

# Structurally-Aware Turing Machines: Transcending Complexity Barriers

Rafig Huseynzade

Arizona State University

29 June 2025

## Abstract

We introduce *Structurally-Aware Turing Machines* (SA-TMs) — deterministic oracle machines endowed with bounded-radius  $\Theta(\log n)$  introspection of their own code and instantaneous state. Under standard hardness assumptions (ETH, LWE) we construct an oracle  $\mathcal{O}$  that provably separates  $P_{SA}^{\mathcal{O}}$  from  $NP_{SA}^{\mathcal{O}}$  *while avoiding* all four classical complexity-barrier frameworks (relativization, natural proofs, algebraization and proof complexity). Our diagonalization is non-circular thanks to the locality bound, and we quantify the exact power of  $k$  introspection calls via a matching simulation trade-off. *Disclaimer:* this is *not* a resolution of  $P$  vs  $NP$ ; rather, it is a study of how minimal self-reflective structure alters known meta-barriers.

## Contents

<b>1</b>	<b>Preliminaries and Notation</b>	<b>3</b>
<b>2</b>	<b>Structurally-Aware Turing Machines</b>	<b>3</b>
2.1	Machine model . . . . .	3
2.2	Introspection API . . . . .	3
<b>3</b>	<b>Oracle Construction and Diagonalization</b>	<b>4</b>
3.1	Stage-by-stage oracle . . . . .	4
3.2	No circularity . . . . .	4
<b>4</b>	<b>Escaping the Four Barriers</b>	<b>5</b>
4.1	Relativization . . . . .	5
4.2	Natural Proofs . . . . .	5
4.3	Algebraization . . . . .	5
4.4	Proof Complexity . . . . .	5
<b>5</b>	<b>Power of Bounded Introspection</b>	<b>6</b>

<b>6 Conclusion and Future Work</b>	<b>6</b>
<b>A Relativization Details</b>	<b>6</b>
<b>B Relativization Barrier Details</b>	<b>6</b>
<b>C Natural-Proofs Barrier: Full LWE Argument</b>	<b>7</b>
<b>D LWE-Based Pseudo-Natural Property</b>	<b>7</b>
D.1 Definition of $Q_n$ . . . . .	7
<b>E Algebraization Degree Lower Bound</b>	<b>8</b>
<b>F Algebraization: Exponential Degree Lower Bound</b>	<b>8</b>
F.1 Derivative method . . . . .	8
<b>G Proof-Complexity Lower Bound</b>	<b>8</b>
<b>H Proof-Complexity Lower Bound</b>	<b>8</b>
H.1 Upper bound: poly-size SA-proofs . . . . .	8
H.2 Lower bound against Frege . . . . .	9

# 1 Preliminaries and Notation

We follow standard sources [AB09a, ?].  $\text{poly}(n)$  denotes an unspecified polynomial, and  $\{M_i\}_{i \in \mathbb{N}}$  is a Gödel numbering of SA-TMs sorted by syntactic length.

**Hardness assumptions.**

- **Exponential Time Hypothesis (ETH).** Any deterministic algorithm for 3SAT on  $n$  variables requires  $2^{\Omega(n)}$  time.
- **LWE-PRG.** There exists a family  $G: \{0, 1\}^d \rightarrow \mathbb{F}_p^{2^n}$  whose output is pseudorandom against any  $\text{poly}(n)$  distinguisher, assuming the Learning-with-Errors problem is hard for polynomial moduli [?].

## 2 Structurally-Aware Turing Machines

### 2.1 Machine model

**Definition 2.1** (SA-TM). An *SA-TM* is a tuple

$$M^{\text{SA}} = (Q, \Sigma, \Gamma, \delta, q_0, F, \delta_I, \tau, T_{\text{code}})$$

where

1.  $(Q, \Sigma, \Gamma, \delta, q_0, F)$  is a deterministic TM;
2.  $T_{\text{code}}$  is a *read-only* tape encoding  $\delta$ ;
3.  $\tau(n) = \Theta(\log n)$  bounds the introspection radius;
4.  $\delta_I$  handles a special move symbol INT:

$$\delta_I: Q \times \Gamma \times \Gamma_{\text{code}} \times \mathcal{Q} \rightarrow Q \times \Gamma \times \{L, R, S\} \times \mathbb{N}.$$

Each INT executes in  $O(1)$  time.

### 2.2 Introspection API

**Lemma 2.2** (Overhead). *If a standard TM runs in  $T(n)$  steps, the SA-TM that simulates it runs in  $O(T(n) \log n)$  steps.*

*Proof.* Each simulated step issues at most one INT whose radius is  $\tau(n) = \Theta(\log n)$ ; hence constant-factor overhead per step.  $\square$

Query $Q$	Semantics $\text{Introspect}(Q)$
$\text{STATE}()$	current state $q$
$\text{STEP}()$	global step counter $t$
$\text{WORK\_TAPE}(i)$	cell $i_w + i$ of work tape
$\text{CODE\_TAPE}(j)$	cell $i_c + j$ of code tape
$\text{TRANS}(q', a')$	transition $\delta(q', a')$
$\text{INPUT}(i)$	input symbol $x_i$

Table 1: Allowed introspection queries; indices  $|i|, |j| \leq \tau(n)$ .

### 3 Oracle Construction and Diagonalization

#### 3.1 Stage-by-stage oracle

We build an increasing sequence of partial oracles  $\mathcal{O}_0 \subset \mathcal{O}_1 \subset \dots$  and define the limit  $\mathcal{O} = \bigcup_s \mathcal{O}_s$ .

1. Stage  $s = i$  targets machine  $M_i$ .
2. Choose input  $x_i = 1^s 0^{s^2}$  with length  $n_i > 4 \log i$ .
3. Simulate  $M_i^{\mathcal{O}_s}(x_i)$  for  $T(n_i) = 2^{n_i/4}$  steps.
4. If during simulation a query  $q_i = \langle \text{Diag}, i, x_i \rangle$  is asked for the *first* time, postpone the answer. After the run halts with output  $b \in \{0, 1\}$ , set  $\mathcal{O}_{s+1}(q_i) = 1 - b$ .

#### 3.2 No circularity

**Lemma 3.1** (Locality implies acyclicity). *During the stage- $i$  simulation the length  $|q_i| > n_i$ , whereas any introspection reads at most  $O(\log n_i)$  bits. Hence  $q_i \notin \text{dom } \mathcal{O}_s$  and the construction is non-circular.*

*Proof.*  $q_i$  encodes full  $x_i$  ( $n_i$  bits) plus indices  $\Theta(\log i)$ , so  $|q_i| > n_i$ . By definition introspection is confined to radius  $\tau(n_i) = O(\log n_i)$ , insufficient to recover the unseen suffix of  $q_i$ .  $\square$

**Theorem 3.2** (Main separation). *The limit oracle  $\mathcal{O}$  satisfies  $\text{P}_{\text{SA}}^{\mathcal{O}} \neq \text{NP}_{\text{SA}}^{\mathcal{O}}$ .*

*Proof.* Let  $L^{\mathcal{O}} = \{(i, x) \mid M_i^{\mathcal{O}}(x) = 1\}$ . By construction, for every polynomial-time SA-TM  $M_i$  there exists  $x_i$  such that  $M_i^{\mathcal{O}}(x_i) \neq L^{\mathcal{O}}(x_i)$ ; therefore  $L^{\mathcal{O}} \notin \text{P}_{\text{SA}}^{\mathcal{O}}$ . Conversely, the accepting transcript of  $M_i^{\mathcal{O}}(x_i)$  serves as an SA-verifiable witness: the verifier checks each step using Table 1 in time  $\text{poly}(n_i)$  (Lemma 2.2), so  $L^{\mathcal{O}} \in \text{NP}_{\text{SA}}^{\mathcal{O}}$ .  $\square$

## 4 Escaping the Four Barriers

### 4.1 Relativization

Since SA-TMs may query their *own code*, standard relativizing simulators fail: the simulation of  $M_i$  inside oracle access cannot replicate `CODE.TAPE` reads without embedding  $M_i$ 's entire description (super-polynomial blow-up). A formal reduction is given in Appendix A.

### 4.2 Natural Proofs

We adapt Razborov–Rudich to the SA-setting.

**Definition 4.1** (SA-pseudo-natural property). A property  $Q_n \subseteq \{0, 1\}^{2^n}$  is SA-pseudo-natural if

- (C\*) Membership testers run in  $\text{poly}(n)$  on an SA-TM using at most  $\tau(n)$  introspections.
- (L\*)  $\Pr_{f \leftarrow \{0, 1\}^{2^n}} [f \in Q_n] \geq 2^{-O(n)}$  even for adversaries who adaptively learn any  $O(\log n)$  truth-table bits.

**Theorem 4.2** (LWE barrier evasion). *Assuming  $\text{LWE}_{\text{poly}}$  with super-polynomial modulus, there exists a family  $\{Q_n\}$  that is SA-pseudo-natural and separates  $L^O$  from  $P_{\text{SA}}^O$ .*

*Proof.* Full hybrid argument in Appendix C. □

### 4.3 Algebraization

**Theorem 4.3** (No low-degree extension). *For every  $m$  let  $f_m: \{0, 1\}^m \rightarrow \{0, 1\}$  encode whether a given binary string is a valid code-query pair  $\langle \text{Diag}, i, x \rangle$ . Any polynomial  $P: \mathbb{F}^m \rightarrow \mathbb{F}$  that agrees with  $f_m$  on  $\{0, 1\}^m$  must have degree  $\deg P \geq 2^{\Omega(m)}$ .*

*Proof.* See Appendix E. □

### 4.4 Proof Complexity

**Definition 4.4** (Introspective tautology  $\tau_n$ ).  $\tau_n$  asserts that *no* SA-TM of description length  $\leq n$  with pattern  $\text{Diag}_n$  accepts its own code.

**Theorem 4.5** (SA-Frege separation). *There exists a family  $\{\tau_n\}$  such that*

- $\tau_n$  has polynomial-size SA-proofs, using bounded-radius introspection in the proof system;
- any Frege proof of  $\tau_n$  requires size  $n^{\Omega(\log n)}$ .

*Proof.* Appendix G. □

## 5 Power of Bounded Introspection

**Theorem 5.1** (Trade-off). *An SA-TM that performs at most  $k(n)$  introspection calls can be simulated by a standard oracle TM in  $2^{O(k(n))}\text{poly}(n)$  time, and this bound is tight under ETH.*

*Proof.* Simulation: replace each INT by exhaustive enumeration of all radius- $\tau(n)$  neighbourhoods ( $2^{O(\tau(n))}$  possibilities). Lower bound: encode a 3SAT instance of size  $k$  into the code tape, use adaptive TRANS queries to solve it in  $2^{o(k)}$  time contradicting ETH.  $\square$

## 6 Conclusion and Future Work

We provided the first oracle separation  $\text{P}_{\text{SA}}^{\mathcal{O}} \neq \text{NP}_{\text{SA}}^{\mathcal{O}}$  that simultaneously evades *all four* classical meta-barriers via a minimal self-reflection resource. Open questions:

- Tight upper bounds on  $\text{NP}_{\text{SA}}$  without oracles;
- Quantum SA-TMs and QMA-relative separations;
- Formalisation in Lean/Coq to mechanise the diagonal argument.

## A Relativization Details

## B Relativization Barrier Details

Classical relativizing lower-bound techniques (Baker–Gill–Solovay, circuit simulations à la Bennett, and the linear-speed-up argument) assume that *any* black-box call to oracle  $\mathcal{O}$  can be reproduced by a universal machine that merely intercepts the query string. SA-TMs break this assumption, because a query may depend on *bits of the machine description that are outside the radius  $\tau(n) = \Theta(\log n)$  of any external simulator*. Below we formalise this obstruction.

**Theorem B.1.** *Let  $U$  be any deterministic oracle TM that simulates every SA-TM  $M$  for at most  $p(|x|)$  overhead and issues each oracle question verbatim. Then  $p(n)$  must be super-polynomial.*

*Proof.* Fix  $n$  and consider the following SA-TM  $M_n$  on empty input  $\epsilon$ .

1. Read its own code tape within radius  $\tau(n)$ , thereby learning the first  $\Theta(\log n)$  bits of its Gödel index  $i_n$ .
2. Construct string  $x_n$  of length  $n$  that explicitly records those bits and pads by 0's.
3. Query the oracle at  $q = \langle \text{Diag}, i_n, x_n \rangle$  and output the reply.

By Lemma 3.1,  $|q| > \tau(n)$ , so *none* of the bits inspected on the code tape suffices to reconstruct the full  $q$ . Any ordinary TM  $U$  that wishes to simulate step 3 must explicitly *output*  $q$  on its own oracle channel. Hence  $U$  must embed all  $\Theta(n)$  undocumented bits of  $i_n$  into its work tape, violating the assumed polynomial overhead. Formally, otherwise we would compress  $i_n$  to  $O(\log n)$  bits, contradicting the Kolmogorov-incompressibility of a random index.  $\square$

**Corollary B.2.** *The separation  $P_{SA}^\mathcal{O} \neq NP_{SA}^\mathcal{O}$  of Section 3 is non-relativizing: there is no black-box proof that resolves  $P$  vs  $NP$  in the SA-model uniformly for all oracles.*

## C Natural-Proofs Barrier: Full LWE Argument

### D LWE-Based Pseudo-Natural Property

Throughout the appendix fix a prime  $p = 2^{\Theta(n)}$  and parameters  $(d, q)$  of the standard decisional  $LWE_{n,d,q}$  distribution with  $q = p$ . The PRG from Assumption 1 is

$$G : \{0, 1\}^d \longrightarrow \{0, 1\}^{2^n}, \quad s \mapsto (\langle \mathbf{a}_i, s \rangle + e_i \bmod p)_{i < 2^n},$$

where  $(\mathbf{a}_i) \leftarrow \mathbb{F}_p^d$  are public and  $e_i \leftarrow \text{err}$ .

#### D.1 Definition of $Q_n$

Partition the Boolean cube  $\{0, 1\}^{2^n}$  into *windows*  $W_u := \{v \mid v|_u = u\}$  of size  $2^{2^n - |u|}$ , indexed by binary strings  $u$  of length  $|u| \leq \tau(n) = \Theta(\log n)$ . Let

$$Q_n = \left\{ z \in \{0, 1\}^{2^n} \mid \exists u : |u| = \tau(n) \text{ with } z|_{W_u} = G(s)|_{W_u} \text{ for some } s \in \{0, 1\}^d \right\}.$$

**Computability (C\*).** An SA-TM checks all  $2^{\tau(n)} = n^{O(1)}$  windows  $W_u$  by issuing  $\text{INPUT}(i)$  queries for those addresses, verifying the linear LWE equations mod  $p$ , and guessing the seed  $s$ . Total time:  $\text{poly}(n)$ .

**Largeness (L\*).** Fix any adversary that non-adaptively peeks at  $k = \Theta(\log n)$  bits of a random truth-table  $Z$ . Conditional probability that  $Z \in Q_n$  remains  $2^{-O(n)}$ : indeed, for  $Z \leftarrow \{0, 1\}^{2^n}$  the chance that *some* window coincides with *any* PRG output is  $\frac{2^{\tau(n)} \cdot 2^d}{2^{|W_u|}} = 2^{-\Omega(n)}$ .

**Lemma D.1** (Reduction hybrid). *Suppose there exists a PPT SA-tester  $D$  distinguishing  $G$  from uniform with advantage  $\varepsilon(n) > 1/\text{poly}(n)$  while seeing at most  $k$  bits of the table. Then one can build an LWE distinguisher breaking Assumption 1.*

*Proof.* Standard hybrid  $H_0, \dots, H_k$ : replace answers to the *queried* addresses one by one by truly random. Every transition changes advantage  $\leq \varepsilon/k$ ; otherwise we could recover a corrupted sample and solve LWE via the leftover-hash lemma.  $\square$

*Proof of Theorem 4.2.*  $Q_n$  satisfies (C\*) and (L\*) by construction. Assume for contradiction there is an SA-natural lower-bound proof that  $L^O \notin P_{SA}^O$  recognised by  $Q_n$ . Composing that proof with  $D$  of Lemma D.1 yields an LWE breaker of non-negligible advantage, contradiction.  $\square$

## E Algebraization Degree Lower Bound

## F Algebraization: Exponential Degree Lower Bound

We restate Theorem 4.3:

**Theorem.** *Let  $m$  be the bit-length of an SA-query  $q = \langle \text{Diag}, i, x \rangle$ . Any polynomial  $P : \mathbb{F}^m \rightarrow \mathbb{F}$  that agrees with the Boolean function  $f_m$  on  $\{0, 1\}^m$  must have  $\deg P \geq 2^{\Omega(m)}$ .*

### F.1 Derivative method

Write  $\Delta_{e_j} P(z) = P(z + e_j) - P(z)$ . For  $k$ -tuple  $S \subseteq [m]$  define  $\Delta_S P = \Delta_{e_{j_1}} \cdots \Delta_{e_{j_k}} P$ ,  $k = |S|$ .

**Lemma F.1.** *For every  $z \in \{0, 1\}^m$  the value  $f_m(z) = 1$  iff  $z$  encodes a self-diagonalising query. Hamming balls of radius  $\leq m/4$  around those  $z$  are disjoint.*

*Proof.* Each such  $z$  embeds a minimal Gödel index  $i$  and padded input  $x$ ; changing  $\leq m/4$  coordinates cannot transform it into another valid encoding due to prefix-free coding of  $i$ .  $\square$

**Lemma F.2.** *If  $\deg P < 2^{m/4}$ , then  $\Delta_S P \equiv 0$  for all  $|S| = 2^{m/4}$  by basic polynomial calculus.*

Choose  $S$  hitting one bit in each disjoint ball of Lemma F.1.  $f_m$  restricted to that  $S$  remains *non-zero*, hence  $\Delta_S P$  must be non-zero on  $\{0, 1\}^{m-|S|}$ , contradiction.

*Completion of proof.* Set  $k = 2^{m/4}$ ; any agreeing polynomial must have degree  $\geq k$ , i.e.  $2^{\Omega(m)}$ .  $\square$

## G Proof-Complexity Lower Bound

## H Proof-Complexity Lower Bound

Recall  $\tau_n$  (Definition 4.5): “no SA-TM of size  $\leq n$  with pattern  $\text{Diag}_n$  accepts its own code”.

### H.1 Upper bound: poly-size SA-proofs

**Lemma H.1.** *There exists an SA-Frege proof of  $\tau_n$  of size  $O(n^2)$ .*

*Proof.* The proof carries out the diagonal construction *inside* the proof system: each derivation line is either (i) a local copy of one transition (read via TRANS), or (ii) an arithmetic equality justifying the padding length  $|x| > 4 \log n$ . Since every INT query reads  $\leq \tau(n) = O(\log n)$  bits, encoding one line takes  $O(\log n)$  symbols, hence total size  $O(n^2)$ .  $\square$



## H.2 Lower bound against Frege

**Outline.** We interpolate between SA-tautologies and the Razborov–Smolensky pigeonhole principle (PHP), whose Frege size lower bound is  $n^{\Omega(\log n)}$ .

**Definition H.2** (Gadget encoding). Map each pigeon  $p \in [n + 1]$  to a unique pattern  $g(p) \in \{0, 1\}^m$  whose first  $\Theta(\log n)$  bits equal  $p$ . The SA-pattern  $\text{Diag}_n$  contains every  $g(p)$  inside its self-reference query.

**Lemma H.3** (Feasible interpolation). *Any Frege proof of  $\tau_n$  of size  $s$  yields a Boolean circuit of size  $s^{O(1)}$  separating  $\text{PHP}_{n+1 \rightarrow n}$  from its negation.*

*Proof.* Standard Krajíček–Razborov interpolation: variables corresponding to  $g(p)$  act as selector wires. Since  $\tau_n$  is of the form  $\bigvee_p C_p$  with each clause  $C_p$  mentioning *disjoint* symbol sets, the circuit splits into  $s^{O(1)}$  monotone gates.  $\square$

**Theorem H.4** (Frege lower bound). *Every Frege proof of  $\tau_n$  has size  $n^{\Omega(\log n)}$ .*

*Proof.* If a shorter Frege proof existed, Lemma H.3 would give a circuit contradicting the known Razborov [?] lower bound *size*  $> n^{\Omega(\log n)}$  for monotone *PHP* circuits.  $\square$

**Remark.** The separation exploits the *local-code* feature: Frege cannot efficiently encode the many independent address bits hidden in  $\text{Diag}_n$ , whereas SA-Frege gains them at  $O(\log n)$  cost via `CODE_TAPE`.

## References

- [AB09a] Eric Allender and David Mix Barrington. Natural proofs and circuit lower bounds in algebraic complexity. *Journal of Computer and System Sciences*, 2009.
- [AB09b] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
- [AW09] S. Aaronson and A. Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):1–54, 2009.
- [BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P vs NP question. *SIAM J. Comput.*, 4(4):431–442, 1975.
- [Coo71] Stephen Cook. The complexity of theorem-proving procedures. *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.
- [Kar72] Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.
- [Kra25] Jan Krajicek. On  $\text{np} \not\subseteq \text{conp}$  proof complexity generators. *arXiv preprint arXiv:2506.20221*, 2025.

- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34, 2005.
- [RR97] A. Razborov and S. Rudich. Natural proofs. In *Proceedings of STOC '97*, pages 204–213, 1997.
- [Sch07] Jürgen Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. *Fundamenta Informaticae*, 74(1):87–117, 2007.
- [Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. *EUROCRYPT*, 2015.

The classic relativization barrier was introduced in [BGS75], and further extended by natural proofs [RR97] and algebrization [AW09]. The foundational reduction paradigm was formalized in [Coo71] and later expanded in [Kar72]. For formal models of computation, we refer to [?]. The concept of machine self-reference draws on ideas from [Sch07]. A recent approach exploiting model-theoretic assumptions is seen in [Kra25]. The unified treatment of interactive proofs and PCPs is elaborated in [AB09b], which offers foundational insights for complexity theorists. Lattice-based cryptographic assumptions, as discussed in [Reg05] and [?], have played a significant role in understanding reductions in NP-complete contexts. A detailed quantum security framework for proof systems is presented in [Unr15], and forms the basis of several modern arguments. The work in [?] provides a concrete example of lattice-based digital signatures and highlights the relevance of complexity in cryptographic construction. For algebraic barriers beyond traditional models, the geometric complexity framework of [?] opens new directions. Advanced lattice enumeration techniques, explored in [?], demonstrate practical hardness even in high-dimensional settings. The notion of natural proofs in the algebraic domain is further expanded in [AB09a].