure. Thus membership in the least closure coincides with logical consequence under the finite definite-Horn semantics used here.

G.2 Proof of Theorem 6

Proof. The initial extracted facts are Γ by hypothesis. Applying the first inclusion in (3) to x_t places x_{t+1} in the corresponding successor cell and therefore gives $\alpha_{t+1}(x_{t+1}) = H_t(\alpha_t(x_t))$. Induction proves the displayed abstract/native trajectory identity.

Every fact update is extensive, so the initial facts remain present. Suppose the current fact set is contained in $\mathrm{Cl}_{\mathcal{R}}(\Gamma)$. An idle step adds nothing. A selected rule with a false body also adds nothing. If its body is true, all its premises belong to the closure, and closure under that ground rule places its head there as well. Induction proves the two inclusions in (13).

For the proof trace, treat each initial fact as a leaf of the derivation. Whenever a head is added for the first time, record the selected ground rule and connect its previously established premises to that head. Each premise was available before this update, so each edge points to an earlier fact node. The resulting finite directed graph is acyclic. Empty-body rules add nodes without premise edges. Induction over these nodes proves a Horn derivation of each added atom. No record node is needed when a rule merely retains an existing head.

For completeness, let $K^{(j)} = \mathcal{T}_{\mathcal{R}}^{j}(\Gamma)$. Consider a window of B_{cov} consecutive steps beginning with fact set K^{start}. Every rule with body contained in K^{start} is selected at least once during the window. Monotonicity preserves all its body facts until that selection, and (1) therefore ensures its head is present by the window's end. Thus the endpoint contains $\mathcal{T}_{\mathcal{R}}(K^{\mathrm{start}})$.

Induction over disjoint complete windows gives $K^{(j)} \subseteq \mathsf{K}(k_{jB_{\mathrm{cov}}})$ whenever that boundary exists. After $n_{\mathrm{at}} - |\Gamma|$ such windows, $K^{(n_{\mathrm{at}}-|\Gamma|)} = \mathrm{Cl}_{\mathcal{R}}(\Gamma)$. Soundness gives the reverse inclusion, so the represented fact set equals the closure. Later steps preserve that equality because no rule can add anything outside closure. The terminal decision obligation then makes the original output y_1 exactly for an entailed query, and y_0 otherwise. □

When the ground rule set is empty, every high-level step is idle and closure equals Γ from the start. The coverage condition is vacuous, and any positive B_{cov} can be used; the stated horizon remains sufficient but is not needed. If $\Gamma = \mathcal{A}$, the bound is zero and the fact set is already closed. For a family of fact sets, a uniform bound is $B_{\mathrm{cov}} n_{\mathrm{at}}$, though the sharper instance-specific count is valid

when the actual initial cardinality is known. The coverage condition is required along the runs being certified; it is not deduced from the finite number of rules.

G.3 Why soundness does not prove a negative answer

Take the program $\mathsf{P} \to \mathsf{Q}$ with input $\{\mathsf{P}\}$. A controller that selects only idle steps has a legal, sound trace and never inserts an unsupported fact. Its terminal query bit for Q is zero. If its output follows that bit, it answers the entailment question incorrectly. This satisfies soundness and native/abstract faithfulness in an appropriate source system, but not bounded rule coverage. It is a counterexample to deriving completeness from legal transitions alone.

Conversely, once q has been added, soundness already supplies a proof of the positive answer; saturation is unnecessary for that particular query. Early stopping can therefore be justified for a positive query when native decoding and terminal readout are also certified. Early stopping after failing to find a proof requires a separate progress or nonderivability argument. A completed rule scan with no new addition witnesses closure, provided every rule was actually tested and the represented fact set was not modified by unreported operations during the scan.

H Intervention implementation and proof support

There is a concrete feasibility test for affine scores. At a fixed source let permitted controls be $x \mapsto x + L_{\mathrm{edit}} h_{\mathrm{edit}}$, subject to a supplied polyhedral domain, polyhedral bounds on h_{edit}, and linear equations holding named nonedited registers fixed. Stack fact and controller scores as $M_{\mathrm{code}} x + b^{\mathrm{code}}$. For a fully specified target bit code b^{tgt}, an implementation with margin $\gamma_{\mathrm{edit}} > 0$ exists exactly when the resulting linear constraints, including

$$\mathrm{diag}(2b^{\mathrm{tgt}} - \mathbf{1})\big(M_{\mathrm{code}}(x + L_{\mathrm{edit}} h_{\mathrm{edit}}) + b^{\mathrm{code}}\big) \geq \gamma_{\mathrm{edit}} \mathbf{1}, \tag{90}$$

are feasible. This test applies to the declared affine intervention class at a fixed source state. A single intervention over a domain requires the corresponding uniform implementation identity. Appendix H proves the test and gives a finite-precision alternative.

H.1 Proof support determines which premise deletions matter

The logical program makes a more precise causal prediction than an isolated proof trace. Define the minimal premise supports

$$\mathcal{U}_q(\Gamma) = \min_{\subseteq} \{S \subseteq \Gamma : q \in \mathrm{Cl}_{\mathcal{R}}(S)\}, \tag{91}$$

where $\min_{\subseteq}$ retains the inclusion-minimal sets, not only those of smallest cardinality. An empty support is possible when rules derive the query without input facts.

Proposition 16 (Premise deletion and complete deduction)**.** *Suppose the query follows from Γ, and both the original run and the run initialized with $\Gamma \setminus \Delta_\Gamma$ satisfy Theorem 6, including completeness and the terminal decision obligation. Their decisions differ exactly when*

$$S \cap \Delta_\Gamma \neq \varnothing \text{ for every } S \in \mathcal{U}_q(\Gamma). \tag{92}$$

The same conclusion holds for a permitted initial-state edit implementing that complete initialization change. For an explicitly listed ground Horn program, deciding whether such a deletion exists within a supplied cardinality bound is NP-complete, already when every rule has the query as head and contains only input atoms in its body.

The characterization follows from monotonicity and the existence of a minimal support inside any sufficient finite fact set. The complexity statement is a direct reduction from hitting set, the incidence dual of set cover (Karp, 1972); Appendix H gives both proofs. It concerns optimization over the supplied logical program, not the cost of verifying an arbitrary checkpoint. A single observed derivation provides only one sufficient support. Deleting a premise from it need not destroy all other supports, whereas an empty support makes every input-fact deletion ineffective.

For example, with $\mathsf{P} \to \mathsf{Q}$ and $\mathsf{R} \to \mathsf{Q}$ and input $\{\mathsf{P}, \mathsf{R}\}$, either single deletion preserves the answer, and deleting both changes it. With $\mathsf{P} \wedge \mathsf{R} \to \mathsf{Q}$ instead, either deletion changes the answer. These two

cases can produce the same positive unedited response, but their certified counterfactual predictions differ. A model that fails the appropriate native intervention test has not implemented that candidate mechanism on the intervention domain, even when its unedited answer is correct.

H.2 Proof of Theorem 7

Proof. Fix the source boundary t. Repeated application of the native cell identities and the terminal decision condition shows that the indicator of a positive native decision from any $x \in X_t$ with a covered continuation equals $J_t(\alpha_t(x))$. Apply this statement to both x and $I(x)$. Substituting (14) in the latter gives $J_t(\upsilon(\alpha_t(x)))$. Subtracting the baseline gives (15). The expression no longer depends on the particular low-level implementation, proving implementation independence for this decision.

For a partial edit, let $\mathcal{E}^{\mathrm{end}}(x)$ be its nonempty set of permitted native endpoints with covered continuations. Its possible positive-decision indicators are exactly $\{J_t(\alpha_t(\widetilde{x})) : \widetilde{x} \in \mathcal{E}^{\mathrm{end}}(x)\}$. They are identical precisely when J_t is constant on the extracted endpoints. Subtracting the fixed baseline does not change this criterion. An empty set supplies no implementable counterfactual and is excluded from the constancy statement. □

The fixed decoder may assign different log odds to two native endpoints with the same logical code while selecting the same answer token at both. Theorem 7 therefore concerns decision effects only; implementation-independent numerical effects require an additional real-valued factorization. If certified intervals for the positive-versus-negative log odds are available at each terminal cell, they give interval-valued effects after subtracting the actual baseline. A singleton effect interval would require additional information, not merely the decision theorem.

H.3 Proof of the affine-control feasibility criterion

Let the proposed target fact/controller bit code be $b^{\mathrm{tgt}} \in \{0,1\}^{n_{\mathrm{code}}}$, with a valid controller encoding. Assume the endpoint must lie in a supplied polyhedron, the control vector must satisfy supplied polyhedral constraints, and named nonedited linear registers are fixed by supplied linear equations. At the fixed source x, the endpoint code scores are affine in h_{edit}:

$$M_{\mathrm{code}}x + M_{\mathrm{code}}L_{\mathrm{edit}}h_{\mathrm{edit}} + b^{\mathrm{code}}. \tag{93}$$

For a bit with target one, a margin of γ_{edit} requires the corresponding score to be at least γ_{edit}. For a bit with target zero, it requires the score to be at most $-\gamma_{\mathrm{edit}}$. Multiplying each row by $2b_j^{\mathrm{tgt}} - 1$ yields exactly (90). Endpoint-domain inequalities remain linear after substitution, as do the control constraints and the held-fixed equations. Thus a feasible solution implements the entire edit with the requested margin, and any such implementation satisfies the same linear system. This proves both directions.

For rational or real-algebraic input data, exact linear feasibility is decidable; an infeasible system admits a linear inconsistency certificate through Farkas' lemma. The assertion is relative to those explicit constraints, not a replacement for a nonlinear original-state domain by its outer polyhedral approximation. An outer approximation can produce spurious feasible endpoints. A certified inner polyhedron produces only genuine feasible endpoints but its infeasibility does not exclude controls outside it.

A finite-precision implementation of a real solution must be checked after rounding against the actual native endpoint computation. Alternatively, one can enumerate a supplied finite set of permitted control words and evaluate the exact machine semantics, although this can be prohibitively expensive. Neither route allows a reset or historical-cache edit that the stated intervention interface excludes.

H.4 Proof of Proposition 16

Proof. For any $\Delta_\Gamma \subseteq \Gamma$, monotonicity implies that if some $S \in \mathcal{U}_q(\Gamma)$ is disjoint from Δ_Γ, then $S \subseteq \Gamma \setminus \Delta_\Gamma$ and $q \in \mathrm{Cl}_{\mathcal{R}}(\Gamma \setminus \Delta_\Gamma)$. Conversely, if the latter entailment holds, finiteness permits removal of unnecessary premises until an inclusion-minimal subset $S \subseteq \Gamma \setminus \Delta_\Gamma$ still entails q. It is also minimal among subsets of Γ: any strictly smaller sufficient subset would contradict its chosen minimality. Thus $S \in \mathcal{U}_q(\Gamma)$ and $S \cap \Delta_\Gamma = \varnothing$.

We have proved

$$q \in \mathrm{Cl}_{\mathcal{R}}(\Gamma \setminus \Delta_\Gamma) \iff \exists S \in \mathcal{U}_q(\Gamma) \text{ with } S \cap \Delta_\Gamma = \varnothing. \tag{94}$$

Negating this equivalence gives (92). Since the original query is entailed, completeness and the original decision obligation turn nonentailment of the edited input into exactly a change from y_1 to y_0. A low-level edit realizing the same complete initialization has the same conclusion by Theorem 7. If the support family contains the empty set, its intersection with every deletion is empty, so no deletion can remove entailment, as required.

For hardness, start from a finite hitting-set instance, equivalently the incidence transpose of set cover (Karp, 1972), with universe U^{hit}, a family of nonempty sets $S_1^{\mathrm{hit}}, \ldots, S_m^{\mathrm{hit}}$ with $m \geq 1$, and a cardinality bound h_{hit}. Associate one input atom p_u with each $u \in U^{\mathrm{hit}}$, include all these atoms in Γ, and introduce a fresh query atom q not in Γ. For each set include the rule

$$\bigwedge_{u \in S_i^{\mathrm{hit}}} p_u \longrightarrow q. \tag{95}$$

There are no other rules. The query remains derivable after a deletion exactly when at least one of these bodies remains intact. Therefore a deletion of at most h_{hit} input atoms eliminates the query exactly when their universe elements hit every supplied set. The construction is polynomial in the explicit hitting-set instance. This proves NP-hardness of the cardinality-bounded deletion problem.

For membership in NP for an explicitly supplied finite ground Horn program, guess the removed input atoms and compute the closure of the remaining facts by repeated rule scans. There are at most n_{at} strict additions; each scan is polynomial in the total rule-body length. This verifies both the cardinality constraint and query absence in polynomial time. In particular membership holds for the one-step programs in the reduction, proving the stated restricted NP-completeness. Succinct nonground programs with an exponentially large grounding are a different input-size convention and are not covered by this complexity claim. □

The characterization concerns all sufficient supports, not only all traces of a particular scheduler. Under completeness, different legal schedules can choose different proofs while making the same deletion prediction. The support family can itself be exponentially large. Neither the proposition nor its proof claims that enumerating it is an efficient way to compute a minimum deletion.

H.5 Initial edits and edits after partial deduction

At initialization, setting the full retained fact state to $\Gamma \setminus \Delta_\Gamma$ and its controller to the corresponding initial control gives a clear logical starting point. A permitted native edit must realize that entire state and leave the original continuation within the certified domains. Editing only a visible prompt substring is not sufficient evidence that all retained facts and controller bits have those values; the resulting initialization must be checked.

After some deductions, a simple deletion $K \mapsto K \setminus \Delta_\Gamma$ takes already established conclusions not in Δ_Γ as remaining premises of the continued computation. For example, once $\mathsf{P} \to \mathsf{Q}$ has added Q, removing P but keeping Q does not remove the positive query bit. This is the exact prediction of the edited high-level state, not an inconsistency. A counterfactual intended to remove all dependence on the original fact must either reinitialize and rerun from edited input or explicitly revise all retained consequences whose status is being changed. Alternative supports must be considered in such a revision. The monotone Horn transition system contains no truth-maintenance step that retracts previously derived consequences.

I Logical execution and perturbation experiments

Implementation and comparisons. We evaluate the seven-stage coordinate equations for the reachability program in Appendix E.8 using double-precision PyTorch and the parameters specified in Appendix E. The protected normalization coordinate and null caches are treated as fixed constants. The implementation evaluates the coordinate formulas rather than materializing the full attention and feed-forward matrices. We compare SiLU correction, the corresponding ReLU correction, and replacement of the correction function by the identity, leaving the threshold expressions and

normalization unchanged. The identity replacement changes Boolean gate realization as well as numerical correction.

Reference states and perturbations. For $c = 2$, we enumerate all 2^8 fact assignments and all 12 valid controller states, giving 3,072 reference states with temporary coordinates set to -1. For each reference state and each seed in 1000–1009, we independently perturb every mutable coordinate uniformly in $[-r_{\mathrm{sh}}, r_{\mathrm{sh}}]$, where $r_{\mathrm{sh}} = 1/832$. This gives 30,720 perturbed states per variant, with identical perturbations across variants in this test. Each state undergoes one macrostep. We record the maximum coordinate error after each of its seven stages and sign agreement with the complete mutable reference code at the final boundary.

Long executions. For $c \in \{4, 8\}$, constants are labeled $0, \ldots, c-1$. The chain contains edges $(i, i+1)$ for $0 \le i < c-1$; the cycle adds $(c-1, 0)$; the disconnected graph removes $(c/2-1, c/2)$ from the chain; and the multiple-path graph adds $(i, i+2)$ for $0 \le i < c-2$. Initially, exactly these edge facts are true, all P facts are false, and the controller selects the first rule with zero digits. Each graph is run for 10^5 macrosteps under ten initial perturbations sampled from the same coordinatewise intervals, giving 40 runs per size and variant. Graphs are fixed across seeds. Long-run perturbations are shared across graph families within each size and seed, but use separate random streams for the three variants.

Evaluation and aggregation. A symbolic evaluator advances the reference facts and controller without feeding them back into the numerical execution. Fact and controller signs are checked after every macrostep, and the first disagreement defines a logical failure. Maximum coordinate errors, including temporary coordinates, are recorded at the first macrostep, every 500 macrosteps, and the terminal boundary. Thus the reported long-run maximum is a maximum over recorded boundaries, not over all intermediate stages. Checkpoints through the symbolic execution's last fact addition are classified as deduction; later checkpoints are classified as preservation. Perturbation summaries pool state–seed pairs, long-run summaries pool the 40 graph–seed runs at each size, and phase-specific summaries pool the corresponding checkpoints. Standard deviations describe variation across these records rather than standard errors across independent seeds.

J Storage and substitution-enumeration experiment

Storage accounting. We use $c \in \{2, 4, 8, 16, 32\}$ and count NumPy array storage through `nbytes`, excluding Python-object overhead. Shared metadata consists of the atom table, compatible premise and head atom indices, their variable–argument maps, and the rule metadata, all stored as 64-bit integers. These arrays occupy $8(31c^2 + 10)$ bytes. The explicit table contains one row per rule–substitution pair, storing the rule index, two premise indices, head index, and body length; an unused premise index is -1. Its storage is $40(c^2 + c^3)$ bytes. Shared metadata includes the $48c^2$-byte atom table, whereas the reported explicit-table size excludes it; the combined explicit size is recorded separately. Neither quantity measures the full neural parameter or working-state storage.

Inputs and timing. For each size and seed in 1000–1009, each possible directed edge, including self-loops, is independently present with probability 0.15, and all P facts are initially false. Fact arrays use double precision. Two CPU-based Python/NumPy evaluators perform the same sequential Horn updates: one generates substitutions through nested loops, and the other reads the explicit table. Both scan the first rule before the second, with the first variable changing fastest. For each input and method, three warm-up calls precede ten timed calls. Each call starts from a copy of the initial facts and includes one complete scan and checksum computation; metadata and table construction are excluded. Output agreement compares the complete fact arrays after the scan, not only a query answer or checksum.

Statistics and scope. For each size and method, timing means, sample standard deviations, medians, and extrema are computed over the 100 recorded calls. The standard deviation is therefore not the standard deviation of ten seed-level means. Log–log slopes are obtained by least-squares regression with an intercept over all five sizes, regressing the logarithm of storage bytes or arithmetic mean scan time on $\log c$. These timings concern the symbolic evaluators, not the seven-stage Trans-

former. Formula-based parameter counts and lightweight metadata-check timings are not used as measurements of complete model storage or certificate-verification complexity.

K INTERVENTION EXPERIMENT

Protocol. We use the seven-stage SiLU executor in double precision, with $c \in \{4, 8\}$ and seeds 1000–1009. Initial mutable coordinates receive independent uniform perturbations in $[-r_{\mathrm{sh}}, r_{\mathrm{sh}}]$, where $r_{\mathrm{sh}} = 1/832$. Each deletion sets only the selected fact coordinates to -1, $-1 + r_{\mathrm{sh}}/2$, or $-1 - r_{\mathrm{sh}}/2$. All other coordinates, including the current controller, remain unchanged, and the fixed caches are preserved. The three implementations of each edit share the same pre-intervention state.

Logical edits. Constants are labeled $0, \ldots, c-1$, and all unlisted facts are initially false. The unique-support case starts from $\{E(0,1), E(1,2)\}$, queries $P(0,2)$, and deletes $E(0,1)$ at initialization. The two-support case starts from $\{E(0,1), E(1,3), E(0,2), E(2,3)\}$ and queries $P(0,3)$; we delete either $E(0,1)$ alone or both $E(0,1)$ and $E(0,2)$. The late-edit case uses the unique-support input but deletes $E(0,1)$ after $c+1$ macrosteps, when $P(0,1)$ is present and $P(0,2)$ has not yet been derived.

Evaluation. Each edited state is continued for ten substitution cycles, comprising 800 macrosteps for $c = 4$ and 5,760 for $c = 8$. A separate symbolic evaluator follows the same rule order from the edited facts and unchanged control state; its outputs are used only for comparison. Edited and unedited decisions are evaluated at the same terminal boundary using the two answer logits and the common zero logit of all other vocabulary tokens. The maximum endpoint error was $1.201707 \times 10^{-3} < r_{\mathrm{sh}}$, the largest recorded macrostep-boundary error during continuation was 1.18×10^{-6}, and the minimum terminal margin was 2.00 to two decimal places. In the late-edit case, the symbolic reinitialization from the deleted input returned a negative answer, whereas the neural continuation retaining $P(0,1)$ returned a positive answer. This reinitialization comparison was computed symbolically, not as an additional neural run. The experiment tests the specified state edits and sampled endpoints, not attention-head attribution or universal validity over continuous domains.

L HIDDEN-STATE INTERVENTIONS IN A PRETRAINED MODEL

Task and data. We use a Boolean two-path reachability task with $c \in \{4, 8\}$ named constants. Four ordered input facts describe the two edges of each candidate path, and the query holds when either pair is true. Unique-support deletion, deletion of one redundant support, deletion of both supports, and restoration change these ordered fact bits as `1100` $\rightarrow$ `0100`, `1111` $\rightarrow$ `0111`, `1111` $\rightarrow$ `0101`, and `0100` $\rightarrow$ `1100`, respectively. For each seed in 1000–1009 and each size, we use 32 training contexts, four development contexts, and four test contexts, each covering all 16 fact assignments. This gives 256 training edit pairs and 32 development and test edit conditions per seed.

Protocol selection. The prompt was calibrated separately and frozen before this experiment. Protocol calibration determined layer 17, the final 96 token residuals, and a shared output rank of 256. The same context splits and cached states are used throughout the intervention evaluation. Test contexts are excluded from predictor fitting and selection, but were included during protocol calibration. We therefore report an exploratory evaluation with a predictor-level holdout, rather than a wholly independent confirmation of the complete discovery procedure.

State representation and prediction. For each seed, we estimate an uncentered rank-256 subspace from 512 token-level training difference vectors sampled across the matched counterfactual pairs. The same basis is used at every token position. Each token's predictor input contains its 256 projection coordinates, the first 64 coordinates of the window mean, 16 deterministic relative-position Fourier features, and a four-dimensional edit indicator. Training targets are the projected counterfactual state differences, not answer labels. Input and target normalization use training-set statistics.

We compare ridge regression with regularization strengths 10^{-4}, 10^{-3}, and 10^{-2}; nearest-neighbor regression with three or five neighbors; and a single-hidden-layer perceptron with 256 SiLU units. The perceptron is trained with AdamW for 60 epochs, using batch size 1024, learning rate 10^{-3}, and weight decay 10^{-4}. Development-set intervention results select one predictor per seed; every

Table 2: Test decision-effect agreement pooled over ten seeds. A condition succeeds only when all three endpoint variants match the symbolic prediction. Denominators count conditions whose original and counterfactual inputs are both answered correctly.

Edit	Successful / eligible	Success rate
Delete unique support	$80/80$	100%
Delete one redundant support	$50/78$	64.10%
Delete both supports	$80/80$	100%
Restore missing support	$73/80$	91.25%
Overall	$283/318$	88.99%

seed selected the perceptron. Test endpoints are predicted from the current state and edit indicator without using matched test counterfactual hidden states. Counterfactual model answers are used only to evaluate the predicted effects.

Continuous endpoint variants. The center endpoint adds the predicted subspace update to the current residual window. Two additional endpoints add or subtract a perturbation orthogonal to the output basis at every token. Its direction is the normalized orthogonal component of the source window, with a deterministically seeded random orthogonal direction used if that component is numerically zero. Its Frobenius norm is 0.02 times the largest of the predicted-update norm, 10^{-6} times the source-state norm, and 10^{-12}. The three endpoints have identical projected coordinates before conversion to FP16. We do not assume that this equality remains exact after conversion. Each endpoint is written into the original model, whose remaining operations and vocabulary projection are unchanged.

Evaluation and aggregation. The primary answer is the greedy next token over the full vocabulary; any token other than the declared Yes or No token is invalid. A test condition is eligible when the original and matched counterfactual inputs are both answered correctly. It succeeds only if all three endpoint variants produce the symbolic decision effect. Decision agreement across variants is recorded separately, since identical incorrect answers do not constitute success. The 320 test conditions require 960 intervention executions; these three executions per condition are not treated as independent samples. Of the 320 conditions, 318 are eligible. Table 2 reports pooled counts rather than averages of seed-level percentages. Unedited test accuracy is $1269/1280 = 99.14\%$, and decisions agree across the three variants in every eligible condition.

Development criteria and limitations. The prespecified acceptance criteria require development behavior accuracy of at least 90%, at least eight eligible nonzero-effect conditions and two eligible zero-effect conditions, three-variant success rates of at least 95% and 85% on those two groups, respectively, and decision agreement of at least 95%. Only seed 1001 satisfies every criterion; its test success count is $29/31$. All seeds remain in the reported aggregate, which is therefore exploratory. The lower success rate for redundant-support deletion shows that recovering answer changes does not ensure preservation when another derivation remains available. The experiment neither identifies a complete fact-and-control extractor nor verifies the cell inclusions required by Theorem 7. Agreement among these three endpoints is not a guarantee over all implementations of a complete logical edit.

Implementation and precision. Experiments use a single RTX 4090 with PyTorch $2.5.1$, Transformers $4.49.0$, and bitsandbytes $0.45.2$. Mistral weights remain frozen in NF4 four-bit representation, with FP16 forward computation, eager attention, and caching disabled during model execution. Perceptron training and evaluation use FP32; subspace calculations, intervention geometry, and statistical analysis use FP64. All ten no-op intervention checks reproduce the original full-vocabulary logits exactly.