End-to-End Verification of Neuro-symbolic Automata via Contrastive Logit-Gaps

Abdelrahman Hekal[1,2] **Vasilis Manginas**[3,4]
Nikolaos Manginas[3,4] **Alessio Lomuscio**[1,5]
[1]Department of Computing, Imperial College London [2]Refractal Intelligence
[3]Institute of Informatics and Telecommunications, NCSR "Demokritos"
[4]Department of Computer Science, KU Leuven [5]Safe Intelligence

Abstract

Neuro-symbolic systems for sequence classification combine a neural perception network with a finite automaton over abstract events. Certifying their trajectory-level decisions under input perturbations is critical for safety-sensitive deployment. We address the open problem of end-to-end formal robustness certification for such Neurosymbolic Automata (NeSyA) under norm-bounded input perturbations. Naïve layerwise linear bound propagation fails on this composition: per-step relaxation slack compounds through the temporal recurrence (interval explosion), and the standard log-space transition score subtracts two log-sum-exp terms over overlapping coordinate sets that independent relaxations cannot tighten. To address these issues, we introduce contrastive logit-gap (CLG), a reparameterization whose two log-sum-exp heads act on disjoint coordinates and admits substantially tighter linear bounds, coupled with a fused temporal operator that yields corner-tight bounds from a single linearization. On NeSyA sequence benchmarks, our formulation scales significantly beyond recursive baselines, maintaining tight safety certificates at long horizons while reducing verification wall time by up to $\sim 6\times$. To our knowledge, this is the first end-to-end formal verification framework for temporal neuro-symbolic systems. Code is available at `https://github.com/Abdu-Hekal/nesyveri`.

1 Introduction

In neuro-symbolic systems for sequence classification, a neural network maps each time step in a sequence of images or sensor readings to abstract events and a discrete finite automaton (or an equivalent temporal specification over those events) defines which trajectories are admissible. Within this class, we adopt the Neurosymbolic Automata (NeSyA) formulation of Manginas et al. [1]: neural networks are composed with symbolic automata (temporal structure over events combined with propositional constraints) under probabilistic semantics in an end-to-end differentiable model for sequence classification and tagging.

For such systems, safety-critical settings, from autonomous driving to process monitoring, motivate not only accurate predictions but certified guarantees on behavior over entire trajectories under input uncertainty. The relevant safety claim is at the trajectory level: downstream actions depend on whether the whole sequence is admitted by the specification, not on any individual per-step prediction. Per-frame robustness does not imply trajectory robustness: small log-probability shifts that preserve each step's argmax can compound through the temporal recurrence and flip the trajectory-level decision. End-to-end verification, propagating bounds through both perception and automaton in a single graph, is needed to certify the trajectory output as a function of the original sensor input.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

While learning and inference for NeSyA systems have been developed [1], formal verification of their behavior under adversarial or norm-bounded input perturbations remains unaddressed in that line of work. Recent work on probabilistic neuro-symbolic robustness verification [2] proposes scalable relaxation-based methods for certifying robustness when a neural network extracts symbols from sub-symbolic inputs and a separate symbolic component reasons over those symbols to answer a query. That framework targets an important class of NeSy systems, but it does not treat the temporal case central to NeSyA, i.e., a sequence of perturbed observations composed with a differentiable automaton over an unrolled horizon. To our knowledge, no prior work studies end-to-end formal verification (bound propagation through both the perception network and the temporal automaton) for NeSyA-style systems. This is the topic of the present investigation.

Modern bound-propagation methods for deep networks can certify robustness of individual outputs in large networks by propagating interval or linear bounds through layers. When verification is lifted to sequences, however, a standard approach to verifying such compositions is to lift per-step bound propagation through the temporal recurrence, feeding bounds on the perception network's outputs into a differentiable encoding of the automaton and propagating intervals forward in time. This recursive composition is conceptually simple but often suffers from interval explosion: intervals widen at each step, so the composed bound on the trajectory property becomes unnecessarily loose, or vacuous, as the horizon T grows.

Our approach. We study end-to-end verification in the NeSyA setting: the perception network and the automaton are composed as a single differentiable module amenable to bound propagation. We address interval explosion in two complementary ways. Firstly, we introduce contrastive logit-gap (CLG), an algebraically equivalent reparameterization of the baseline transition score that routes interval bounds through a monotone map applied to a contrastive gap of restricted log-sum-exp terms, thereby avoiding the independent relaxation of $\mathrm{LSE}(\tilde{z}; \mathcal{S})$ and full-K $\mathrm{LSE}(\tilde{z})$ before subtraction (Section 3.2; Appendix A). Secondly, we contrast two structural encodings of the same temporal specification: path unrolling and a temporal-operator formulation that avoids redundant computation in the unrolled graph. We expect the temporal-operator formulation to be particularly advantageous as the symbolic system scales, where naïve unrolling pays a full multiplicative cost in the bound-propagation depth.

Contributions.

- We formulate end-to-end formal verification for NeSyA-style temporal neuro-symbolic systems under L_∞ input perturbations, treating the perception network and the automaton recurrence as a single differentiable map (Section 2).
- We introduce contrastive logit-gap (CLG), an algebraically equivalent reparameterization of the log-space transition score. The equivalence is classical (Lemma 1), but the two forms relax differently. We prove that CLG's linear bounds have disjoint coordinate support and characterize when this yields tighter CROWN bounds (Lemma 3, Theorem 4; Section 3.2).
- We propose a fused temporal operator. Using the classical convexity and monotonicity of the log-semiring path form (Lemma 7), we relax the entire depth-T program as a single operator with a corner-tight certificate (Proposition 8) instead of T chained LSE relaxations, eliminating compounding relaxation slack across time (Section 3.3).
- We empirically show that CLG combined with the fused temporal operator yields substantial gains in certified fraction over a strengthened lift of prior NeSy verification [2], with wall-time savings that scale with the symbolic system, reaching $\sim 6\times$ on our largest benchmark (Section 4 and Appendix D). Appendix B adds simplex-constraint refinements.

2 Preliminaries

We consider sequence classification under the Neurosymbolic Automata (NeSyA) semantics [1]: a neural perception module feeds a deterministic finite automaton whose transitions are guarded by propositional formulas over abstract symbols; Figure 1 depicts an example NeSyA system. Throughout, $T \in \mathbb{N}$ denotes the sequence length. Inputs are tensors $\mathbf{x} = (x_1, \ldots, x_T)$ with $x_t \in \mathbb{R}^{C \times H \times W}$. The threat model is an L_∞ ball around the clean sequence,

$$\mathcal{B}_\infty(\mathbf{x}, \varepsilon) \;=\; \big\{\mathbf{x}' : \|x'_t - x_t\|_\infty \le \varepsilon \text{ for all } t\big\}, \tag{1}$$

for fixed $\varepsilon > 0$. Uniformly over $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$, we seek sound lower and upper bounds on $\mathbb{P}_{\mathrm{acc}}(\mathbf{x}')$ or on $\log \mathbb{P}_{\mathrm{acc}}(\mathbf{x}')$; we define $\mathbb{P}_{\mathrm{acc}}$ and the robustness criterion below, then at the end of this section the LSE notation and bound-propagation primitives used in Section 3.

Perception model. Let f_θ be a convolutional network applied independently at each time step with shared weights, producing logits $z_t := f_\theta(x_t) \in \mathbb{R}^{K_{\mathrm{tot}}}$ (concatenated pre-softmax outputs for each categorical group). The logits partition into disjoint categorical groups indexed by $j = 1, \ldots, G$; each group induces a categorical distribution via softmax. Symbols of the automaton alphabet correspond to atomic predicates (Boolean variables); each predicate is tied to one class index within a group. Let n_j be the number of classes in group j, and $z_t^{(j)} \in \mathbb{R}^{n_j}$ the slice of z_t for group j, so that $z_t = (z_t^{(1)}, \ldots, z_t^{(G)})$. A joint assignment a is a tuple $(a_1, \ldots, a_G)$ with $a_j \in \{1, \ldots, n_j\}$. Following standard NeSyA semantics, we assume independence across groups at each time step, so

$$\mathbb{P}(a \mid x_t) \;=\; \prod_{j=1}^{G} \mathbb{P}_j(a_j \mid x_t), \tag{2}$$

with $\mathbb{P}_j$ the softmax over $z_t^{(j)}$. The number of joint assignments is $K := \prod_{j=1}^{G} n_j$, indexed by $k \in \{1, \ldots, K\}$. The corresponding joint logit is

$$\tilde{z}_{t,k} \;:=\; \sum_{j=1}^{G} z_t^{(j)}\big[a_j\big], \tag{3}$$

where $(a_1, \ldots, a_G)$ is the joint assignment indexed by k; stacking gives $\tilde{z}_t \in \mathbb{R}^K$, a linear function of z_t. Softmax on $\{1, \ldots, K\}$ with scores $\tilde{z}_t$ matches the product-of-softmax distribution above, and its log-normalizer is $\mathrm{LSE}(\tilde{z}_t)$ as in (5) below.

Automaton and acceptance probability. Let $\mathcal{A} = (Q, \delta, q_0, \mathcal{F})$ be a deterministic finite automaton over a finite input alphabet of abstract events, with $\mathcal{F} \subseteq Q$ the accepting states. Let $E \subseteq Q \times Q$ be the set of directed transitions (q, q') that carry a guard, and for $q \in Q$ let $\mathrm{Out}(q)$ denote the transitions in E with source q. Each transition carries a guard $\phi_{q,q'}$ expressed as a Boolean formula (NNF) over the atomic predicates. At time t, the probability of taking transition (q, q') is the sum of $\mathbb{P}(a \mid x_t)$ over all joint assignments a that satisfy $\phi_{q,q'}$; determinism means that for each q and each fully specified a, exactly one outgoing transition is enabled. Let $\pi_t(q)$ denote the (marginal) probability mass of being in state $q \in Q$ after processing $x_1, \ldots, x_t$. The acceptance probability is

$$\mathbb{P}_{\mathrm{acc}}(\mathbf{x}) \;=\; \sum_{q \in \mathcal{F}} \pi_T(q), \tag{4}$$

computed by the usual forward recurrence (probability or log-semiring). Equivalently, in log-space set $\alpha_t(q) = \log \pi_t(q)$, so $\mathbb{P}_{\mathrm{acc}} = \sum_{q \in \mathcal{F}} \exp\big(\alpha_T(q)\big)$ and $\log \mathbb{P}_{\mathrm{acc}} = \log \sum_{q \in \mathcal{F}} \exp\big(\alpha_T(q)\big)$; Section 3.2 gives the recurrence for α_t.

Verification objective. Each sequence carries a Boolean label $y \in \{0, 1\}$ indicating whether $\mathcal{A}$ should reject or accept under the data-generating process. Robustness is certified if, for all $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$, the acceptance probability stays on the correct side of $\frac{1}{2}$: for $y = 1$, $\mathbb{P}_{\mathrm{acc}}(\mathbf{x}') > \frac{1}{2}$ (equivalently $\log \mathbb{P}_{\mathrm{acc}} > -\log 2$), and for $y = 0$, $\mathbb{P}_{\mathrm{acc}}(\mathbf{x}') < \frac{1}{2}$. Let $[\ell, u]$ be sound bounds on $\log \mathbb{P}_{\mathrm{acc}}$ (or on $\mathbb{P}_{\mathrm{acc}}$ in probability-space variants) over the input box. Certification succeeds if $u < -\log 2$ when $y = 0$ and $\ell > -\log 2$ when $y = 1$ (log-space), or the analogous inequalities on $\mathbb{P}_{\mathrm{acc}}$ in $[0, 1]$.

Log-sum-exp and bound propagation. Throughout, K denotes the joint-assignment count from the perception model. For $K \geq 1$ and $v \in \mathbb{R}^K$, the log-sum-exp function is

$$\mathrm{LSE}(v) \;:=\; \log\Big(\sum_{i=1}^{K} \exp(v_i)\Big). \tag{5}$$

For nonempty $\mathcal{S} \subseteq \{1, \ldots, K\}$ we also write $\mathrm{LSE}(v; \mathcal{S}) := \log \sum_{i \in \mathcal{S}} \exp(v_i)$ for the restriction to coordinates in $\mathcal{S}$ (equivalently, coordinates outside $\mathcal{S}$ may be sent to $-\infty$ in the full-K LSE). For

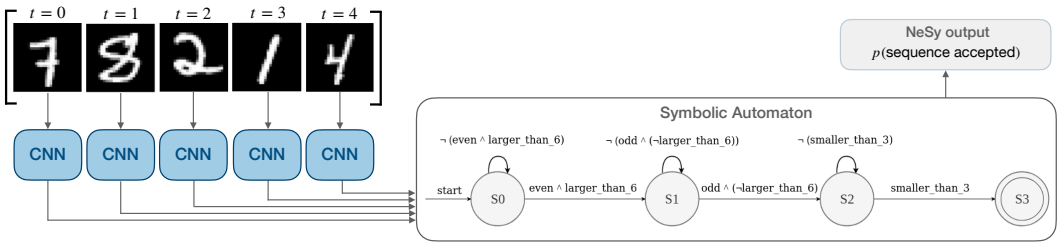

Figure 1. Example of a NeSyA system: a shared perception network maps each frame x_t to abstract events (e.g. magnitude and parity predicates on a digit); the automaton consumes the resulting symbol sequence and assigns acceptance probability $\mathbb{P}_{\mathrm{acc}}(\mathbf{x})$. Certification bounds $F(\mathbf{x}) = \log \mathbb{P}_{\mathrm{acc}}(\mathbf{x})$ over the full sequence under L_∞ perturbations of every frame.

joint logits $\tilde{z} \in \mathbb{R}^K$, $\mathrm{LSE}(\tilde{z})$ is the log-normalizer of the softmax distribution on $\{1, \ldots, K\}$. The softplus function is $\mathrm{softplus}(t) := \log(1 + e^t)$ for $t \in \mathbb{R}$. Because f_θ and the automaton forward map compose into a fixed feedforward function of $\mathbf{x}$, bounds on $\mathbb{P}_{\mathrm{acc}}$ or $\log \mathbb{P}_{\mathrm{acc}}$ can in principle be obtained by propagating axis-aligned input boxes through that composition using interval arithmetic and linear relaxations of the coordinatewise nonlinearities (e.g., ReLU and log-sum-exp); the methods in Section 3 are agnostic to which concrete relaxation is used and address how to structure that composition.

3 Method

We certify $\log \mathbb{P}_{\mathrm{acc}}$ uniformly over $\mathcal{B}_\infty(\mathbf{x}, \varepsilon)$ by viewing the NeSyA as a single feedforward map and applying neural network certification to it. Standard linear relaxations of $\log \mathbb{P}_{\mathrm{acc}}$ accumulate slack at two distinct points along the bound graph. (A) Transition scores: the per-step log-probabilities $s_{t,m}$ that the automaton consumes (formal definition in (6)); softmax mass is expressed through LSE, and a textbook difference of overlapping LSE heads couples coordinates in a way that standard independent linear relaxations fail to exploit. (B) Temporal aggregation: encoding the log-semiring forward pass as T sequential Bellman LSE-type updates deepens the relaxation graph and lets slack compound along the horizon. We treat (A) via contrastive logit-gap reparameterization (Section 3.2) and (B) via reorganizing how scores are aggregated in time (Section 3.3).

3.1 End-to-end certification setup

Define $F(\mathbf{x}) := \log \mathbb{P}_{\mathrm{acc}}(\mathbf{x})$, composing (i) time-distributed perception $z_t = f_\theta(x_t)$; (ii) per (t, m) scores $s_{t,m}$ computed from the joint logits $\tilde{z}_t \in \mathbb{R}^K$ formed from z_t (Section 3.2); and (iii) temporal aggregation to $\log \mathbb{P}_{\mathrm{acc}}$ (Section 3.3). The same acceptance log-probability, viewed as a function of stacked transition scores, is written $F(s_{1:T})$ in Section 3.3.

Incomplete certifiers based on layerwise linear relaxations [3, 4] or interval bound propagation [5, 6] enclose each nonlinearity on an axis-aligned input box; these compose to bounds $\underline{w}^\top \mathbf{x} + \underline{b} \le F(\mathbf{x}) \le \overline{w}^\top \mathbf{x} + \overline{b}$ on $\mathcal{B}_\infty(\mathbf{x}, \varepsilon)$. End-to-end here means relaxing the full map $\mathbf{x} \mapsto \log \mathbb{P}_{\mathrm{acc}}$ as a single feedforward computation, with the CNN and automaton sharing one bound graph. Under a recursive temporal encoding (Section 3.3), effective depth grows with T because each Bellman LSE stage adds slack that composes with the next (Appendix A.2).

3.2 Transition scores: baseline and contrastive logit-gap (CLG)

The baseline instantiates $s_{t,m}$ by the textbook softmax decomposition. For transition m with feasible joint indices $\mathcal{S}_m \subseteq \{1, \ldots, K\}$ and joint logits $\tilde{z}_t \in \mathbb{R}^K$ at time t,

$$s_{t,m} \;=\; \mathrm{LSE}(\tilde{z}_t; \mathcal{S}_m) - \mathrm{LSE}(\tilde{z}_t) \;=\; \log \mathbb{P}(m \mid \tilde{z}_t). \tag{6}$$

Let $\alpha_t \in \mathbb{R}^{|Q|}$ denote log-masses with $\alpha_0(q_0) = 0$ (i.e., probability mass one on q_0) and $\alpha_0(q) = -\infty$ for $q \neq q_0$. The log-semiring forward pass is

$$\alpha_t(q') \;=\; \log \sum_{q \in Q:\,(q,q') \in E} \exp\big(\alpha_{t-1}(q) + s_{t,(q,q')}\big), \tag{7}$$

$$\log \mathbb{P}_{\mathrm{acc}} \;=\; \log \sum_{q \in \mathcal{F}} \exp\big(\alpha_T(q)\big). \tag{8}$$

Why the baseline score is hard to relax. Equation (6) subtracts two LSE maps that share support on $\mathcal{S}_m$. The true Jacobian at coordinate $i \in \mathcal{S}_m$ is the difference of two restricted softmax projections, one over $\mathcal{S}_m$ and one over the full alphabet $\{1, \ldots, K\}$, which nearly cancel on shared support since both share most of the same exponentials. However, layerwise linear relaxations bound each LSE tangent slope independently, then difference the bounds, leaving residual coefficients on shared $\tilde{z}_{t,i}$ that do not see the cancellation (Lemma 3, Appendix A.1). Separately, (7) chains T LSE-type aggregations whose layerwise slack accumulates (Section 3.3; Remark 9).

CLG reparameterization. We retain (7)–(8) and replace only each $s_{t,m}$ as a function of $\tilde{z}_t$. Fix feasible $\mathcal{S} \subsetneq \{1, \ldots, K\}$ (we drop the m subscript while working with a single transition) and write $\tilde{z} \in \mathbb{R}^K$ at a single time step. The contrastive gap is

$$g(\tilde{z}; \mathcal{S}) \;:=\; \mathrm{LSE}(\tilde{z}; \mathcal{S}^{\mathrm{c}}) - \mathrm{LSE}(\tilde{z}; \mathcal{S}), \qquad \mathcal{S}^{\mathrm{c}} := \{1, \ldots, K\} \setminus \mathcal{S}. \tag{9}$$

With $p_{\mathcal{S}} = (1 + e^{g})^{-1}$,

$$s_{\mathrm{clg}}(\tilde{z}) \;:=\; \log \mathbb{P}(\mathcal{S}) \;=\; -\,\mathrm{softplus}\big(g(\tilde{z}; \mathcal{S})\big) \;=\; -\log\big(1 + \exp g(\tilde{z}; \mathcal{S})\big), \tag{10}$$

identical to (6) for all $\tilde{z}$ (Lemma 1, Appendix A.1); (10) is the standard logistic–softplus form [7].

The intuition for this encoding is twofold. (i) Disjoint support. The two LSE heads in g act on disjoint coordinates, so relaxations are not differenced on shared $\tilde{z}_i$ as in (6). (ii) Outer $-\,\mathrm{softplus}$. Its derivative is $-\varsigma(g) \in (-1, 0)$, where $\varsigma(t) := 1/(1 + e^{-t})$ is the logistic sigmoid; concavity then gives tangent/chord bounds whose slopes lie in $(-1, 0)$, contracting inner coefficient magnitudes relative to identity propagation on g whenever the outer multiplier is bounded away from -1. The outer slope's range makes the effect asymmetric in $p_{\mathcal{S}}$: when $p_{\mathcal{S}} \to 1$, $\varsigma(g) \to 0$ damps propagated slack strongly, while when $p_{\mathcal{S}} \to 0$ (equivalently $g \to +\infty$), $-\,\mathrm{softplus}(g)$ is asymptotically linear with slope -1, so the additive secant slack is small but the multipliers approach -1, transmitting rather than contracting inner LSE relaxation error from $\mathrm{LSE}(\tilde{z}; \mathcal{S})$ and $\mathrm{LSE}(\tilde{z}; \mathcal{S}^{\mathrm{c}})$. CLG therefore is not guaranteed to tighten the baseline in the low-mass regime (Remark 5); Theorem 4 states the asymmetric structure formally.

3.3 Temporal encodings

Given $s_{t,m}$ from (6) or CLG (10), three equivalent aggregations differ in how LSE-shaped nonlinearities surface to the bound propagator.

Recursive dynamic programming. The forward pass is (7)–(8): depth scales linearly in T in LSE-shaped stages, so slack may accumulate (Remark 9).

Path aggregation. For moderate $|\mathcal{P}_{\mathrm{acc}}|$ and T, enumerate the set $\mathcal{P}_{\mathrm{acc}}$ of accepting state paths of length T. Each path picks transitions $m_{p,t}$ and

$$\log \mathbb{P}(p \mid \mathbf{x}) \;=\; \sum_{t=1}^{T} s_{t,m_{p,t}}. \tag{11}$$

Let $\eta \in \mathbb{R}^{|\mathcal{P}_{\mathrm{acc}}|}$ with components $\eta_p := \log \mathbb{P}(p \mid \mathbf{x})$. Then

$$\log \mathbb{P}_{\mathrm{acc}} \;=\; \log \sum_{p \in \mathcal{P}_{\mathrm{acc}}} \exp(\eta_p) \;=\; \mathrm{LSE}(\eta) \;=\; \log \sum_{p \in \mathcal{P}_{\mathrm{acc}}} \mathbb{P}(p \mid \mathbf{x}), \tag{12}$$

i.e. one LSE over paths, trading a depth-T chain of per-step LSE updates for an outer sum of size $|\mathcal{P}_{\mathrm{acc}}|$, which is generally exponential in T (a $|Q|$-state automaton with branching factor b admits up to b^T accepting state paths); path aggregation is therefore practical only when $|\mathcal{P}_{\mathrm{acc}}|$ stays manageable. A single LSE relaxation introduces one secant slack term, whereas the recursive form of (7) introduces T secant slacks whose contributions accumulate through the chained Jacobians, so path aggregation is typically tighter in addition to flattening the relaxation depth.

Table 1: NeSyA-MNIST: certified fraction (%) by variant and ε, with amortized wall time (*s/seq*). Per-label split in Appendix F. Best per-column percentage values and lowest *s/seq* bold.

Variant	$\varepsilon = 0.001$		$\varepsilon = 0.002$		$\varepsilon = 0.005$		$\varepsilon = 0.01$	
	cert. %	*s/seq*	cert. %	*s/seq*	cert. %	*s/seq*	cert. %	*s/seq*
Baseline (recursive)	99.2	*0.16*	57.2	*0.20*	42.5	*0.23*	26.7	*0.25*
CLG (recursive)	**99.5**	*0.15*	98.7	*0.19*	92.5	*0.22*	59.9	*0.24*
CLG path-unroll	**99.5**	*0.14*	**99.0**	*0.17*	93.8	***0.20***	64.7	*0.22*
CLG temporal-op	99.4	***0.13***	**99.0**	***0.16***	**93.9**	***0.20***	**65.0**	***0.21***

Fused temporal operator. Let $F(s_{1:T}) := \log \mathbb{P}_{\mathrm{acc}}$ from the same log-semiring program. Path form $F(s) = \mathrm{LSE}_{p \in \mathcal{P}_{\mathrm{acc}}}(\sum_t s_{t,m_{p,t}})$ makes F convex and coordinatewise nondecreasing (Lemma 7; standard log-partition structure [8]). The gradient $\nabla F(s)$ collects edge marginals from one forward–backward pass, classical inside–outside / semiring backprop, at the same $\Theta(T|E|)$ cost as the recursive Bellman pass (Appendix A.2). Convexity gives a lower bound from any subgradient at s^ℓ, and coordinatewise monotonicity places the box maximum at the corner s^u, so on score box $[s^\ell, s^u]$,

$$F(s^\ell) + \nabla F(s^\ell)^\top (s - s^\ell) \;\le\; F(s) \;\le\; F(s^u), \qquad s \in [s^\ell, s^u], \tag{13}$$

with $\nabla F(s^\ell)$ from that backward pass (Proposition 8). The temporal map thus receives one joint linearization instead of T nested ones along (7)–(8); accepting paths are unchanged.

4 Evaluation

We evaluate end-to-end robustness certification for NeSyA models under the threat model of Section 2. Unless stated otherwise, we report the fraction of test sequences for which sound bounds certify the correct accept/reject decision uniformly on $\mathcal{B}_\infty(\mathbf{x}, \varepsilon)$, together with amortized wall-clock time per sequence (s/seq). We compare the four aligned variants from Section 3; tables denote them Baseline (recursive), CLG (recursive), CLG path-unroll, and CLG temporal-op.

All runs use logits and the log-semiring automaton forward pass unless noted. The baseline row Baseline (recursive) temporally lifts the relaxation-based NeSy verifier of Manginas et al. [2] to NeSyA: scores (6) along (7)–(8). We additionally replace the IBP instantiation in their work with CROWN-style linear bounds [3] (Section 3.1); this makes the baseline strictly stronger than transplanting their pipeline to sequences without that upgrade, so gains attributed to CLG and the temporal encodings are conservative. The three CLG rows replace (6) by (10) and differ only in aggregation: recursive (same recurrence as the baseline), path unroll, or fused temporal operator. To fully optimize the baseline, we implement it in its factored per-group form (Appendix C), which is strictly tighter under CROWN than the literal joint two-LSE form (6); the literal joint form appears only in the per-step width study below, head-to-head with CLG.

We instantiate the layerwise linear-bound framework of Section 3.1 using CROWN-style relaxations from the auto_LiRPA library [4] on top of PyTorch [9]; wall-clock times are reported on a single Linux GPU workstation (Appendix G).

4.1 NeSyA-MNIST

We construct a NeSyA-MNIST pipeline (Figure 1, full specification in Appendix C.1) over the MNIST digit dataset [10]: each frame is one digit image scored by a CNN whose logits split into two independent categorical groups (digit magnitude and parity), yielding joint-assignment cardinality $K = 6$ per timestep; sequence length T is drawn uniformly from $\{3, \ldots, 10\}$ for the main robustness sweep. We sample a class-balanced test set of 2000 sequences, and run verification only on those whose clean-input prediction matches the ground-truth label (991 of 1000 for y=0 and 965 of 1000 for y=1, totaling 1956); reported certified fractions are over these post-filter counts.

Certified fraction versus ε. CLG matches or outperforms the baseline at every radius in the sweep (Table 1): at ε=0.01 the baseline drops to 26.7% while the strongest CLG variant retains 65.0% (a 38pp gap), and the spread widens with ε. Among CLG encodings, path unrolling and the fused

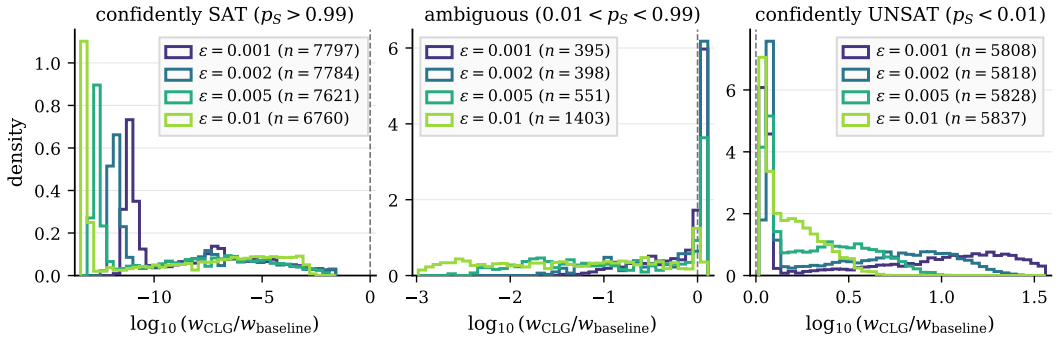

Figure 2. Per-pair CLG-vs-baseline bound widths, stratified by clean-image regime: density histograms of $\log_{10}(w_{\mathrm{CLG}}/w_{\mathrm{baseline}})$, one curve per ε. Dashed line at 0 marks equal widths. **Left:** confidently-SAT pairs ($p_{\mathcal{S}_m} > 0.99$, ∼47–56% of pairs); mass at $\log_{10} \leq -10$, where CLG is essentially exact (regime A). **Middle:** ambiguous pairs ($0.01 < p_{\mathcal{S}_m} < 0.99$, ∼3–10%); mass near zero with slight CLG lean. **Right:** confidently-UNSAT pairs ($p_{\mathcal{S}_m} < 0.01$, ∼41%); mass just right of zero, where CLG is mildly looser (regime B). P(CLG strictly tighter) ≈57% marginally, regime-weighted from 100%/∼72%/0%.

temporal operator achieve the strongest certified fractions; recursive CLG remains competitive but does not fully close the gap to the two aggregations that group the LSE-shaped stages differently for relaxation (Proposition 8 and Remark 9).

Isolating the CLG mechanism. To isolate the per-step CLG contribution from temporal aggregation, we measure CROWN bounds at the transition-score interface directly. For each test image x and each automaton transition m, we compute the interval bound $[s_m^\ell, s_m^u]$ on $\log \mathbb{P}(m \mid x)$ under two heads that share identical CNN weights, joint logits $\tilde{z}$ formed from network logits $z = f_\theta(x)$ as in Section 2, and input box $\mathcal{B}_\infty(x, \varepsilon)$: the baseline two-LSE subtraction $s_m = \mathrm{LSE}(\tilde{z}; \mathcal{S}_m) - \mathrm{LSE}(\tilde{z})$ from (6), and the CLG softplus form $s_m = -\mathrm{softplus}\big(\mathrm{LSE}(\tilde{z}; \mathcal{S}_m^{\mathrm{c}}) - \mathrm{LSE}(\tilde{z}; \mathcal{S}_m)\big)$ from (10). Per-pair width is $w_m = s_m^u - s_m^\ell$ and the per-pair statistic is $\log_{10}(w_{\mathrm{CLG}}/w_{\mathrm{baseline}})$ (negative ⇒ CLG tighter). The slice is the 2000-sequence MNIST test set used throughout Section 4.1, taking the first frame of each sequence (one image per sequence is sufficient because the perception network is time-distributed); for each image the 7 automaton transitions give 14,000 pairs per ε, over 4 radii.

Figure 2 shows three distinct regimes (Theorem 4). Regime A ($p_{\mathcal{S}_m} \to 1$): the outer-softplus multiplier $\varsigma(g^*)$ collapses to zero and CLG is essentially exact on the box; 32–47% of pairs (depending on ε) hit the numerical floor $w_{\mathrm{CLG}} < 10^{-6}$, while baseline median widths in this regime range from 0.11 at ε=0.001 to 16.6 at ε=0.01. Regime B ($p_{\mathcal{S}_m} \to 0$): the multiplier saturates at one, the inner LSE relaxation slack is preserved, and CLG is mildly looser than baseline, with median 1.3–3.3× wider widths on the looser subset. Ambiguous ($0.01 < p_{\mathcal{S}_m} < 0.99$, matching the figure's stratification): neither asymptote applies, but CLG still leans tighter, with 72% of these pairs CLG-tighter pooled (rising from ∼45% at ε=0.001 to ∼90% at ε=0.01; Remark 5). Pooled, CLG is strictly tighter than baseline on 57% of (image, m) pairs at every ε in the sweep, and the median pair has $w_{\mathrm{CLG}} \approx 0$ versus a baseline scale $\propto \varepsilon$ (e.g. 0.06 vs 15.7 at ε=0.01); per-pair statistics and per-path arithmetic are in Appendix A.1.

This per-step structure is label-symmetric: the per-frame regime distribution is identical between y=0 and y=1 sequences (Appendix F, Figure 8 left). The recursive LSE-DP, however, is dominated by its max term in the trained-model regime and operationally routes the bound on $\log \mathbb{P}_{\mathrm{acc}}$ through the SAT-accepting path (Figure 9: end-to-end CROWN bounds match the Viterbi-summed prediction with $R^2 \geq 0.99$ on the lower bounds, which drive y=1 certification). The regime mix on that path differs sharply by label, with 96.7% regime A on y=1 versus 72.8% A and 24.8% B on y=0 (Figure 8 right); combined with the small headroom on y=1 (∼0.7 units against $-\log 2$), this increases the baseline-vs-CLG gap on y=1 sequences. Full per-label decomposition is in Appendix F.

Scaling of certification with the horizon T. Figure 3 measures the same four methods at fixed $\varepsilon = 0.005$ while increasing the sequence length on the grid $T \in \{4, 6, \ldots, 20\}$, isolating growth in

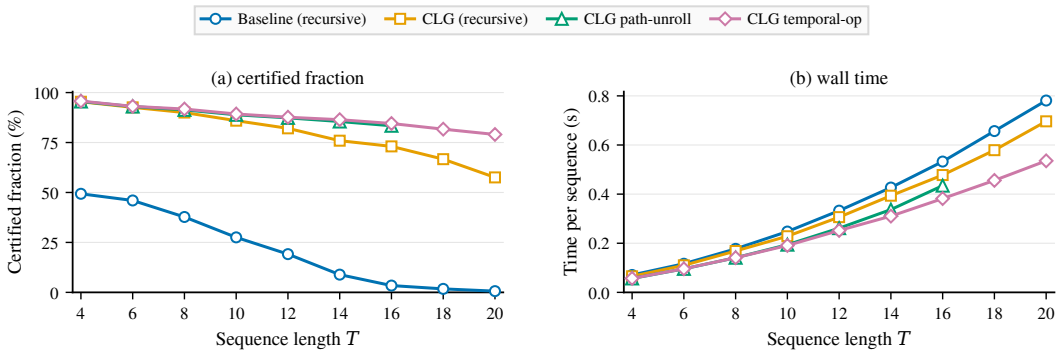

Figure 3. NeSyA-MNIST: certified fraction (%, *left*) and amortized wall time per sequence (s, *right*) versus sequence length $T \in \{4, 6, \ldots, 20\}$ at ε=0.005.

Table 2: CAVIAR (stride 25): clips certified out of 13 by variant and ε, with amortized wall time (*s/seq*). Best per-column certified count and lowest *s/seq* bold.

Variant	$\varepsilon = 2{\times}10^{-5}$		$\varepsilon = 5{\times}10^{-5}$		$\varepsilon = 10^{-4}$		$\varepsilon = 2{\times}10^{-4}$	
	cert.	*s/seq*	cert.	*s/seq*	cert.	*s/seq*	cert.	*s/seq*
Baseline (recursive)	**13**	*177*	**13**	*200*	**12**	*232*	1	*283*
CLG (recursive)	**13**	*158*	**13**	*181*	11	*210*	**9**	*261*
CLG temporal-op	**13**	***72***	**13**	***91***	11	***116***	**9**	***165***

relaxation depth along the log-semiring chain (Remark 9) from variation in ε. Certified fractions decline as T increases across methods, but the four variants separate cleanly: the baseline collapses from 49.3% at T=4 to 0.6% at T=20 (per-step slack accumulating linearly through the recurrence per Remark 9), recursive CLG drops (95.4% $\rightarrow$ 57.5%), and the two corner-tight aggregations stay highest, with the fused temporal operator retaining 79.0% at T=20. Path unrolling tracks the fused operator closely up to T=16 (83.4% vs 84.5%) but runs out of memory at $T \geq 18$; the same combinatorial cost rules it out for the larger benchmark below. Appendix D reports MNIST-extended, which scales the symbolic system simultaneously along K (joint alphabet), automaton size, and horizon: K=216, ten-state automaton, T=30. The qualitative ordering carries over at radii $\sim$10$\times$ tighter (consistent with the 36$\times$ alphabet expansion), with CLG variants retaining 89–99% certified at $\varepsilon \geq 5{\times}10^{-4}$ while the baseline drops to 52.1%, and to 0% on y=1 sequences (a sharp cliff from 93.5% at ε=$2{\times}10^{-4}$). Appendix E varies automaton size, alphabet, and guard fan-in one at a time, and shows that the remaining looseness is almost entirely per-step while runtime is dominated by per-frame CNN bound propagation.

4.2 CAVIAR: real-world video event recognition

CAVIAR is the real-world video event-recognition benchmark used in the original NeSyA pipeline [1]: short clips of two-person tracking are scored by a per-frame CNN whose categorical predictions feed a three-state deterministic automaton over complex events (`no_event`, `moving`, `interacting`); joint-assignment cardinality is K=32. For each clip we certify $\mathbb{P}_{\text{model}}\big(\text{state}[T_{\text{ds}}] = c_i^\star\big) > \frac{1}{2}$ uniformly on $\mathcal{B}_\infty(\mathbf{x}, \varepsilon)$, where $c_i^\star$ is the state at the final downsampled frame; we filter to the 13 of 17 clips that satisfy this on clean inputs. Raw clips are sampled at 1 fps (stride 25), giving downsampled lengths $T_{\text{ds}} \in [4, 43]$ across the 13 test clips. By construction every verification label is a positive instance (y=1): the spec asks the verifier to maintain the model's final-state probability above $\frac{1}{2}$ at the target, and Table 2 reports a single certified count per cell. Full perception network, automaton, and spec are in Appendix C.3.

Findings. Table 2 reproduces the same per-step pattern observed on NeSyA-MNIST (Section 4.1: CLG essentially exact on confidently-fired transitions, mildly looser on confidently-violated ones) on a real-video benchmark, with one new feature: baseline is marginally better than CLG at $\varepsilon = 10^{-4}$ (12/13 vs 11/13), but collapses to 1/13 at $\varepsilon = 2{\times}10^{-4}$ while CLG retains 9/13. The mechanism is

the one formalized in Theorem 4: on most clips the trained CNN's clean per-frame predictions place every fired transition m in regime A ($p_{\mathcal{S}_m} \to 1$), where the outer multiplier $\varsigma(g^*) \to 0$ damps inner LSE slack to near zero and CLG yields effectively exact transition-score bounds, while baseline's factored per-group log-softmax retains a quadratic-in-ε secant slack that accumulates linearly through the recurrence. Per-clip inspection confirms this: on a representative all-regime-A clip (T_{ds}=10), CLG produces zero-width bounds at both $\varepsilon = 10^{-4}$ and $\varepsilon = 2\times10^{-4}$ while baseline's lower bound shifts by -0.63, crossing $-\log 2$; the same shift pattern recurs across the other all-regime-A clips and accounts for the 1-vs-9 certified gap at $\varepsilon = 2\times10^{-4}$. The smaller-$\varepsilon$ baseline edge comes from a single ambiguous-regime clip on which CLG pays its joint $K = 32$ LSE cost while baseline operates on per-group inputs of size $n_j \in \{2, 4\}$, compounding over T_{ds}=43 steps into a $\sim$0.5-unit lower-bound gap (Remark 5's ambiguous-range behavior). The fused temporal operator preserves CLG's certified counts at every ε while reducing wall time by roughly 1.5–$2\times$ relative to recursive CLG (165 vs 261 s/seq at $\varepsilon = 2\times10^{-4}$; comparable savings at smaller radii), consistent with Proposition 8: one joint linearization in place of T_{ds}=43 nested per-step relaxations.

5 Related Work

There exists extensive work on verifying purely symbolic systems: temporal logics and branching-time specifications [11, 12], automata-theoretic and finite-state model checking [13, 14], and related algorithmic techniques that underlie industrial temporal verification. Temporal-logic specifications have also been used to verify learning-enabled control, e.g., compositions of neural controllers for LTL tasks under stochastic dynamics [15]; there the networks act as controllers rather than as perception. Separately, neural network certification spans exact SMT/SAT- and MILP-based analysis [16–19] and, most relevant to our pipeline, incomplete but scalable bounds from abstract interpretation, interval propagation, layerwise linear relaxations, and semidefinite relaxations [3–6, 20–26]. Some of the more recent optimization approaches maintain excellent performance in practice but drop theoretical soundness [27]. However, verification of hybrid systems that combine learned perception with symbolic components remains comparatively less developed, especially under end-to-end robustness semantics over long horizons. COLEP [28] certifies such systems under bounded perturbations, but only in the non-temporal case and as conformal coverage of prediction sets rather than an instance-level certificate.

Xie et al. [29] propose a framework for verifying neuro-symbolic properties of neural networks: the network and specification are expressed in a Hoare-style assertion language (NESAL), reduced to deductive verification, and discharged with SMT-based tools such as Marabou [17]. Daggitt et al. [30] extend this direction with Vehicle, a language and compiler for neuro-symbolic programs that bridges problem-space specifications to neural verifiers (again including Marabou) and other proof interfaces.

Akintunde et al. [31] verify temporal strategic properties of multi-agent neuro-symbolic systems: each agent combines a feed-forward perception network with symbolic control logic, and specifications in a bounded fragment of alternating-time temporal logic are compiled to MILP feasibility.

None of the above targets end-to-end temporal robustness certification for NeSyA-style models with probabilistic semantics: Xie et al. [29] and Daggitt et al. [30] lack native temporal operators and probabilistic semantics, Akintunde et al. [31] have temporality but uses non-probabilistic strategic semantics over input-perturbation-free agents, and all three rely on solver-based methods (SMT and MILP) that scale poorly under recursion or unrolling along the temporal axis.

Closest to our setting, Manginas et al. [2] verify a neural network composed with a probabilistic logical layer via scalable relaxation-based bounds, but in the non-temporal case. Our recursive baseline variant is exactly the lift of their construction to NeSyA, transition log-probabilities of the form (6) chained along the log-semiring recurrence (7)–(8); we further strengthen it by replacing their IBP per-step relaxation with the tighter layerwise linear bounds of [3, 4].

6 Conclusions

We studied end-to-end robustness certification for Neurosymbolic Automata (NeSyA) models under L_∞ perturbations of input sequences, with the perception network and probabilistic automaton composed into a single map certified by layerwise linear bounds. Our experiments on NeSyA-MNIST, MNIST-extended, and CAVIAR video event recognition show that contrastive logit-gap

(CLG) transition scores greatly improve certified fractions over a standard log-space baseline that compounds through the temporal recurrence, and that reorganizing temporal evidence, via path aggregation or a fused temporal operator, yields additional gains and substantial wall-time savings as the symbolic system scales (up to $\sim 6\times$ on our largest benchmark).

Limitations. While our fused temporal operator eliminates compounding relaxation error across time, the spatial relaxation of the perception network remains the dominant cost driver: bound propagation through a time-distributed CNN scales as $O(T)$ separate per-frame evaluations, so wall-clock cost grows linearly with sequence length; scaling to deeper perception stacks (e.g., ResNet-class) at long horizons remains future work. Our experiments use small perception networks; scaling to larger networks and radii remains open. Our path-unrolling variant provides tight certificates but can suffer from combinatorial explosion on highly non-deterministic automata. Finally, we target the deterministic-finite-automaton family with calibrated-probability acceptance thresholds; non-deterministic, weighted, or multi-class semantics introduce structural changes that are out of scope for this paper.

Future work. The log-semiring machinery we use is not restricted to deterministic finite automata. A natural next step is extending CLG and the fused temporal operator to non-deterministic/weighted automata, and finite-horizon fragments of LTL or LTL_f. Preserving the convexity and monotonicity structure that powers our exact corner-bounds (Proposition 8) across these broader semantics is a promising theoretical direction. On the practical side, we plan to couple our temporal operators with complementary spatial NeSy verifiers [2] to tackle richer perception stacks. We hope these building blocks make temporal neuro-symbolic certification a primary target for the robustness community.

Acknowledgments and Disclosure of Funding

Abdelrahman Hekal, Vasilis Manginas, and Nikolaos Manginas were supported by the European Union's Horizon research and innovation programme under the EVENFLOW project (Robust Learning and Reasoning for Complex Event Forecasting, grant agreement No. 101070430), Abdelrahman Hekal while at Imperial College London. Vasilis Manginas was additionally funded by the Flemish Government under the "Flanders Artificial Intelligence Research program". Alessio Lomuscio acknowledges partial support from the Royal Academy of Engineering via a Chair of Emerging Technologies. The authors declare no competing interests.

References

[1] Nikolaos Manginas, Georgios Paliouras, and Luc De Raedt. NeSyA: Neurosymbolic automata. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 5950–5958, 2025. doi: 10.24963/ijcai.2025/662.

[2] Vasileios Manginas, Nikolaos Manginas, Edward Stevinson, Sherwin Varghese, Nikos Katzouris, Georgios Paliouras, and Alessio Lomuscio. A scalable approach to probabilistic neuro-symbolic robustness verification. In Leilani H. Gilpin, Eleonora Giunchiglia, Pascal Hitzler, and Emile van Krieken, editors, *Proceedings of the 19th International Conference on Neurosymbolic Learning and Reasoning*, volume 284 of *Proceedings of Machine Learning Research*, pages 52–69. PMLR, 08–10 Sep 2025. URL `https://proceedings.mlr.press/v284/manginas25a.html`.

[3] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/d04863f100d59b3eb688a11f95b0ae60-Paper.pdf`.

[4] Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=nVZtXBI6LNn`.

[5] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2018. doi: 10.1109/SP.2018.00058.

[6] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelović, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models, 2019. URL https://arxiv.org/abs/1810.12715. arXiv:1810.12715.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[8] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[11] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, pages 46–57, 1977. doi: 10.1109/SFCS.1977.32.

[12] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Dexter Kozen, editor, *Logics of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71, Berlin, Heidelberg, 1982. Springer.

[13] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, Cambridge, MA, 1999.

[14] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, Cambridge, MA, 2008.

[15] Jun Wang, Haojun Chen, Zihe Sun, and Yiannis Kantaros. Verified compositional neuro-symbolic control for stochastic systems with temporal logic tasks, 2023. URL https://arxiv.org/abs/2311.10863. arXiv:2311.10863.

[16] Rüdiger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In Deepak D'Souza and K. Narayan Kumar, editors, *Automated Technology for Verification and Analysis*, volume 10482 of *Lecture Notes in Computer Science*, pages 269–286, Cham, 2017. Springer.

[17] Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, David L. Dill, Mykel J. Kochenderfer, and Clark W. Barrett. The Marabou framework for verification and analysis of deep neural networks. In *International Conference on Computer Aided Verification*, volume 11561 of *Lecture Notes in Computer Science*, pages 443–452, Cham, 2019. Springer. URL https://doi.org/10.1007/978-3-030-25540-4_26.

[18] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, volume 10426 of *Lecture Notes in Computer Science*, pages 3–29, Cham, 2017. Springer. URL https://doi.org/10.1007/978-3-319-63387-9_1.

[19] Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward ReLU neural networks, 2017. URL https://arxiv.org/abs/1706.07351. arXiv:1706.07351.

[20] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit S. Dhillon. Towards fast computation of certified robustness for ReLU networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5276–5285, 2018. URL http://proceedings.mlr.press/v80/weng18a.html.

[21] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):41:1–41:30, January 2019. doi: 10.1145/3290354.

[22] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Skxuk1rFwB.

[23] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=ahYIlRBeCFw.

[24] Panagiotis Kouvaros, Benedikt Brückner, Patrick Henriksen, and Alessio Lomuscio. Dynamic back-substitution in bound-propagation-based neural network verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27383–27391, 2025. doi: 10.1609/aaai.v39i26.34949.

[25] Jianglin Lan, Yang Zheng, and Alessio Lomuscio. Iteratively enhanced semidefinite relaxations for efficient neural network verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14937–14945, 2023. doi: 10.1609/aaai.v37i12.26744.

[26] Hong-Ming Chiu, Hao Chen, Huan Zhang, and Richard Y. Zhang. SDP-CROWN: Efficient bound propagation for neural network verification with tightness of semidefinite programming. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 10449–10468, 2025. URL https://proceedings.mlr.press/v267/chiu25a.html.

[27] Yanghao Zhang, Panagiotis Kouvaros, and Alessio Lomuscio. Scalable neural network geometric robustness validation via Hölder optimisation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0458.

[28] Mintong Kang, Nezihe Merve Gürel, Linyi Li, and Bo Li. COLEP: Certifiably robust learning-reasoning conformal prediction via probabilistic circuits. In *International Conference on Learning Representations*, 2024.

[29] Xuan Xie, Kristian Kersting, and Daniel Neider. Neuro-symbolic verification of deep neural networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 3622–3628, 2022. URL https://doi.org/10.24963/ijcai.2022/503.

[30] Matthew L. Daggitt, Wen Kokke, Robert Atkey, Ekaterina Komendantskaya, Natalia Ślusarz, and Luca Arnaboldi. Vehicle: Bridging the embedding gap in the verification of neuro-symbolic programs (invited talk). In *10th International Conference on Formal Structures for Computation and Deduction (FSCD 2025)*, volume 337 of *Leibniz International Proceedings in Informatics*, pages 2:1–2:20, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL https://doi.org/10.4230/LIPIcs.FSCD.2025.2.

[31] Michael E. Akintunde, Elena Botoeva, Panagiotis Kouvaros, and Alessio Lomuscio. Verifying strategic abilities of neural-symbolic multi-agent systems. In *Proceedings of the Seventeenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 22–32, 2020. URL https://doi.org/10.24963/kr.2020/3.

[32] James K. Baker. Trainable grammars for speech recognition. In *Proceedings of the Spring Conference of the Acoustical Society of America*, pages 547–550, 1979.

[33] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[34] Joshua Goodman. Semiring parsing. *Computational Linguistics*, 25(4):573–605, 1999.

[35] Jason Eisner. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 1–17, 2016.

A Full proofs

This appendix proves the claims referenced from Section 3 (transition scores and CLG in Section 3.2, encodings in Section 3.3). Section A.1 treats CLG versus the baseline transition score as compositions of layerwise linear bounds: it records pointwise equivalence, establishes the support structure of the composed linear coefficients, analyzes composition with the outer $-\,\mathrm{softplus}$ map (including its asymmetric confident regimes), and exhibits a $K{=}2$ instance where the CLG pipeline is strictly tighter, together with a $K{=}3$ numeric cross-check of the outer-layer composition rule. Section A.2 proves convexity and coordinatewise monotonicity of the log-semiring functional and derives corner-tight bounds for the fused temporal operator, then counts forward work across encodings.

A.1 CLG versus the baseline transition score

Notation. Fix $K \geq 2$, a nonempty proper subset $\mathcal{S} \subsetneq \{1, \ldots, K\}$, and an axis-aligned input box $\mathcal{Z} = \prod_{i=1}^{K} [\tilde{z}_{l,i}, \tilde{z}_{u,i}] \subset \mathbb{R}^K$. Write $\mathcal{S}^{\mathrm{c}} := \{1, \ldots, K\} \setminus \mathcal{S}$ and $\mathcal{I} := \{1, \ldots, K\}$ (the full joint-index alphabet, used here as a subscript-friendly local name). For $\mathcal{T} \subseteq \mathcal{I}$ let $\mathrm{LSE}_{\mathcal{T}}(\tilde{z}) := \log \sum_{i \in \mathcal{T}} e^{\tilde{z}_i}$, i.e. $\mathrm{LSE}_{\mathcal{T}}(\tilde{z}) = \mathrm{LSE}(\tilde{z}; \mathcal{T})$ in the notation of Section 2; the restricted softmax mass vector is $\sigma_{\mathcal{T}}(\mu)_i := e^{\mu_i} / \sum_{j \in \mathcal{T}} e^{\mu_j}$ for $i \in \mathcal{T}$ and 0 otherwise. We write $\varsigma(t) := 1/(1 + e^{-t})$ for the logistic sigmoid, to distinguish it from $\sigma_{\mathcal{T}}$ (restricted softmax). $\mathrm{softplus}(t) = \log(1 + e^t)$. The baseline score on $\mathcal{S}$ is $s(\tilde{z}) := \mathrm{LSE}_{\mathcal{S}}(\tilde{z}) - \mathrm{LSE}_{\mathcal{I}}(\tilde{z})$ and the CLG score is $s_{\mathrm{clg}}(\tilde{z}) := -\mathrm{softplus}\big(\mathrm{LSE}_{\mathcal{S}^{\mathrm{c}}}(\tilde{z}) - \mathrm{LSE}_{\mathcal{S}}(\tilde{z})\big)$.

Lemma 1 (Pointwise equivalence)**.** *$s(\tilde{z}) = s_{\mathrm{clg}}(\tilde{z})$ for all $\tilde{z} \in \mathbb{R}^K$.*

Proof. With $A = \sum_{i \in \mathcal{S}} e^{\tilde{z}_i}$ and $B = \sum_{i \in \mathcal{S}^{\mathrm{c}}} e^{\tilde{z}_i}$, both sides equal $-\log(1 + B/A)$. □

Linear relaxation of $\mathrm{LSE}_{\mathcal{T}}$ on a box. $\mathrm{LSE}_{\mathcal{T}}$ is convex in $\tilde{z}$, so a tangent at any anchor $\mu_{\mathcal{T}} \in \mathcal{Z}$ is a sound linear lower bound with coefficient vector supported on $\mathcal{T}$:

$$\mathrm{LSE}_{\mathcal{T}}(\tilde{z}) \;\geq\; \sigma_{\mathcal{T}}(\mu_{\mathcal{T}})^{\top} \tilde{z} + c_L(\mu_{\mathcal{T}}, \mathcal{T}), \qquad c_L(\mu_{\mathcal{T}}, \mathcal{T}) := \mathrm{LSE}_{\mathcal{T}}(\mu_{\mathcal{T}}) - \sigma_{\mathcal{T}}(\mu_{\mathcal{T}})^{\top} \mu_{\mathcal{T}}. \tag{14}$$

A sound linear upper bound has the form

$$\mathrm{LSE}_{\mathcal{T}}(\tilde{z}) \;\leq\; \bar{a}_{\mathcal{T}}^{\top} \tilde{z} + \bar{b}_{\mathcal{T}}, \qquad (\bar{a}_{\mathcal{T}})_i = 0 \text{ for } i \notin \mathcal{T}, \tag{15}$$

for all $\tilde{z} \in \mathcal{Z}$. Two canonical constructions satisfy (15): an interval upper $\bar{a}_{\mathcal{T}} = 0$, $\bar{b}_{\mathcal{T}} = \mathrm{LSE}_{\mathcal{T}}(\tilde{z}_u^{(\mathcal{T})})$ (valid because $\mathrm{LSE}_{\mathcal{T}}$ is coordinatewise nondecreasing in $\tilde{z}_i$ for $i \in \mathcal{T}$), or a corner chord tight at two opposite box corners of the $\mathcal{T}$-coordinates. The statements below depend only on (14)–(15); zero coefficients on $i \notin \mathcal{T}$ are the only property of the upper envelope we use.

Theorem 2 (Soundness of both pipelines under layerwise linear bounds)**.** *Under* (14)–(15) *applied to each* LSE *head, both the baseline pipeline ($s = \mathrm{LSE}_{\mathcal{S}} - \mathrm{LSE}_{\mathcal{I}}$) and the CLG pipeline (inner $g := \mathrm{LSE}_{\mathcal{S}^{\mathrm{c}}} - \mathrm{LSE}_{\mathcal{S}}$, outer $-\mathrm{softplus}$) produce sound linear lower and upper bounds on $s(\tilde{z})$ uniformly over $\tilde{z} \in \mathcal{Z}$.*

Proof. Linear bounds compose through subtraction: $\underline{f} - \overline{g} \leq f - g \leq \overline{f} - \underline{g}$ whenever $\underline{f} \leq f \leq \overline{f}$ and $\underline{g} \leq g \leq \overline{g}$ on $\mathcal{Z}$. This yields linear bounds on s (baseline) and on g (inner CLG). For the outer CLG layer, $-\mathrm{softplus}$ is concave and nonincreasing; its tangent at any $g^* \in [\underline{g}, \overline{g}]$ is a linear upper bound, and the chord between $(\underline{g}, -\mathrm{softplus}(\underline{g}))$ and $(\overline{g}, -\mathrm{softplus}(\overline{g}))$ is a linear lower bound. Composing these with linear bounds on g (flipping signs because the outer slope is negative) yields linear bounds on s_{clg}; by Lemma 1 these bound s as well. □

Lemma 3 (Support of composed linear bounds)**.** *Let $\bar{a}^{\mathrm{B}} \in \mathbb{R}^K$ be the coefficient vector of the linear upper bound on the baseline score s obtained by combining* (14)–(15) *through the subtraction rule, with anchors $\mu_{\mathcal{S}}, \mu_{\mathcal{I}} \in \mathcal{Z}$ for the tangent lowers. Let $\bar{a}^{\mathrm{g}} \in \mathbb{R}^K$ be the analogous coefficient vector of the linear upper bound on $g = \mathrm{LSE}_{\mathcal{S}^{\mathrm{c}}} - \mathrm{LSE}_{\mathcal{S}}$, with anchors $\mu_{\mathcal{S}^{\mathrm{c}}}, \mu_{\mathcal{S}}$. Then*

$$(\bar{a}^{\mathrm{B}})_i = \begin{cases} (\bar{a}_{\mathcal{S}})_i - \sigma_{\mathcal{I}}(\mu_{\mathcal{I}})_i & i \in \mathcal{S}, \\ -\sigma_{\mathcal{I}}(\mu_{\mathcal{I}})_i & i \in \mathcal{S}^{\mathrm{c}}, \end{cases} \qquad (\bar{a}^{\mathrm{g}})_i = \begin{cases} -\sigma_{\mathcal{S}}(\mu_{\mathcal{S}})_i & i \in \mathcal{S}, \\ (\bar{a}_{\mathcal{S}^{\mathrm{c}}})_i & i \in \mathcal{S}^{\mathrm{c}}. \end{cases} \tag{16}$$

In particular, every coordinate of $\bar{a}^{\mathrm{B}}$ is generically nonzero (both terms on $\mathcal{S}$, and $-\sigma_{\mathcal{I}}(\mu_{\mathcal{I}})_i < 0$ on $\mathcal{S}^{\mathrm{c}}$); each coordinate of $\bar{a}^{\mathrm{g}}$ is drawn from a single LSE *relaxation. This "dense" nonzero pattern is specific to the upper linear bound on s under the subtraction rule above; the corresponding lower bound on s can be sparse in coordinates (e.g., in the K=2 worked example below, the baseline lower uses only $\tilde{z}_1$ under tangent lowers and an interval upper on $\mathrm{LSE}_{\mathcal{I}}$). The analogous coefficient formulas hold for lower bounds with the roles of* (14) *and* (15) *swapped on the subtracted head.*

Proof. Direct substitution into the subtraction rule $\bar{a}_{f-g} = \bar{a}_f - \underline{a}_g$, using that tangent lowers have coefficient $\sigma_{\mathcal{T}}(\mu_{\mathcal{T}})$ supported on $\mathcal{T}$ and upper envelopes have coefficient $\bar{a}_{\mathcal{T}}$ supported on $\mathcal{T}$. Over the full alphabet $\mathcal{T} = \mathcal{I}$ we have $\sigma_{\mathcal{I}}(\mu_{\mathcal{I}})_i > 0$ for every $i \in \mathcal{I}$, which fills every coordinate of $\bar{a}^{\mathrm{B}}$. □

Outer $-\operatorname{softplus}$ **composition.** Suppose the inner CLG step yields $\underline{a}^{\mathrm{g}\top}\tilde{z}+\underline{b}^{\mathrm{g}} \le g(\tilde{z}) \le \bar{a}^{\mathrm{g}\top}\tilde{z}+\bar{b}^{\mathrm{g}}$ on $\mathcal{Z}$, with induced range $[\underline{g}, \overline{g}]$. The tangent of $-\operatorname{softplus}$ at anchor $g^* \in [\underline{g}, \overline{g}]$ is a linear upper bound $-\operatorname{softplus}(g) \le -\operatorname{softplus}(g^*) - \varsigma(g^*)\,(g - g^*)$; because the slope $-\varsigma(g^*)$ is negative, composing with the lower linear bound on g yields a linear upper on s_{clg} with coefficient vector $-\varsigma(g^*)\,\underline{a}^{\mathrm{g}}$. The chord between $(\underline{g}, -\operatorname{softplus}(\underline{g}))$ and $(\overline{g}, -\operatorname{softplus}(\overline{g}))$ has slope $-\gamma$ where $\gamma = (\operatorname{softplus}(\overline{g}) - \operatorname{softplus}(\underline{g}))/(\overline{g} - \underline{g}) \in (0, 1)$; symmetrically, composing this chord lower of $-\operatorname{softplus}$ with the upper linear bound on g yields a linear lower on s_{clg} with coefficient vector $-\gamma\,\bar{a}^{\mathrm{g}}$. By the mean value theorem, $\gamma \in [\min\{\varsigma(\underline{g}), \varsigma(\overline{g})\}, \max\{\varsigma(\underline{g}), \varsigma(\overline{g})\}]$. When $\underline{g} = \overline{g}$ the chord degenerates: it coincides with the tangent at that single value and the outer relaxation is exact.

Theorem 4 (Support and outer $-\operatorname{softplus}$ composition)**.** *Under* (14)–(15)*, the linear bounds on s produced by the two pipelines satisfy:*

1. (Support.) *The CLG linear bounds on g have disjoint coordinate support across the two differenced heads (Lemma 3, right); the baseline linear bounds on s are supported on $\mathcal{I} = \mathcal{S} \sqcup \mathcal{S}^{\mathrm{c}}$, with residual coefficients on $\mathcal{S}$ coming from the shared-coordinate subtraction (Lemma 3, left).*

2. ($-\operatorname{softplus}$ layer.) *The linear bounds on s_{clg} compose those on g with tangent/chord relaxations of $-\operatorname{softplus}$, multiplying the inner coefficient vectors by scalars in $(-1, 0)$ (namely $-\varsigma(g^*)$ for the tangent upper bound on $-\operatorname{softplus}$ and $-\gamma$ for the chord lower), hence scaling inner coefficient magnitudes relative to propagating bounds on g without this layer; when $\varsigma(g^*)$ or γ is bounded away from 1, the scaling strictly contracts coordinate magnitudes, whereas as $g^* \to +\infty$ these multipliers can approach -1 and the outer map is asymptotically linear (no contraction in the limit). As $g^* \to -\infty$ (equivalently $p_{\mathcal{S}} \to 1$), $\varsigma(g^*) \to 0$ and these outer multipliers damp toward 0. As $g^* \to +\infty$ (equivalently $p_{\mathcal{S}} \to 0$), $\varsigma(g^*) \to 1$ instead: the tangent multiplier approaches -1, and tightness of the outer relaxation reflects that $-\operatorname{softplus}(g)$ is asymptotically linear with slope -1 (equivalently $\operatorname{softplus}(g) - g \to 0$), not vanishing derivatives. The baseline composition has no analogous scalar outer map: its outermost operation is a linear subtraction, which preserves inner coefficient magnitudes.*

Proof. (1) is Lemma 3. (2) is the composition in the preceding paragraph. For limits, $\varsigma(g^*) \to 0$ as $g^* \to -\infty$ and $\varsigma(g^*) \to 1$ as $g^* \to +\infty$; when $g^* \to +\infty$, $\operatorname{softplus}(g^*) - g^* \to 0$, so $-\operatorname{softplus}$ matches its linearization $-g$ in the limit. □

Worked example (K=2). Let $K = 2$, $\mathcal{S} = \{1\}$, $\mathcal{Z} = [-\varepsilon, \varepsilon]^2$, with anchor $\mu = (0, 0)$ for every tangent and the interval upper for (15). The exact range of $s(\tilde{z}) = \log\varsigma(\tilde{z}_1 - \tilde{z}_2)$ on $\mathcal{Z}$ is $[-\log 2 - \varepsilon + O(\varepsilon^2),\ -\log 2 + \varepsilon + O(\varepsilon^2)]$, width $2\varepsilon + O(\varepsilon^2)$.

Baseline pipeline. $\mathrm{LSE}_{\mathcal{S}}(\tilde{z}) = \tilde{z}_1$ is exact. $\mathrm{LSE}_{\mathcal{I}}(\tilde{z}) \ge \frac{1}{2}\tilde{z}_1 + \frac{1}{2}\tilde{z}_2 + \log 2$ (tangent at 0) and $\mathrm{LSE}_{\mathcal{I}}(\tilde{z}) \le \log 2 + \varepsilon$ (interval at $\tilde{z}_u = (\varepsilon, \varepsilon)$). By Lemma 3 the linear bounds on s are $\bar{s}^{\mathrm{B}}(\tilde{z}) = \frac{1}{2}\tilde{z}_1 - \frac{1}{2}\tilde{z}_2 - \log 2$ and $\underline{s}^{\mathrm{B}}(\tilde{z}) = \tilde{z}_1 - \log 2 - \varepsilon$, with ranges $[-\varepsilon - \log 2, \varepsilon - \log 2]$ and $[-2\varepsilon - \log 2, -\log 2]$ over $\mathcal{Z}$. The induced enclosure is $[-2\varepsilon - \log 2,\ \varepsilon - \log 2]$, width 3ε.

CLG pipeline. $g(\tilde{z}) = \tilde{z}_2 - \tilde{z}_1$ is exact with $[\underline{g}, \overline{g}] = [-2\varepsilon, 2\varepsilon]$. Tangent upper of $-\operatorname{softplus}$ at $g^* = 0$: $-\operatorname{softplus}(g) \le -\log 2 - \frac{1}{2}g$. Chord slope $\gamma = (\operatorname{softplus}(2\varepsilon) - \operatorname{softplus}(-2\varepsilon))/(4\varepsilon) = \frac{1}{2}$ exactly, since $\operatorname{softplus}(x) - \operatorname{softplus}(-x) = x$; the chord intercept then carries the $O(\varepsilon^2)$ correction, and the chord lower is $-\operatorname{softplus}(g) \ge -\frac{1}{2}g - \log 2 - \frac{\varepsilon^2}{2} + O(\varepsilon^4)$. Substituting $g = \tilde{z}_2 - \tilde{z}_1$: $\bar{s}^{\mathrm{C}}(\tilde{z}) = -\frac{1}{2}(\tilde{z}_2 - \tilde{z}_1) - \log 2$ and $\underline{s}^{\mathrm{C}}(\tilde{z}) = -\frac{1}{2}(\tilde{z}_2 - \tilde{z}_1) - \log 2 - \frac{\varepsilon^2}{2}$. Their ranges over $\mathcal{Z}$ give the enclosure $[-\log 2 - \varepsilon - \frac{\varepsilon^2}{2},\ -\log 2 + \varepsilon]$, width $2\varepsilon + O(\varepsilon^2)$ and matching the exact range to leading order.

Nondegeneracy. Here $|\mathcal{S}^{\mathrm{c}}| = 1$, so $g(\tilde{z}) = \tilde{z}_2 - \tilde{z}_1$ is affine; with exact inner heads the induced linear lower and upper bounds on g agree as linear functions on $\mathcal{Z}$ (equivalently $\underline{a}^{\mathrm{g}} = \bar{a}^{\mathrm{g}}$). The outer $-\operatorname{softplus}$ step then multiplies a single inner slope vector by $-\varsigma(g^*)$ and $-\gamma$, so the correct pairing "tangent upper with the inner lower on g, chord lower with the inner upper on g" cannot be distinguished at the level of coefficient vectors from swapping those inner bounds; both pairings propagate the same slopes. The K=3 instance below separates those slopes and makes the pairing observable.

Summary. At equal anchors and the interval upper, the CLG enclosure width is $\frac{2}{3}$ that of the baseline to leading order in ε. The two pipelines yield the same upper linear bound on s here ($\bar{s}^{\mathrm{B}} = \bar{s}^{\mathrm{C}} = \frac{1}{2}\tilde{z}_1 - \frac{1}{2}\tilde{z}_2 - \log 2$); the gap arises entirely in the lower bound. The baseline lower $\tilde{z}_1 - \log 2 - \varepsilon$ depends only on $\tilde{z}_1$, because the interval upper on $\mathrm{LSE}_{\mathcal{I}}$ contributes a constant and severs the $\tilde{z}_2$ dependence (the lower-side sparsity flagged in Lemma 3); it ranges down to $-\log 2 - 2\varepsilon$. The CLG chord lower retains the contrastive direction $\tilde{z}_2 - \tilde{z}_1$ in its slope, since the inner gap g couples both coordinates and the chord slope $-\gamma$ propagates through both; the chord-versus-tangent slack contributes only $O(\varepsilon^2)$. On this instance the structural advantage of CLG is therefore concentrated in the lower bound rather than the upper.

Outer-composition cross-check (K=3). Let $K = 3$, $\mathcal{S} = \{1\}$, $\mathcal{S}^{\mathrm{c}} = \{2, 3\}$, and $\mathcal{Z} = [-0.4, 0.4]^3$, still with tangents at $\mu = 0$ on every LSE head and the interval upper for (15) on each head (for $\mathrm{LSE}_{\mathcal{S}^{\mathrm{c}}}$, the maximizing corner is $(\tilde{z}_2, \tilde{z}_3) = (0.4, 0.4)$). Then $\mathrm{LSE}_{\mathcal{S}}(\tilde{z}) = \tilde{z}_1$ exactly, while $\mathrm{LSE}_{\mathcal{S}^{\mathrm{c}}}(\tilde{z}) \geq \frac{1}{2}\tilde{z}_2 + \frac{1}{2}\tilde{z}_3 + \log 2$ and $\mathrm{LSE}_{\mathcal{S}^{\mathrm{c}}}(\tilde{z}) \leq \log 2 + 0.4$. Lemma 3 yields different inner slopes, $\underline{a}^{\mathrm{g}} = (-1, \frac{1}{2}, \frac{1}{2})$ and $\bar{a}^{\mathrm{g}} = (-1, 0, 0)$, whereas the exact gap $g(\tilde{z}) = \mathrm{LSE}_{\mathcal{S}^{\mathrm{c}}}(\tilde{z}) - \tilde{z}_1$ ranges over $[\underline{g}, \overline{g}] = [\log 2 - 0.8,\ \log 2 + 0.8]$ on $\mathcal{Z}$. Take $g^* = 0 \in [\underline{g}, \overline{g}]$ in the tangent upper on $-\mathrm{softplus}$, so $\varsigma(g^*) = \frac{1}{2}$. For the chord lower, $\gamma = (\mathrm{softplus}(\overline{g}) - \mathrm{softplus}(\underline{g}))/(\overline{g} - \underline{g}) \approx 0.659$. Sound propagation gives a linear upper bound on s_{clg} with coefficient vector $-\varsigma(g^*)\underline{a}^{\mathrm{g}} = (\frac{1}{2}, -\frac{1}{4}, -\frac{1}{4})$ and a linear lower bound with $-\gamma\bar{a}^{\mathrm{g}} \approx (0.659, 0, 0)$. In particular, incorrectly pairing the tangent with $\bar{a}^{\mathrm{g}}$ would yield $(\frac{1}{2}, 0, 0)$ and drop the required $(\tilde{z}_2, \tilde{z}_3)$ coupling from the inner lower bound on g.

Remark 5 (Not a universal dominance inequality). The K=2 instance above gives a concrete strict improvement for CLG at equal anchors and the interval upper. Whether CLG or baseline yields a narrower enclosure in a given instance depends on the anchors $\mu_{\mathcal{S}}, \mu_{\mathcal{I}}, \mu_{\mathcal{S}^{\mathrm{c}}}, g^*$, on the choice of upper envelope (15), and on the box geometry; disjoint support and the outer $-\mathrm{softplus}$ composition do not imply a universal inequality $\mathrm{slack}(\mathrm{CLG}) \leq \mathrm{slack}(\mathrm{baseline})$. We claim structural differences in the linear bounds (Theorem 4) and a concrete strict improvement at equal anchors (the worked example), not universal dominance. Downstream composition through the recurrence (7)–(8) and the CNN relaxations can interact with either pipeline.

Remark 6 (Marginal-interval special case). When every upper envelope is taken as a constant (interval value) and every lower as a tangent, the pipelines in Theorem 2 specialize to marginal interval arithmetic: each LSE head has a scalar enclosure $[L, U]$ and differences use $[L, U] - [L', U'] = [L - U', U - L']$. In this regime the support argument of Lemma 3 collapses (all linear coefficients are zero), and the two pipelines differ only in which pair of marginal intervals is subtracted; Theorem 4 is strictly more informative whenever the propagator retains nontrivial linear coefficients (as in backward linear relaxation of hidden layers).

Per-pair scatter (supplementary visualization of Section 4.1). The regime-stratified histograms in Figure 2 aggregate per-pair widths within each regime; for completeness we include the raw scatter of $(w_{\mathrm{baseline}}, w_{\mathrm{CLG}})$ per (image, transition) pair below.

Per-path composition arithmetic (mechanism behind Section 4.1). Per-step CLG savings on confidently-SAT transitions are large in absolute log-space (medians at the floor; $\sim$8 log-units of slack relative to baseline at ε=0.01), and per-step CLG costs on confidently-UNSAT transitions are small ($\sim$1 log-unit). On any accepting path of length T, transitions partition into roughly $T - u$ SAT and $u \in \{1, 2\}$ UNSAT contributions, with the u UNSAT contributions dominated by very negative true log-probabilities and the $T - u$ SAT contributions accumulating per-step slack across the recurrence. Composing through the log-semiring recurrence, the per-path CLG advantage scales as $\sim 8 \cdot (T - u)$ versus a per-path CLG cost of $\sim 1 \cdot u$; for $T \sim 10$ and $u \in \{1, 2\}$ the SAT savings dominate the UNSAT cost by an order of magnitude or more, and the resulting per-path bounds tighten on both sides.

A.2 Temporal encodings: convexity, corner bounds, and forward cost

Let $M := |E|$ be the number of automaton transitions, and stack transition scores as $s = (s_{t,m})_{t \in \{1,\ldots,T\},\, m \in E} \in \mathbb{R}^{TM}$. Write $F(s) := \log \mathbb{P}_{\mathrm{acc}}$ for the log-semiring functional in (7)–(8) viewed as a function of s. Let $\mathcal{P}_{\mathrm{acc}}$ denote the set of accepting state paths of length T; each $p \in \mathcal{P}_{\mathrm{acc}}$ selects one transition $m_{p,t}$ at each t. The path-form identity, convexity and monotonicity of F, and the edge-marginal form of ∇F are classical in structured prediction [8, 32–35]; we record the statements here in the form used by Section 3.3 for the corner-tight bound.

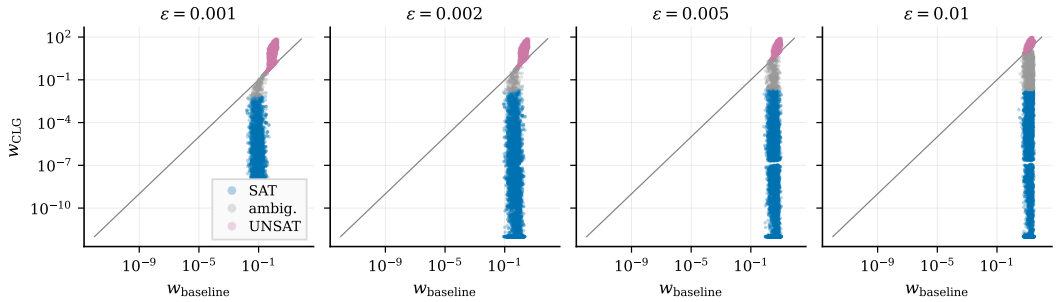

Figure 4. Supplementary per-pair scatter for the bound-width study of Section 4.1: $(w_{\text{baseline}}, w_{\text{CLG}})$ on log–log axes, one panel per ε, with each point colored by the transition's clean-image regime (read off the midpoint of the CLG bound): *SAT* blue ($p_{\mathcal{S}_m} > 0.99$), *ambiguous* gray ($0.01 < p_{\mathcal{S}_m} < 0.99$), *UNSAT* pink ($p_{\mathcal{S}_m} < 0.01$). The diagonal $y = x$ marks "no improvement." SAT pairs descend by many orders of magnitude below the diagonal (CLG essentially exact); UNSAT pairs sit just above the diagonal (CLG mildly looser); ambiguous pairs hug the diagonal (no advantage). The same data underlies Figure 2.

Lemma 7 (Convexity and coordinatewise monotonicity of F)**.** *Assume $\mathcal{P}_{\text{acc}} \neq \emptyset$, i.e., the automaton admits at least one accepting trajectory of length T from q_0 (otherwise $\mathbb{P}_{\text{acc}} \equiv 0$ and $F \equiv -\infty$ identically, in which case the certification problem is trivial under the convention $\log 0 = -\infty$). Then $F : \mathbb{R}^{TM} \to \mathbb{R}$ is convex and coordinatewise nondecreasing. Its gradient is the edge-marginal vector: for $(t, m) \in \{1, \dots, T\} \times E$,*

$$\frac{\partial F}{\partial s_{t,m}}(s) \;=\; \sum_{p \in \mathcal{P}_{\text{acc}}:\, m_{p,t} = m} \frac{\exp\big(\sum_\tau s_{\tau, m_{p,\tau}}\big)}{\sum_{p' \in \mathcal{P}_{\text{acc}}} \exp\big(\sum_\tau s_{\tau, m_{p',\tau}}\big)} \;\in\; [0, 1]. \tag{17}$$

Proof. Because the automaton is deterministic, the set $\mathcal{P}_{\text{acc}}$ of accepting state paths coincides with the set of accepting transition sequences and

$$F(s) \;=\; \text{LSE}_{p \in \mathcal{P}_{\text{acc}}} \Big(\textstyle\sum_{t=1}^{T} s_{t, m_{p,t}} \Big), \tag{18}$$

since (7)–(8) sum acceptance mass over all accepting paths. Each path score $\eta_p(s) = \sum_t s_{t,m_{p,t}}$ is linear in s, and LSE on $\mathbb{R}^{|\mathcal{P}_{\text{acc}}|}$ is convex and coordinatewise nondecreasing; composing with a linear map preserves convexity, and composing with a coordinatewise nondecreasing linear map with nonnegative-sign entries $\partial \eta_p / \partial s_{t,m} \in \{0, 1\}$ preserves monotonicity in each $s_{t,m}$. The gradient formula (17) is the chain rule: $\partial F / \partial s_{t,m} = \sum_p \pi_p(s)\, \mathbf{1}[m_{p,t} = m]$ where $\pi_p(s) = \exp(\eta_p - \text{LSE}(\eta))$ is the softmax over path scores; it equals the edge marginal at (t, m) under the log-semiring, and is the standard inside–outside / forward–backward gradient of a log-partition function over a fixed combinatorial support [32, 33, 35]. □

Proposition 8 (Corner-tight bounds for the fused temporal operator)**.** *Let $[s^\ell, s^u] \subset \mathbb{R}^{TM}$ be a box. Then*

$$F(s^\ell) + \nabla F(s^\ell)^\top (s - s^\ell) \;\leq\; F(s) \;\leq\; F(s^u), \qquad s \in [s^\ell, s^u], \tag{19}$$

where $\nabla F(s^\ell)$ is given by (17)*. The lower bound is linear in s and tight at $s = s^\ell$; the upper bound is a constant (interval value) and tight at $s = s^u$. A nontrivial linear upper envelope (e.g., a chord through two box corners) could be tighter in the interior of $[s^\ell, s^u]$, but it requires choosing a slope vector; the constant $F(s^u)$ is the natural interval upper induced by coordinatewise monotonicity, is tight at the worst corner s^u, and avoids that additional choice.*

Proof. The lower bound is the tangent of a convex function at s^ℓ, which is a global affine minorant of F. Tightness at s^ℓ is by construction. For the upper bound, coordinatewise monotonicity gives $F(s) \leq F(s^u)$ for all $s \in [s^\ell, s^u]$ since increasing any coordinate cannot decrease F; tightness is by evaluation at s^u. □

Remark 9 (Slack accumulation in the recursive encoding). Under layerwise linear bound propagation, the recursive encoding (7) exposes T sequential LSE-shaped aggregations to the propagator: at each

t, the Bellman update is a vector LSE over the incoming edges per target state, and each such LSE is linearized on the interval of its inputs by a tangent lower and a valid upper envelope. Denote by $\Delta_t \geq 0$ the slack of the linearization at time t (the gap between the upper and lower envelopes over the current box at that layer). Heuristically, each stage contributes its own relaxation gap and composing T sequential LSE linearizations tends to compound slack along the chain rather than collapsing to a single corner evaluation; we do not claim a sharp inequality of the form "output gap $= \sum_t \Delta_t$" without additional assumptions on the propagator. Proposition 8 sidesteps this layerwise accumulation by linearizing the composite map F once, using convexity and monotonicity to place the tangent at the lower box corner and the interval upper at the upper box corner.

Theorem 10 (Forward arithmetic of temporal encodings)**.**
(A) *Recursive dynamic programming. Computing* $\log \mathbb{P}_{\mathrm{acc}}$ *via* (7)–(8) *requires* $\Theta(TM)$ *score evaluations and* $\Theta(T)$ *vector* LSE *aggregations, each over at most* $|Q|$ *incoming terms per target state.*

(B) *Path aggregation. Forming each path logit* $\eta_p = \sum_{t=1}^{T} s_{t,m_{p,t}}$ *for* $p \in \mathcal{P}_{\mathrm{acc}}$ *costs* $\Theta(T)$ *additions per path, hence* $\Theta(T|\mathcal{P}_{\mathrm{acc}}|)$ *additions in total; the final acceptance* $\mathrm{LSE}(\eta)$ *is one* LSE *over* $|\mathcal{P}_{\mathrm{acc}}|$ *arguments.*

(C) *Fused temporal operator. Forward evaluation of* F *still executes the log-semiring recurrence internally and remains* $\Theta(TM)$ *in score work plus* $\Theta(T)$ *internal* LSE *updates at automaton width* $\Theta(|Q|)$*; the lower bound in Proposition 8 requires one additional forward–backward pass to compute* $\nabla F(s^\ell)$ *via* (17)*, at the same* $\Theta(TM)$ *cost.*

Proof. (A)–(B) by direct counting. (C): same forward recurrence as (A) for $F(s^\ell)$ and $F(s^u)$; $\nabla F(s^\ell)$ is obtained by one backward pass over the DP graph, matching the forward cost up to a constant factor; this is the standard semiring forward–backward / inside–outside computation [32, 34, 35]. □

Closing remarks. Theorem 4 isolates the per-step mechanism behind CLG (disjoint support and composition with $-\,\mathrm{softplus}$, damping coefficients when $p_{\mathcal{S}} \to 1$ and asymptotic linearity when $p_{\mathcal{S}} \to 0$); Theorem 2 records soundness of either parameterization. Neither implies universal dominance in end-to-end certified fractions: composition with the recurrence (7)–(8) and the CNN relaxations can favor either pipeline on a given instance (Remark 5). Lemma 7 and Proposition 8 reuse classical structured-prediction facts (the path-form / log-partition view, convexity and coordinatewise monotonicity, and the edge-marginal form of the gradient via inside–outside / forward–backward [8, 32–35]) to give the fused-temporal-operator bounds a corner-tight structure that does not compound across T, in contrast to the recursive encoding's T layerwise linearizations (Remark 9).

B Simplex tightening: proofs and experiments

This appendix formalizes the transition and state simplex refinements, proves that they yield sound interval refinements under stated automaton semantics, and records how they compose with the fused temporal-operator bound from Section 3.3. The refinements are orthogonal to CLG (Section 3.2). Transition tightening applies Lemma 11 at outgoing transition groups; bidirectional mode additionally applies Lemma 12 at exhaustive states. State simplex applies Lemma 13 in the interval forward DP. In the main structural comparison (Table 1), all of these refinements are disabled; Table 3 turns them on selectively.

B.1 Transition simplex

Fix a source state q and let $\mathrm{Out}(q)$ index transitions with source q. At a fixed timestep, let $(p_m)_{m\in\mathrm{Out}(q)}$ denote true outgoing transition probabilities and (s_m) the corresponding CLG transition log-scores, so $p_m = e^{s_m}$ on the modeled edges. NeSyA semantics imply $p_m \geq 0$ and

$$\sum_{m\in\mathrm{Out}(q)} p_m \leq 1, \tag{20}$$

with equality when every joint assignment leaving q is captured by some listed transition (exhaustivity at q). Suppose a sound bounding procedure supplies independent intervals $s_m \in [s_m^\ell, s_m^u]$ for the image of the input perturbation ball (every realized score lies in its interval). Intersecting the orthotope with the implied constraints from (20) yields a (possibly) smaller sound box.

Lemma 11 (Upper tightening from sub-probability)**.** *Assume* (20) *and fix* $m \in \mathrm{Out}(q)$*, with* $s_j \geq s_j^\ell$ *for all* $j \in \mathrm{Out}(q) \setminus \{m\}$*. If* $\sum_{j \in \mathrm{Out}(q)\setminus\{m\}} e^{s_j^\ell} < 1$*, then*

$$s_m \;\leq\; \log\Bigl(1 - \sum_{j \in \mathrm{Out}(q)\setminus\{m\}} e^{s_j^\ell}\Bigr). \tag{21}$$

If instead $\sum_{j \neq m} e^{s_j^\ell} \geq 1$*, the same derivation yields* $p_m \leq 0$*, forcing* $p_m = 0$ *(equivalently* $s_m = -\infty$*); the supplied lower bounds are jointly incompatible with any positive mass on* m*, and no finite valid upper cap on* s_m *follows from* (20) *alone.*

Proof. For each $j \neq m$, $p_j = e^{s_j} \geq e^{s_j^\ell}$, hence $\sum_{j \neq m} p_j \geq \sum_{j \neq m} e^{s_j^\ell}$. From $p_m = 1 - \sum_{j \neq m} p_j - r$ with $r \geq 0$ accounting for mass not modeled on $\mathrm{Out}(q)$, $p_m \leq 1 - \sum_{j \neq m} p_j \leq 1 - \sum_{j \neq m} e^{s_j^\ell}$. When the right-hand side is positive, monotonicity of $\log$ yields (21); otherwise it yields $p_m \leq 0$, hence $p_m = 0$. □

Lemma 12 (Lower tightening under exhaustivity)**.** *Assume equality in* (20) *on the modeled outgoing set at* q *(no probability leaks to symbols outside the listed transitions). Fix* $m \in \mathrm{Out}(q)$ *with* $s_j \leq s_j^u$ *for all* $j \in \mathrm{Out}(q) \setminus \{m\}$*. If* $\sum_{j \in \mathrm{Out}(q)\setminus\{m\}} e^{s_j^u} < 1$*, then*

$$s_m \;\geq\; \log\Bigl(1 - \sum_{j \in \mathrm{Out}(q)\setminus\{m\}} e^{s_j^u}\Bigr). \tag{22}$$

When $\sum_{j \in \mathrm{Out}(q)\setminus\{m\}} e^{s_j^u} \geq 1$*, the floor reduces to the trivial bound* $s_m \geq -\infty$ *and no tightening is applied.*

Proof. For $j \neq m$, $p_j = e^{s_j} \leq e^{s_j^u}$, so $\sum_{j \neq m} p_j \leq \sum_{j \neq m} e^{s_j^u}$. Under equality, $p_m = 1 - \sum_{j \neq m} p_j \geq 1 - \sum_{j \neq m} e^{s_j^u}$; if this lower bound on p_m is positive, monotonicity of $\log$ yields (22), and otherwise the inequality is implied by $p_m \geq 0$. □

Iterative propagation. Refinement applies (21) and, when bidirectional tightening is enabled at an exhaustive state q, also (22) within each group $\mathrm{Out}(q)$, using the current interval endpoints and iterating coordinate updates for a bounded number of rounds so that caps propagate across coordinates. When the right-hand side of (21) falls below s_m^ℓ, the axis-aligned box is inconsistent with (20); intersecting with the cap would be unsound, so that tightening step is skipped.

B.2 State simplex in the interval forward DP

Let $\alpha_t(q)$ denote log-masses in the interval forward recurrence (Section 3) and write $\hat{p}_t(q) = e^{\alpha_t(q)}$ when convenient. If at each t the state masses $(\hat{p}_t(q))_{q \in Q}$ satisfy $\sum_q \hat{p}_t(q) \leq 1$ (sub-probability on Q), then coordinatewise upper caps follow from others' lower bounds.

Lemma 13 (State upper cap)**.** *Fix* t *and* $q \in Q$*. Suppose* $\sum_{r \in Q} \hat{p}_t(r) \leq 1$ *and* $\hat{p}_t(r) \geq e^{\alpha_t^\ell(r)}$ *for all* $r \neq q$*. Then* $\hat{p}_t(q) \leq 1 - \sum_{r \neq q} e^{\alpha_t^\ell(r)}$*, and if additionally* $\sum_{r \neq q} e^{\alpha_t^\ell(r)} < 1$*,*

$$\alpha_t(q) \;\leq\; \log\Bigl(1 - \sum_{r \neq q} e^{\alpha_t^\ell(r)}\Bigr). \tag{23}$$

When $\sum_{r \neq q} e^{\alpha_t^\ell(r)} \geq 1$*, the cap forces* $\hat{p}_t(q) = 0$ *(i.e.,* $\alpha_t(q) = -\infty$*); no finite log upper bound follows from the sub-probability constraint alone in this regime.*

Proof. As in Lemma 11, $\sum_{r \neq q} \hat{p}_t(r) \geq \sum_{r \neq q} e^{\alpha_t^\ell(r)}$, hence $\hat{p}_t(q) \leq 1 - \sum_{r \neq q} \hat{p}_t(r) \leq 1 - \sum_{r \neq q} e^{\alpha_t^\ell(r)}$, and the log form follows when the right-hand side is positive. □

When state simplex refinement is used, this cap is applied to the upper interval endpoint of each $\alpha_t(q)$ after the Bellman update at t; no analogous lower tightening is applied at the state layer.

Table 3: NeSyA-MNIST CLG + temporal-operator at $\varepsilon = 0.01$ (N=2000): certified fraction (%) vs. simplex refinement.

Variant	**Cert. (%)**
No simplex	64.98%
+ state simplex	67.59%
+ trans simplex	68.05%
+ state + trans	68.30%
+ state + trans bidir	68.61%

B.3 Soundness as interval refinement

Proposition 14 (Feasible scores stay in the tightened box)**.** *Let $\mathcal{B} = \prod_m [s_m^\ell, s_m^u]$ be any sound axis-aligned enclosure of transition scores over the input perturbation set. Assume the true scores satisfy* (20) *per source state. Let $\mathcal{B}'$ be the result of intersecting $\mathcal{B}$ with the half-spaces implied by Lemma 11 and, for exhaustive states where bidirectional tightening is enabled, Lemma 12. Then every feasible score vector lies in $\mathcal{B}'$. The same principle applies to state-layer caps: if Lemma 13's hypotheses hold for the true masses, tightening upper intervals using its log bound cannot exclude a feasible trajectory.*

Proof. Fix feasible scores (s_m) on the perturbation ball. Soundness of $\mathcal{B}$ gives $s_m \in [s_m^\ell, s_m^u]$ for all m. For each q and $m \in \mathrm{Out}(q)$, true masses satisfy (20) and $s_j \geq s_j^\ell$ for $j \in \mathrm{Out}(q) \setminus \{m\}$, so Lemma 11 yields the cap (21); intersecting with $s_m \leq s_m^u$ keeps every feasible s_m. Where exhaustivity holds and bidirectional tightening is on, $s_j \leq s_j^u$ for $j \neq m$ and Lemma 12 gives the floor (22); the intersection again cannot exclude a feasible value. Repeating these intersections with updated endpoints only adds inequalities valid for all feasible vectors (steps where (21) is inapplicable are skipped so the feasible set is not enlarged).

For the state layer, if $\sum_r \hat{p}_t(r) \leq 1$ and lower bounds on $\hat{p}_t(r)$ for $r \neq q$ are sound, Lemma 13 implies any feasible $\hat{p}_t(q)$ obeys the same cap, hence feasible $\alpha_t(q) = \log \hat{p}_t(q)$ lies below the tightened log upper endpoint. □

B.4 Fused temporal operator and linear-bound composition

For the fused map $s \mapsto F(s) = \log \mathbb{P}_{\mathrm{acc}}$, certification combines (i) an interval upper value b_{up} from the interval forward recurrence evaluated on $[s^\ell, s^u]$ (after transition and state tightening when used), and (ii) a supporting affine minorant $s \mapsto w^\top s + b_{\mathrm{lo}}$ at the lower orthotope corner when F is convex and coordinatewise nondecreasing (Section 3.3).

Remark 15 (Aligned orthotope and anchor)*.* If transition simplex refinement is applied, the orthotope $[s^\ell, s^u]$ for transition scores should be replaced by its refined intersection before evaluating the interval image of F and before evaluating the subgradient at the lower corner s^ℓ. Affine relaxations on those score variables must be taken with respect to the same refined box so that the monotone upper image and the tangent minorant refer to a single feasible orthotope.

Remark 16 (Upper certificates and the lower relaxation)*.* Under backward linear relaxation, the bias contributing to an upper bound at a parent node can depend on both the upper and lower relaxations chosen at a child, depending on the sign pattern of backward coefficients. Thus bidirectional transition tightening (which raises s^ℓ and changes w and b_{lo}) can move the reported scalar upper bound on F without contradicting soundness: Proposition 14 only refines the feasible set, and the composed bounds remain valid if b_{up} upper-bounds F on the refined box and the affine minorant remains valid globally under the convexity assumption. When comparing methods, stratify by label: for y=0 compare certified upper bounds on $\log \mathbb{P}_{\mathrm{acc}}$; for y=1 compare lower bounds; pooled means over both labels mix quantities governed by different inequalities.

B.5 Empirical effect on NeSyA-MNIST

Table 3 isolates the incremental effect of interval simplex refinements on top of the CLG temporal-operator stack at $\varepsilon = 0.01$, using the same default MNIST slice (N=2000) as Table 1 (which

keeps simplex disabled so the four structural encodings are compared on equal footing). Each row corresponds to a combination of optional state simplex caps, transition-facet tightening, and bidirectional transition tightening, as labeled in the first column; every cell is the overall certified fraction in %. Gains over "no simplex" are modest at this radius; the full combination (state and bidirectional transition simplex) attains the highest certified fraction, and transition simplex alone gives the largest single increment.

C System specifications

This appendix records the precise NeSyA systems used in Sections 4.1, 4.2, and D: the perception network, per-frame symbol partition, automaton, and verification spec for each.

C.1 NeSyA-MNIST

Perception network and symbols. Each timestep presents a single grayscale MNIST digit d of shape $1 \times 28 \times 28$. The perception network is a small CNN with three $\text{Conv}_{3\times3} \to \text{ReLU} \to \text{AvgPool}_{2\times2}$ blocks (channels $1 \to 8 \to 16 \to 32$) followed by a $32 \to 5$ linear head, producing 5 logits per timestep partitioned into two categorical groups:

group	atoms (predicates)
`magnitude`	`smaller_than_3`, `between_3_and_6`, `larger_than_6`
`parity`	`even`, `odd`

Independence across groups (Section 2) gives a joint-assignment cardinality of $K = 3 \cdot 2 = 6$ per timestep.

Automaton. The automaton has four states $\{0, 1, 2, 3\}$ with initial state 0 and accepting state 3. The transition table is:

$q \to q'$	guard
$0 \to 0$	`¬(even & larger_than_6)`
$0 \to 1$	`even & larger_than_6`
$1 \to 1$	`¬(¬even & ¬larger_than_6)`
$1 \to 2$	`¬even & ¬larger_than_6`
$2 \to 2$	`¬smaller_than_3`
$2 \to 3$	`smaller_than_3`
$3 \to 3$	`true` (absorbing accepting state)

The seven transitions partition the joint-assignment alphabet exhaustively at every non-accepting state, so the exhaustivity hypothesis of Lemma 12 applies throughout. Logic constraints are imposed on the `magnitude` group (mutually exclusive atoms). The verification horizon is $T \in \{3, \ldots, 10\}$ for the main robustness sweep (Section 4.1) and $T \in \{4, 6, \ldots, 20\}$ for the T-scaling experiment (Section 4.1).

C.2 MNIST-extended (dim-3 triplet symbols)

Perception network and symbols. Each timestep presents a triplet of MNIST digits (d_1, d_2, d_3) as a three-channel grayscale image of shape $3 \times H \times W$. The perception network is the same convolutional backbone as NeSyA-MNIST applied independently to each channel; its outputs are concatenated to a 15-dimensional logit vector per timestep, partitioned into six categorical groups:

group	atoms (predicates)
magnitude_d1	smaller_than_3_d1, between_3_and_6_d1, larger_than_6_d1
parity_d1	even_d1, odd_d1
magnitude_d2	smaller_than_3_d2, between_3_and_6_d2, larger_than_6_d2
parity_d2	even_d2, odd_d2
magnitude_d3	smaller_than_3_d3, between_3_and_6_d3, larger_than_6_d3
parity_d3	even_d3, odd_d3

Independence across groups (Section 2) gives a joint-assignment cardinality of $K = 3^3 \cdot 2^3 = 216$ per timestep, a $36\times$ increase over the NeSyA-MNIST setting.

Automaton. The automaton has ten states $\{0, 1, \ldots, 9\}$ with initial state 0 and accepting state 9. Each non-accepting state has two outgoing transitions: a self-loop guarded by the negation of an advancement guard, and an advancement transition guarded by that formula. The advancement guards are NNF Boolean formulas over atoms drawn from multiple streams, so verification cannot be decoupled into per-stream subproblems. The full transition table is:

$q \to q'$	advancement guard
$0 \to 1$	(odd_d1 & even_d2) | (even_d1 & odd_d3)
$1 \to 2$	(even_d1 & odd_d2) | (odd_d1 & odd_d3)
$2 \to 3$	even_d1 & odd_d2 & larger_than_6_d3
$3 \to 4$	odd_d1 & even_d2 & even_d3
$4 \to 5$	between_3_and_6_d1 & smaller_than_3_d2 & larger_than_6_d3
$5 \to 6$	(even_d1 & odd_d3) | (odd_d1 & odd_d2)
$6 \to 7$	even_d3
$7 \to 8$	even_d1 & smaller_than_3_d2 & larger_than_6_d3
$8 \to 9$	odd_d1 & odd_d2 & odd_d3
$9 \to 9$	true (absorbing accepting state)

At each non-accepting state $q \in \{0, \ldots, 8\}$, the self-loop carries the negation of the row's advancement guard, so outgoing transitions partition the joint-assignment alphabet exhaustively (the exhaustivity hypothesis of Lemma 12 applies at every non-accepting state). The verification horizon is fixed at $T = 30$.

C.3 CAVIAR

Perception network and symbols. Each timestep supplies two 80×80 RGB person crops $(\mathbf{c}_{p_1}, \mathbf{c}_{p_2})$, stacked into a $3\times80\times160$ input. The perception network applies a four-block convolutional encoder ($3 \to 32 \to 64 \to 64 \to 64$ channels with 5×5 kernels, ReLU and 2×2 MaxPool after each block) independently to each crop, yielding a 64-dimensional embedding per person; two linear heads project each embedding to four simple_event logits. Architecture and weights follow the reference NeSyA implementation [1]. The auxiliary predicate close_p1_p2 (Euclidean distance below threshold) is supplied as ground truth from the dataset annotations and routed as a non-perturbed input to the verifier. Per-frame symbol groups are:

group	atoms (predicates)
p1	walking, running, active, inactive
p2	walking, running, active, inactive
close_p1_p2	close, not_close

Independence across groups (Section 2) gives a joint-assignment cardinality of $K = 4 \cdot 4 \cdot 2 = 32$ per timestep.

Automaton. The automaton (taken verbatim from the reference NeSyA implementation [1]) has three complex-event states $\{0 : \texttt{no_event},\ 1 : \texttt{moving},\ 2 : \texttt{interacting}\}$ with initial state 0 and accepting state 1. Each of states 1 and 2 has an exit guard to state 0 and a complementary self-loop, while state 0 has guards advancing to either complex event with the residual self-loop covering all

Table 4: MNIST-extended (digit triplets per frame; $T{=}30$, ten-state automaton): certified fraction (%) by variant and ε, with amortized wall time (*s/seq*); batch size 4. Per-label split in Appendix F. Best per-column percentage values and lowest *s/seq* bold.

Variant	$\varepsilon = 0.0001$		$\varepsilon = 0.0002$		$\varepsilon = 0.0005$		$\varepsilon = 0.001$	
	cert. %	*s/seq*	cert. %	*s/seq*	cert. %	*s/seq*	cert. %	*s/seq*
Baseline (recursive)	**99.0**	*60.8*	96.9	*61.9*	52.1	*61.1*	52.1	*62.6*
CLG (recursive)	**99.0**	*51.2*	**99.0**	*51.3*	**96.9**	*51.5*	89.6	*54.4*
CLG temporal-op	**99.0**	***6.8***	**99.0**	***6.8***	**96.9**	***7.0***	**90.6**	***8.5***

other inputs; full guard formulas (conjunctions over the simple-event and `close_p1_p2` atoms) are provided in our released configuration.

Verification spec and downsampling. Raw clips are sampled at 1 fps (stride 25), giving downsampled lengths $T_{\mathrm{ds}} \in [4, 43]$ across the 17 clips; the automaton is re-run on the kept-frame ground-truth atoms to obtain a downsampled state trajectory. For each clip i we let $c_i^\star$ be the state at the final downsampled frame. After filtering to clips on which the model agrees with the target on clean inputs (13 of 17 clips qualify), we certify

$$\forall \mathbf{x}' \colon \|\mathbf{x}' - \mathbf{x}\|_\infty \le \varepsilon \;\Rightarrow\; \mathbb{P}_{\mathrm{model}}\big(\mathrm{state}[T_{\mathrm{ds}}] = c_i^\star \,\big|\, \mathbf{x}'\big) > \tfrac{1}{2}.$$

D MNIST-extended: scaling the symbolic system

Why this benchmark. The dominant scaling axis in our problem is not the perception architecture (neural verifiers have repeatedly demonstrated scalability across CNN/MLP/ResNet families on the same threat model [3, 4, 23]) but the symbolic system on top of it: the cardinality of the joint-assignment alphabet per time step, the number of automaton transitions and states, the structure of the transition guards, and the verification horizon. NeSyA-MNIST exercises a small symbolic system (two categorical groups per frame, a four-state automaton, $T \in \{3, \ldots, 10\}$). The MNIST-extended benchmark, in contrast, scales every one of these axes simultaneously: three independent symbolic dimensions per frame, six categorical groups, a richer transition language with cross-stream guards, ten rather than four automaton states, and a fixed horizon of $T{=}30$ (Appendix C.2 for the full specification). This is the regime in which interval explosion is most acute and structural refinements most consequential, and it is the empirical stress test for our claims.

System. The benchmark scales the per-frame joint-assignment alphabet to $K{=}3^3 \cdot 2^3{=}216$ (a $36\times$ increase over NeSyA-MNIST) and the automaton to ten states with eighteen cross-stream guarded transitions, with horizon fixed at $T{=}30$ to expose the full layerwise relaxation depth (Section 3.3, Remark 9); full perception network, symbols, and automaton in Appendix C.2. We sample 100 class-balanced test sequences (50 per label) and verify only those whose clean prediction matches the ground-truth label (50 of 50 for $y{=}0$ and 46 of 50 for $y{=}1$, totaling 96); at this granularity, binomial standard errors on each cell are at most $\sim$5 pp. Compared variants are Baseline (recursive), CLG (recursive), and CLG temporal-op; path unrolling is omitted because at $T{=}30$ on this ten-state automaton the accepting-path count grows to $\sim 10^6$, exceeding the 16 GB GPU's LSE-tensor budget.

Findings. Table 4 confirms that the qualitative ordering from NeSyA-MNIST carries to this larger symbolic system, with the same failure mode emerging at radii $\sim 10\times$ tighter (the $36\times$ alphabet expansion shrinks the per-step margin commensurately): at $\varepsilon \ge 0.0005$ the baseline collapses to 52.1% overall while every CLG variant retains 89–99%, the gap concentrated on $y{=}1$ sequences (Appendix F, Table 10: baseline 0% on $y{=}1$ vs CLG 78–93% at the same radii). The fused temporal operator preserves CLG's certified rates while reducing wall-clock time by $\sim 6\times$ relative to recursive CLG ($54.4 \to 8.5$ s/seq at $\varepsilon{=}0.001$; comparable savings at smaller radii), the dominant cost saving over recursive temporal aggregation when both T and K are large. That the baseline costs $\sim$60 s/seq and still drops to 52.1% at moderate ε illustrates that the natural temporal extension of prior NeSy verification, even strengthened with CROWN-style linear bounds, does not by itself scale to symbolic systems of this size.

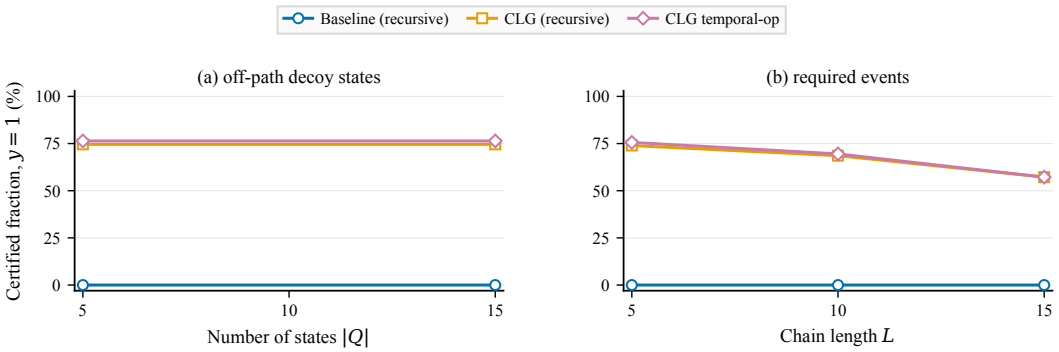

Figure 5. Automaton-size ablation on NeSyA-MNIST (K=6, ε=0.005, T=20): certified fraction on y=1 sequences. (a) Adding off-path decoy states leaves it unchanged. (b) Adding required events to the accepting path lowers it.

E Isolated scaling ablations

MNIST-extended (Appendix D) grows the alphabet, the automaton, and the horizon together. Here we vary one axis at a time on NeSyA-MNIST, reusing the trained perception network. Compared variants are Baseline (recursive), CLG (recursive), and CLG temporal-op.

Table 5: Automaton-size ablation on NeSyA-MNIST (K=6, ε=0.005, T=20): certified fraction y=0 / y=1 (%) by variant, with amortized wall time (*s/seq*). Off-path decoy states grow $|Q|$ with the accepting task fixed; a chain of length L adds $L-1$ required events to the accepting path.

Variant	Decoy states		Chain length		
	$\|Q\|$=5	$\|Q\|$=15	L=5	L=10	L=15
Baseline (recursive)	35.4/0.0 (*0.8*)	35.4/0.0 (*1.6*)	16.5/0.0 (*0.8*)	89.0/0.0 (*1.3*)	100.0/0.0 (*1.9*)
CLG (recursive)	76.2/74.6 (*0.7*)	76.2/74.6 (*1.3*)	73.4/74.0 (*0.7*)	98.4/68.6 (*1.1*)	100.0/57.2 (*1.6*)
CLG temporal-op	95.5/76.4 (*0.5*)	95.5/76.4 (*0.6*)	93.2/75.6 (*0.5*)	99.6/69.5 (*0.6*)	100.0/57.2 (*0.6*)

Automaton size. We fix K=6, ε=0.005, and T=20 and grow the automaton in two ways (Table 5, Figure 5). First, we keep the accepting task fixed (three advancement guards, each satisfied by two of ten digits) and add off-path decoy states, from $|Q|$=5 to $|Q|$=15. No sequence changes certification outcome for any variant, and per-sequence bounds change by at most 1.3×10^{-5}, while wall time roughly doubles for the recursive variants (0.8 vs 1.6 s/seq for the baseline). Second, we grow a pure chain of length $L \in \{5, 10, 15\}$, so every state lies on the accepting path and adds a required event. The certified fraction of recursive CLG on y=1 sequences then falls from 74.0% to 57.2%. The clean margin barely moves (median clean $\log \mathbb{P}_{\text{acc}}$ changes by 0.03), so the decline comes from per-step slack accumulating over more guarded transitions (median end-to-end width from 0.33 to 1.05). The number of states is therefore free on the certifying axis, and the number of required events on the accepting path is what binds.

Alphabet size and guard fan-in. In MNIST-extended, growing the alphabet also grows guard fan-in, the number of perception channels a guard reads. We separate the two with $|Q|$=4 and T=10 (Table 6, Figure 6). With fan-in fixed at one, adding channels that no guard reads grows K from 6 to 36 to 216 and moves ε_{50}, the radius at which half of y=1 sequences certify, from 0.0086 to 0.0048 to 0.0034, since the joint-K log-sum-exp still carries the unread channels. At fixed K=216, reading two or three hard channels (each condition satisfied by two of ten digits) lowers the certified fraction on y=1 sequences at ε=0.002 from 80.7% to 76.4% to 67.6%, so the cost of fan-in grows faster than linearly. Reading three easy channels (six of ten digits each) costs little (78.6%), so the fan-in cost comes from hard channels. Wall time grows by roughly 7× from K=6 to K=216 for the recursive variants, but does not change with fan-in.

Slack decomposition. Table 7 splits the end-to-end width on $\log \mathbb{P}_{\text{acc}}$ at ε=0.005 into per-step widths at the transition scores and a temporal gap, using the SAT-accepting path of Figure 9. The

Table 6: Alphabet and fan-in ablation ($|Q|$=4, T=10), CLG (recursive) on y=1 sequences: ε_{50} is the radius at which half of the sequences certify (log-interpolated), cert. % is the certified fraction at ε=0.002, and width is the median bound width at ε=0.001. Full results in Table 8.

Arm	ε_{50}	cert. % at ε=0.002	width at ε=0.001
K=6, fan-in 1	0.0086	98.1	0.0003
K=36, fan-in 1	0.0048	87.1	0.0011
K=216, fan-in 1	0.0034	80.7	0.0037
K=216, fan-in 2 (hard)	0.0032	76.4	0.0038
K=216, fan-in 3 (hard)	0.0026	67.6	0.0126
K=216, fan-in 3 (easy)	0.0032	78.6	0.0047

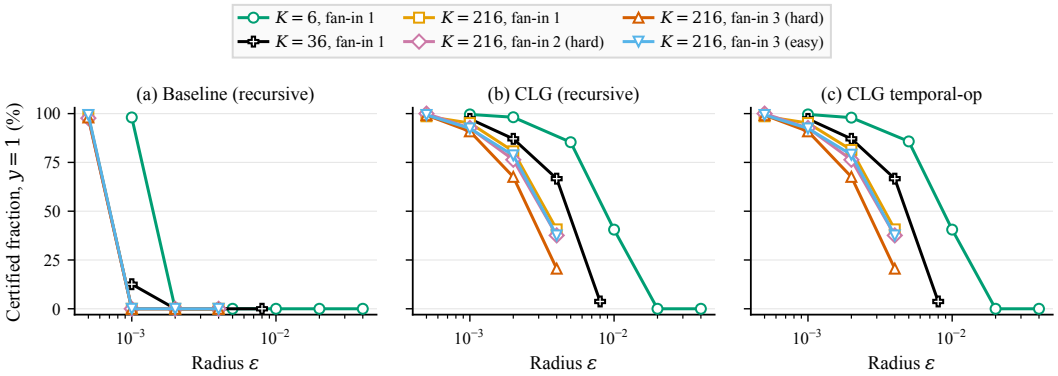

Figure 6. Alphabet and fan-in ablation ($|Q|$=4, T=10): certified fraction on y=1 sequences versus ε for each arm and variant.

Table 7: Slack decomposition on NeSyA-MNIST at ε=0.005 (log-probability units). Transition-score widths are taken over all (frame, transition) pairs; the temporal gap is the actual end-to-end bound minus the bound predicted by summing per-step bounds along the SAT-accepting path.

Quantity	**Value**
CNN logit width (median)	2.26
Transition-score width, baseline (median)	4.87
Transition-score width, CLG (median)	0.0002
Transitions where CLG is strictly tighter	57%
Temporal gap, y=1 lower bound (median)	0.000
Temporal gap, y=0 lower bound (median)	0.000
Temporal gap, y=0 upper bound (mean)	2.42
y=1 sequences with no regime-B transition on the SAT-accepting path	99.1%

Table 8: Alphabet and fan-in ablation, full results ($|Q|$=4, T=10): certified fraction y=0 / y=1 (%) by variant, arm, and ε, with amortized wall time (*s/seq*).

Arm	ε	Baseline (recursive)	CLG (recursive)	CLG temporal-op
K=6, fan-in 1	0.001	99.1/98.0 (*0.2*)	99.1/99.6 (*0.2*)	99.8/99.6 (*0.1*)
	0.002	97.5/0.0 (*0.2*)	98.2/98.1 (*0.2*)	99.3/97.9 (*0.1*)
	0.005	72.5/0.0 (*0.2*)	93.6/85.3 (*0.2*)	97.6/85.8 (*0.2*)
	0.01	12.3/0.0 (*0.2*)	42.8/40.6 (*0.2*)	61.9/40.6 (*0.2*)
	0.02	0.0/0.0 (*0.2*)	0.0/0.0 (*0.2*)	0.0/0.0 (*0.3*)
	0.04	0.0/0.0 (*0.4*)	0.0/0.0 (*0.2*)	0.0/0.0 (*0.3*)
K=36, fan-in 1	0.001	98.7/12.6 (*0.7*)	98.3/97.3 (*0.5*)	100.0/97.3 (*0.4*)
	0.002	93.6/0.0 (*0.7*)	94.6/87.1 (*0.6*)	97.3/87.1 (*0.5*)
	0.004	77.8/0.0 (*0.8*)	85.5/66.7 (*0.7*)	92.6/66.7 (*0.6*)
	0.008	16.5/0.0 (*1.0*)	0.0/3.7 (*0.9*)	2.0/3.7 (*0.8*)
K=216, fan-in 1	0.0005	100.0/98.6 (*1.0*)	100.0/98.6 (*0.8*)	100.0/98.6 (*0.7*)
	0.001	99.0/0.0 (*1.4*)	97.3/95.2 (*1.2*)	100.0/95.2 (*1.0*)
	0.002	93.9/0.0 (*1.6*)	92.9/80.7 (*1.4*)	97.3/81.4 (*1.1*)
	0.004	71.3/0.0 (*1.8*)	73.0/40.7 (*1.6*)	84.1/40.7 (*1.3*)
K=216, fan-in 2 (hard)	0.0005	100.0/97.7 (*0.9*)	100.0/100.0 (*0.8*)	100.0/100.0 (*0.7*)
	0.001	99.7/0.0 (*1.3*)	99.7/92.6 (*1.2*)	99.7/92.6 (*1.0*)
	0.002	99.7/0.0 (*1.5*)	99.7/76.4 (*1.4*)	99.7/76.4 (*1.1*)
	0.004	98.7/0.0 (*1.7*)	97.0/37.6 (*1.6*)	98.3/37.6 (*1.3*)
K=216, fan-in 3 (hard)	0.0005	100.0/98.0 (*0.9*)	100.0/99.2 (*0.8*)	100.0/99.2 (*0.7*)
	0.001	100.0/0.0 (*1.3*)	100.0/90.9 (*1.2*)	100.0/90.9 (*1.0*)
	0.002	100.0/0.0 (*1.4*)	100.0/67.6 (*1.4*)	100.0/67.6 (*1.1*)
	0.004	100.0/0.0 (*1.6*)	100.0/20.6 (*1.6*)	100.0/20.6 (*1.3*)
K=216, fan-in 3 (easy)	0.0005	99.7/99.3 (*1.0*)	99.3/99.3 (*0.8*)	100.0/99.3 (*0.7*)
	0.001	94.9/0.0 (*1.4*)	94.9/92.4 (*1.2*)	99.0/92.4 (*1.0*)
	0.002	91.2/0.0 (*1.5*)	90.1/78.6 (*1.4*)	94.6/79.0 (*1.1*)
	0.004	61.9/0.0 (*1.7*)	57.5/37.6 (*1.6*)	73.5/37.6 (*1.3*)

temporal gap is zero at the median on both lower bounds and is large only on the y=0 upper bound (2.42 units on average), from secondary accepting paths. The CNN's relaxation is contained in the per-step term. The median CROWN width of the CNN logits is 2.26, while the median transition-score width is 4.87 for the baseline and 0.0002 for CLG, whose outer-softplus multiplier damps the incoming slack. We report medians because CLG widths are bimodal, near zero in regime A and loose in regime B, so the pooled mean CLG width (6.02) exceeds the baseline's (4.89). Certification is decided on the SAT-accepting path, where 99.1% of y=1 sequences have no regime-B transition.

Summary. Looseness is per-step. It does not grow with the number of states, and it grows with the number of required events on the accepting path and with the number of hard channels each guard reads. Runtime grows with the alphabet size and the horizon, mildly with the number of states, and not with guard fan-in. Once the fused temporal operator removes the cost of relaxing the recurrence (54.4 vs 8.5 s/seq on MNIST-extended, Table 4), the remaining cost is dominated by per-frame CNN bound propagation.

F Per-label certified fractions and regime routing

The certified fractions reported in Tables 1 and 4 pool both labels. Tables 9 and 10 stratify these by ground-truth label and reveal that the bulk of the baseline-vs-CLG gap concentrates on y=1. This appendix explains the asymmetry in two pieces: (i) a threshold-headroom argument that applies to any verifier under cross-entropy training, and (ii) an empirical regime-routing analysis specific to the recurrence used by both heads.

Table 9: NeSyA-MNIST: certified fraction (%) per label by variant and ε. Per-label totals after the clean-prediction filter: 991 (y=0) and 965 (y=1). Best per-column values bold.

Variant	$\varepsilon = 0.001$		$\varepsilon = 0.002$		$\varepsilon = 0.005$		$\varepsilon = 0.01$	
	y=0	y=1	y=0	y=1	y=0	y=1	y=0	y=1
Baseline (recursive)	**99.8**	98.7	98.4	14.9	83.9	0.0	52.8	0.0
CLG (recursive)	**99.8**	**99.2**	99.0	**98.3**	96.0	88.9	70.1	**49.3**
CLG path-unroll	**99.8**	**99.2**	**99.7**	**98.3**	98.2	89.3	79.7	**49.3**
CLG temporal-op	99.7	**99.2**	**99.7**	**98.3**	**98.3**	**89.4**	**80.2**	**49.3**

Table 10: MNIST-extended: certified fraction (%) per label by variant and ε. Per-label totals after the clean-prediction filter: 50 (y=0) and 46 (y=1); y=0 saturates at 100% for every variant and radius. Best per-column y=1 values bold.

Variant	$\varepsilon = 0.0001$		$\varepsilon = 0.0002$		$\varepsilon = 0.0005$		$\varepsilon = 0.001$	
	y=0	y=1	y=0	y=1	y=0	y=1	y=0	y=1
Baseline (recursive)	100.0	**97.8**	100.0	93.5	100.0	0.0	100.0	0.0
CLG (recursive)	100.0	**97.8**	100.0	**97.8**	100.0	**93.5**	100.0	78.3
CLG temporal-op	100.0	**97.8**	100.0	**97.8**	100.0	**93.5**	100.0	**80.4**

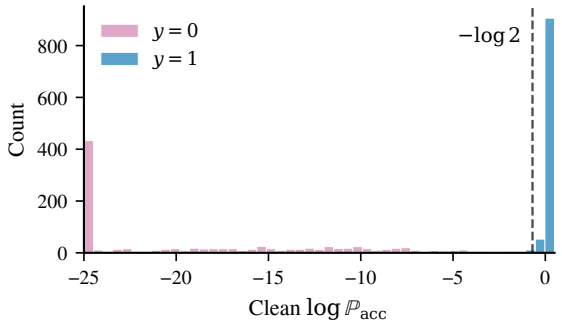

Figure 7. Headroom on NeSyA-MNIST: clean $\log \mathbb{P}_{\text{acc}}$ per test sequence, stratified by ground-truth label. Dashed line marks the certification threshold $-\log 2 \approx -0.69$. Sequences with y=1 concentrate just above the threshold; sequences with y=0 sit several units below it.

Threshold-headroom. Verification of the two labels is asymmetric by construction: a positive-label (y=1) sequence requires the lower bound on $\log \mathbb{P}_{\text{acc}}$ to clear $-\log 2 \approx -0.69$, while a negative-label (y=0) sequence requires the upper bound to stay below the same threshold. Cross-entropy training pushes clean $\mathbb{P}_{\text{acc}}$ to extremes: accepting trajectories sit near $\mathbb{P}_{\text{acc}} \to 1$ (clean $\log \mathbb{P}_{\text{acc}} \approx 0$, so headroom against the threshold is $\sim$0.7 units) while non-accepting trajectories sit near $\mathbb{P}_{\text{acc}} \to 0$ (clean $\log \mathbb{P}_{\text{acc}} \ll 0$, several units of headroom). Layerwise tangent slack on LSE accumulating through the recursive encoding then consumes the small y=1 budget long before the large y=0 one, so any verifier that retains non-trivial per-step slack collapses on y=1 first. Figure 7 validates this empirically on the test set: 96.5% of y=1 sequences sit at clean $\log \mathbb{P}_{\text{acc}} \in [-0.33, 0]$ (median $\approx$0, headroom $\approx$0.7), whereas 99.1% of y=0 sequences sit below $-\log 2$ with median $\log \mathbb{P}_{\text{acc}} = -21.0$ (20+ units of headroom).

Regime routing on the SAT-accepting path. The threshold-headroom argument explains why y=1 is the harder side for *any* method but does not, on its own, explain why CLG closes most of that gap when its per-step regime structure is nominally label-agnostic. To probe this we compute, for every test sequence, the clean $p_{\mathcal{S}_m}^{(t)}$ at each (frame, transition) pair, classify by regime (A: $p_{\mathcal{S}_m} \geq 0.99$;

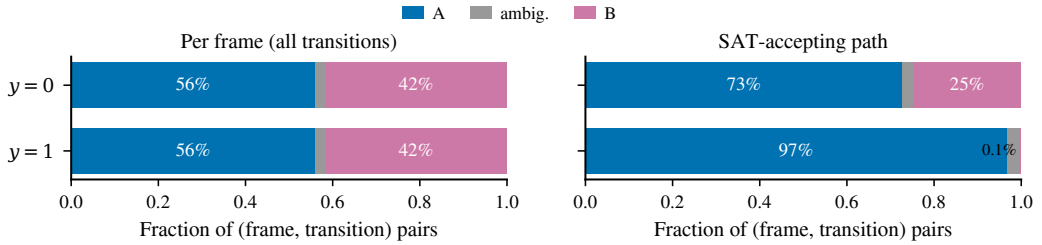

Figure 8. Regime distribution on NeSyA-MNIST, stratified by sequence label. *Left:* every (frame, transition) pair across the test set. *Right:* only transitions on the SAT-accepting path (Viterbi over accepting trajectories). Regime A: $p_{\mathcal{S}_m} \geq 0.99$; ambiguous: $0.01 < p_{\mathcal{S}_m} < 0.99$; regime B: $p_{\mathcal{S}_m} \leq 0.01$.

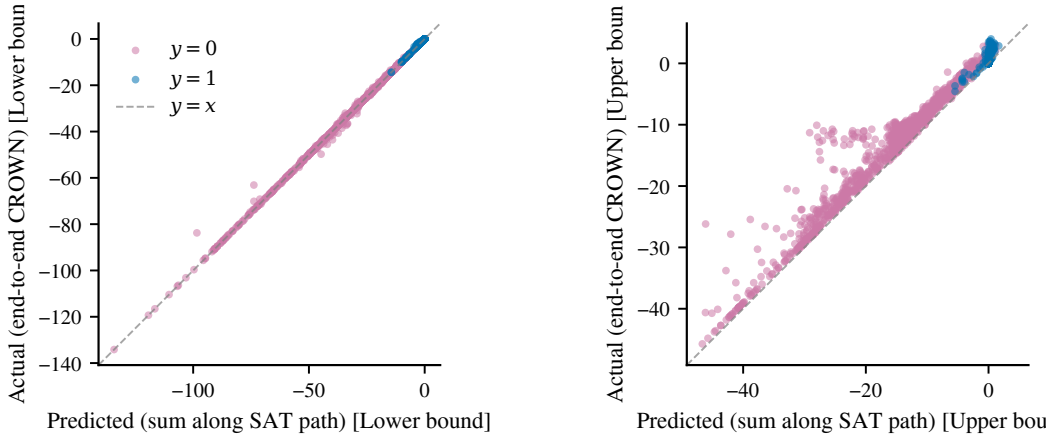

Figure 9. SAT-path-dominance check. Each point is one test sequence at ε=0.005 on NeSyA-MNIST: x-axis is the predicted bound obtained by summing per-step CROWN bounds along the SAT-accepting path; y-axis is the actual end-to-end CROWN bound on $\log \mathbb{P}_{\text{acc}}$. Tight clustering on the y=x diagonal indicates that the LSE-DP recurrence is operationally dominated by the SAT-accepting path's transitions.

ambiguous: $0.01 < p_{\mathcal{S}_m} < 0.99$; B: $p_{\mathcal{S}_m} \leq 0.01$), and aggregate by sequence label both across all (frame, transition) pairs and along the SAT-accepting path: the highest-clean-probability accepting trajectory, identified via Viterbi over accepting trajectories.

Figure 8 (left) shows that the per-frame distribution is essentially identical between labels: 55.9% regime A and 41.6% regime B in both, indistinguishable to within 0.1pp. The recurrence, however, does *not* reduce $\log \mathbb{P}_{\text{acc}}$ to a representative per-frame transition; the lower bound is dominated by the SAT-accepting path, which can differ from the model's fired path at frames where the argmax symbol does not satisfy the accepting path's guard. Figure 8 (right) shows the resulting routing divergence: the SAT-accepting path is 96.7% regime A on y=1 sequences (with 0.14% regime B) but only 72.8% regime A on y=0 sequences, with 24.8% in regime B. Per sequence, 99.1% of y=1 sequences carry zero regime-B transitions on the SAT-accepting path, while 96% of y=0 sequences carry at least one (typically one, two, or three).

The intuition is direct: for a y=1 sequence the SAT-accepting path is the symbolic ground-truth path, whose per-frame guards match the model's argmax (because the sequence passed the clean-prediction filter); these guards therefore land in regime A. For a y=0 sequence the SAT-accepting path is the highest-probability *accepting* trajectory under a model that does not in fact accept the sequence, so at least one frame must cross to a guard that disagrees with the model's argmax, and that crossing is precisely what places the transition in regime B. CLG's per-step bounds are essentially exact in regime A and mildly looser than baseline's in regime B; the recursive encoding therefore routes y=1 verification through transitions where CLG dominates and y=0 verification through a mix that

includes regime-B transitions where CLG's relative advantage shrinks. Composed with the headroom asymmetry above, this is why CLG's improvement over baseline is large on $y{=}1$ (small budget, near-zero CLG slack on regime-A transitions preserves it) and smaller, though still substantial, on $y{=}0$ (large budget, mixed-regime slack but still tighter than baseline's structural cancellation losses).

Empirical SAT-path-dominance. The argument above relies on a heuristic: although the recurrence is LSE-DP and considers all accepting paths simultaneously, LSE is dominated by its max term, so at trained-model confidence levels the SAT-accepting path's per-step transitions effectively drive both the lower and upper bounds on $\log \mathbb{P}_{\mathrm{acc}}$. We verify this directly: for each test sequence we (a) run end-to-end CROWN on the full log-semiring map to obtain the actual bound on $\log \mathbb{P}_{\mathrm{acc}}$, (b) run per-frame CROWN on each transition score $s_{t,m}$ and sum the per-step bounds along the Viterbi-identified SAT-accepting path to obtain a predicted bound, and (c) compare. Figure 9 reports the result over all 2000 test sequences at $\varepsilon{=}0.005$: predicted and actual lower bounds agree to within numerical noise across both labels (Pearson $R^2 = 0.999$ on $y{=}0$ and $R^2 = 0.994$ on $y{=}1$, median actual$-$predicted gap 0.0); the upper bound shows similarly tight agreement on $y{=}0$ ($R^2 = 0.95$, mean actual UB ~ 2.4 units above predicted, reflecting the residual LSE-versus-max gap from secondary accepting paths). Together with the regime distribution of Figure 8, this licenses the operational reading of the recurrence as Viterbi over the SAT-accepting path in this regime.

G Reproducibility: implementation details

In this section, we provide the implementation details necessary to fully reproduce the empirical evaluations presented in the main text.

Software stack. All experiments use Python 3.12, PyTorch 2.x for forward computation and auto-differentiation, and the auto_LiRPA library [3, 4] for CROWN-style linear bound propagation through both the perception network and the log-semiring automaton head. We register a small set of custom auto_LiRPA `Bound` operators specific to our pipeline: a CROWN backward rule for `torch.logsumexp` (tangent-lower / chord-upper), a custom `DCGap` operator implementing the joint contrastive-gap relaxation $\mathrm{LSE}(\tilde{z}; \mathcal{S}^{\mathrm{c}}) - \mathrm{LSE}(\tilde{z}; \mathcal{S})$ that preserves coordinate correlation across the two restricted LSE heads, and a `TemporalDP` operator that wraps the fused temporal aggregator with corner-tight forward/backward bound propagation. Source files: `src/log_semiring_automaton.py`, `src/verification/engine.py`.

Hardware. Wall-clock times reported throughout the paper were measured on a single Linux workstation with an Intel Core i9-10940X CPU (14 physical cores, 28 threads), 256 GiB of system memory, and a single NVIDIA RTX 4060 Ti GPU with 16 GB of VRAM. All verification runs use a single GPU; no distributed or multi-GPU execution is used.

CROWN configuration. End-to-end verification uses the CROWN method (no α-CROWN optimization) with auto_LiRPA's bound options `conv_mode="conv"`, `sparse_intermediate_bounds=True` and `sparse_conv_intermediate_bounds=True` (with `minimum_sparsity=0.9`) to keep memory bounded by exploiting unstable-neuron sparsity through ReLU layers, `crown_batch_size=256` for the spec-dimension chunking in CROWN-backward (lower values trade wall time for peak GPU memory; identical bounds either way), and a per-node truncation of `max_crown_size=2000` during CROWN-backward over highly unstable intermediate ReLU/conv layers. β-CROWN is disabled (`enable_beta_crown=False`) and we use 0 optimization iterations: all bounds reported are pure CROWN with no branch-and-bound or input-region splitting. The same bound options are used uniformly for all four structural variants in the main tables and for the auxiliary per-step bound-width and SAT-path-dominance studies.

Perception network architectures and training. The NeSyA-MNIST CNN (`NormalizedSimpleEventCNN`, defined in `examples/MNIST/model_definition.py`) consists of three Conv2d $\to$ ReLU $\to$ AvgPool blocks with channel widths $1 \to 8 \to 16 \to 32$, 3×3 kernels, and 2×2 pooling, followed by a $32 \to 5$ linear head producing five logits per frame partitioned into the magnitude (3-way) and parity (2-way) categorical groups. Inputs are normalized by the standard MNIST mean/std ($0.1307/0.3081$). The CNN is trained per-frame (no temporal supervision) on the MNIST training split with an $80/20$ train/validation random split (seeded),

batch size 32, the Adam optimizer at learning rate 10^{-3}, and cross-entropy loss summed over the magnitude and parity heads, for 50 epochs (`examples/MNIST/cnn_training.py`). The MNIST-extended benchmark uses the same backbone applied independently to each of three input channels (one per digit position), with the per-channel logit outputs concatenated to a 15-dimensional vector partitioned into six categorical groups; weights are reused from the standard MNIST training without re-training. The CAVIAR perception network (`CaviarSymbolGrounder`, defined in `examples/CAVIAR/model_definition.py`) is the architecture used in the reference NeSyA implementation [1]: a four-block convolutional encoder ($3 \to 32 \to 64 \to 64 \to 64$ channels with 5×5 kernels and 2×2 MaxPool after each block), applied independently to each 80×80 person crop to obtain a 64-dimensional per-person embedding, projected by two $64 \to 4$ linear heads to the four `simple_event` logits. We retrain the CAVIAR grounder following the protocol of [1] verbatim (per-frame supervision via the deterministic three-state automaton on ground-truth atoms; one sequence per Adam step; learning rate 10^{-3}; early stopping on training loss with patience 10; up to 80 epochs).

Data construction and splits. For NeSyA-MNIST we construct a class-balanced test set of 2000 sequences (1000 per label) from the MNIST test split. Each sequence is a sequence of single-digit images of length T drawn uniformly from $\{3, \ldots, 10\}$, with random seed 123; sequence ground-truth labels are computed by running the four-state magnitude-parity automaton on the digits. Verification is then restricted to the 1956 sequences whose model prediction matches the ground-truth label on clean inputs (split: 991 for y=0, 965 for y=1). For the T-scaling experiment (Figure 3) we pin T to a single value and regenerate the test slice at each $T \in \{4, 6, \ldots, 20\}$; the per-$T$ slice contains $\sim$1000 class-balanced sequences (post-filter $\sim$970–988). For MNIST-extended we similarly construct a class-balanced slice of 100 sequences (50 per label, fixed T=30, seed 123) and verify the 96 sequences passing the clean-prediction filter. For CAVIAR we use the standard NeSyA train/test split of the dataset [1], downsample each clip at 1 fps (stride 25), and re-run the deterministic three-state automaton on the ground-truth simple events at the kept frames to obtain a downsampled state trajectory; the 13 of 17 test clips that match the target on clean inputs are kept for verification.

Per-experiment settings. Table 1 (NeSyA-MNIST, four structural variants, $T \in [3, 10]$) and the per-label split tables in Appendix F use $\varepsilon \in \{0.001, 0.002, 0.005, 0.01\}$ and verification batch size 100. Table 4 (MNIST-extended, three structural variants) uses $\varepsilon \in \{10^{-4}, 2{\times}10^{-4}, 5{\times}10^{-4}, 10^{-3}\}$ and verification batch size 4 to fit the K=216 joint alphabet at T=30 in GPU memory. Table 2 (CAVIAR, three variants) uses $\varepsilon \in \{2{\times}10^{-5}, 5{\times}10^{-5}, 10^{-4}, 2{\times}10^{-4}\}$ and verification batch size 1 (forced by the per-clip accepting-state spec, which captures a per-sequence one-hot accepting mask as a constant in the bound graph). Figure 3 (T-scaling) uses $\varepsilon = 0.005$ across $T \in \{4, 6, \ldots, 20\}$ and verification batch size 64. All main-table experiments disable the optional transition and state simplex refinements (Appendix B ablates these on the CLG temporal-operator stack at ε=0.01).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the setting, methods, and empirical scope; the claims are supported by Sections 3–4 and the appendices.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations paragraph in Section 6 discusses perception-network bottlenecks, path-unrolling scalability, and DFA-only / calibrated-probability semantic scope.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Assumptions are stated with the relevant results in Sections 3.2–3.3, with complete proofs in Appendices A and B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 describes the compared methods, threat model, datasets, sample sizes, and evaluation metrics; implementation details accompany the code release.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The experiments use public datasets (MNIST and CAVIAR), and the code release at `https://github.com/Abdu-Hekal/nesyveri` provides the scripts and instructions for reproducing the reported runs.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section G and the benchmark-specific subsections specify the verification setup, splits or sample sizes, radii, horizons, methods, and software stack.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: Our experiments evaluate deterministic robustness certificates for fixed trained perception networks, fixed automata, and fixed benchmark test sequences. Each certified fraction is computed directly from sound bound propagation on those instances, with no stochastic training, random sampling, or repeated trial variability being estimated.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section G specifies the CPU, GPU, RAM, and operating system used for all runs, and the experiment tables report wall-clock time for each configuration.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work studies robustness certification using public benchmark data and does not involve human subjects, sensitive data, or high-risk deployment.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is foundational research on robustness certification methodology; the work is not tied to a particular deployment, and the paper does not discuss broader societal impacts at this stage.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, or generative assets requiring special release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The pre-trained architectures, datasets, and software frameworks used in the experiments are cited in Section 4 and respected; their open-source licenses are unchanged.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces a verification methodology rather than a new dataset or benchmark. Its implementation, together with the perception-network checkpoints used in the experiments, is released at `https://github.com/Abdu-Hekal/nesyveri` under GPL-3.0 and documented in the repository README.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development and experiments do not use LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.