ZONOGPT: TOWARDS AN ABSTRACT DOMAIN FOR VERIFYING LARGE GPT MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer-based models are widely used for reasoning, coding, and multimodal agentic tasks. To provide formal assurance of desirable behaviors, such as robustness, safety, and fairness, neural network verification techniques prove required properties and provide auditable guarantees before deployment. However, prior work remains limited to small or restricted Transformers, and maintaining precision across deep models remains challenging. In this work, we introduce ZONOGPT, an abstract domain for verifying large transformers that maintains a space complexity independent of network depth. ZONOGPT uses a structured zonotope and a generator reduction mechanism to efficiently preserve correlations. To maintain precision, it introduces block-specific fused transformations for Attention and LayerNorm that retain feature relations, along with an affine transform for GELU that preserves generator relations. These mechanisms enable ZONOGPT to be the first approach to verify standard architectures, scaling to official HuggingFace models up to GPT-2 Medium (24 blocks, 300M+ parameters) and successfully verifying 1,339 instances across text and vision tasks.

1 INTRODUCTION

Transformer-based models (Vaswani et al., 2017), *e.g.*, generative pretrained transformer (GPT) (Radford et al., 2019), are widely used for reasoning, coding, and multimodal agentic tasks (Comanici et al., 2025; Liu et al., 2024). To provide formal assurance that these models have desirable behaviors, such as robustness (Jia et al., 2019; Zhu et al., 2024), safety (Zou et al., 2023) and fairness (Gallegos et al., 2024; Wang et al., 2023), researchers have developed Neural Network Verification (NNV) techniques that prove required properties of neural networks, which include Transformer-based models, and provide auditable guarantees before release or deployment (Shi et al., 2020; Bonaert et al., 2021; Huang et al., 2026).

While abstraction (Cousot & Cousot, 1977; Singh et al., 2018; Wang et al., 2021) has successfully scaled NNV to large models (Kaulen et al., 2025), verification remains limited to small or restricted transformers. Existing work handles at most three small simplified blocks (Shi et al., 2020; Huang et al., 2026). Bonaert et al. (2021) scales to 12 blocks with much smaller configurations. None of these tools scale to standard transformer architectures, *e.g.*, GPT-2 (Radford et al., 2019). We discuss these works in more detail in §B.

In addition, beyond scalability, maintaining precision across deep transformers remains challenging, *e.g.*, 24 blocks in GPT-2 Medium. Abstract domains inherently lose precision as they propagate through successive non-linear layers (Wang et al., 2018; Singh et al., 2019), *e.g.*, while COVENN scales via decomposition (Duong et al., 2025b), its interval bounds between blocks substantially degrade precision. Furthermore, recent techniques only tighten isolated operations, such as softmax (Wei et al., 2023) or attention products (Zhang et al., 2024; Huang et al., 2026), and does not address the over-approximation error that accumulates across layers.

In this paper, we introduce ZONOGPT, an abstract domain for verifying large transformers that maintains a space complexity (*e.g.*, memory footprint) independent of network depth while preserving relations between tokens and internal features. First, ZONOGPT uses structured zonotopes capturing: (i) global sequence correlations through shared generators and (ii) localized variations through local generators (§4.1). Second, reduction mechanisms limit number of generators after each block (§4.4), ensuring memory footprint remain bounded and independent of network depth (§4.5).

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

To maintain precision, ZONOGPT introduces block-specific abstract transformations tailored to transformer components. Particularly, ZONOGPT applies a fused transform for attention residual block (§4.2) that jointly over-approximates the skip and attention branches. By preserving the shared generator structure between the two, ZONOGPT yields a tighter bound than bounding each branch independently. For the MLP residual block, ZONOGPT further improves precision by employing a fused transform for LayerNorm to preserve feature relations within each token (§4.3.1), along with an affine transform for GELU (§4.3.2).

Evaluated on 2,880 text and vision instances, ZONOGPT successfully verifies 1,339 instances (46.5%), scaling to official HuggingFace models up to GPT-2 Medium (24 blocks, 300M+ parameters). As perturbation increases, fused attention proves critical for deep configurations (4+ blocks), *e.g.*, 867 verified instances (40.1%), while unfused variants verify fewer than 1%.

Contributions. The primary contributions of this paper include: (i) A structured zonotope (§4.1) and generator reduction mechanism (§4.4) that preserve global and local relations with a depth-independent space complexity; (ii) Fused transforms for attention (§4.2) and LayerNorm (§4.3.1) that preserve feature relations, and an affine GELU transform (§4.3.2) that retains input generators; (iii) Verification of official HuggingFace GPT-2 architectures adapted for vision tasks, scaling to GPT-2 Medium (300M+ parameters) under full-dimensional input perturbations (§5).

While the GPT models considered by ZONOGPT is still very small compared to the largest ones, it is the first approach to verify standard architectures, scaling to official HuggingFace models up to GPT-2 Medium (24 blocks, 300M+ parameters) (§5). We view this as a first step towards verifying larger GPT models, and we hope that our work will inspire future research in this direction.

2 BACKGROUND

Neural Network Verification. Given a neural network N and a property ϕ, the NNV problem asks whether ϕ is a valid property of N. Typically, ϕ is a formula $\phi_{in} \Rightarrow \phi_{out}$, where ϕ_{in} and ϕ_{out} are properties over the inputs and outputs of N, respectively. An NNV tool searches for a *counterexample* input to N that satisfies ϕ_{in} but violates ϕ_{out}: if none exists, ϕ is valid; otherwise, ϕ is invalid (Wang et al., 2021; Ferrari et al., 2022; Wu et al., 2024; Duong et al., 2024). Modern verifiers often use branch-and-bound (Bunel et al., 2020), whose bounding step computes all reachable outputs efficiently (Singh et al., 2018; 2019; Wang et al., 2018; Zhou et al., 2024; 2025).

Among those, zonotope (Singh et al., 2018) is the abstraction that balances precision and scalability:

$$\mathcal{Z}(c, G) = \{c + G\epsilon : \epsilon \in [-1, 1]^m\} \tag{1}$$

where c is the center and each column of G is a generator controlled by one coefficient in ϵ. The same coefficient can influence multiple outputs, allowing a zonotope to preserve their relations. Zonotope transformation for affine layer is exact, *e.g.*, $W\mathcal{Z}(c, G) + b = \mathcal{Z}(Wc + b, WG)$, while nonlinear operations require additional generators to bound their approximation errors. These generators improve precision but make the computational complexity grow with network sizes.

ZONOGPT builds on zonotopes with shared generators for cross-token and local generators for token-specific relations (§4.1). Generator reduction then limits their number after each block (§4.4), preserving the relations needed by Transformer operations without growth across network depth.

Residual Taylor Expansion. Let $A(x) = x + B(x)$ be a twice continuously differentiable residual, $Z = \{c + \sum_j H_j\epsilon_j : |\epsilon_j| \leq 1\}$ be its input zonotope. For every $x \in Z$, the change from the center is $v = x - c = \sum_j H_j\epsilon_j$. Define $g_v(t) = A(c + tv)$ for $t \in [0, 1]$. Taylor's theorem along g_v gives

$$A(c + v) = A(c) + J_A(c)v + e_A(c, v), \qquad e_A(c, v) = \int_0^1 (1 - t)g_v''(t)\, dt \tag{2}$$

where $J_f(c)$ denotes the Jacobian matrix of a function f evaluated at c. Since $A(x) = x + B(x)$, the identity $J_A(c) = I + J_B(c)$ turns the linear term in Eq. 2 into $\sum_j (H_j + J_B(c)H_j)\epsilon_j$. Thus, the Taylor expansion of $A(Z)$ is:

$$A(Z) \subseteq Z' = \Big\{A(c) + \sum_j \big(H_j + J_B(c)H_j\big)\epsilon_j + e_A(c, v) : |\epsilon_j| \leq 1\Big\} \tag{3}$$

ZONOGPT uses Eq. 3 to construct the output zonotope for attention (§4.2) and LayerNorm (§4.3.1).

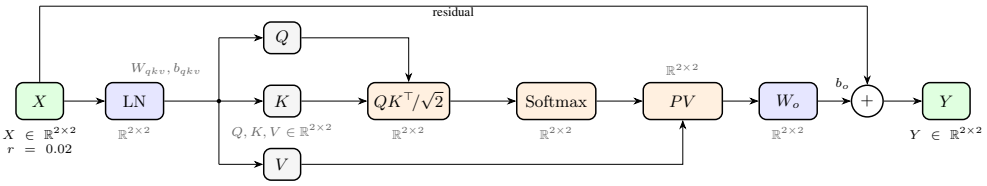


Fig. 1: Running example for the attention residual for $S = 2$ tokens, hidden size $d = 2$, one head.

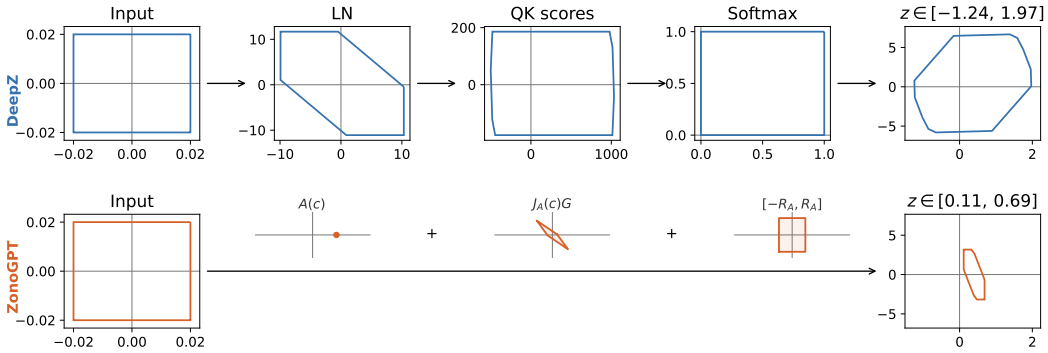


Fig. 2: First-token projections for operator-wise DEEPZ (top) and fused ZONOGPT (bottom).

3 MOTIVATING EXAMPLE

We demonstrate ZONOGPT on an attention-residual block with $S = 2$ tokens and hidden size $d = 2$ (Fig. 1). We verify that its first output coordinate remains positive given an ℓ_∞ perturbation:

$$\varphi \equiv \|x - x_0\|_\infty \le r \Longrightarrow [A(x)]_{1,1} > 0, \qquad x_0 = [[0.0, 0.0], [0.7, 0.8]], \qquad r = 0.02 \tag{4}$$

Scalability. DEEPZ represents the input with Sd dense generators, each containing Sd coefficients, and therefore stores S^2d^2 coefficients at the input. For the running example ($S = 2$, $d = 2$):

$$\begin{aligned} x_{11} &= x_{11}^0 + r\epsilon_1 + 0\epsilon_2 + 0\epsilon_3 + 0\epsilon_4 \quad & x_{12} &= x_{12}^0 + 0\epsilon_1 + r\epsilon_2 + 0\epsilon_3 + 0\epsilon_4 \\ x_{21} &= x_{21}^0 + 0\epsilon_1 + 0\epsilon_2 + r\epsilon_3 + 0\epsilon_4 \quad & x_{22} &= x_{22}^0 + 0\epsilon_1 + 0\epsilon_2 + 0\epsilon_3 + r\epsilon_4 \end{aligned} \tag{5}$$

DEEPZ stores all 16 coefficients in Eq. 5, *e.g.*, $(r, 0, 0, 0)$ for x_{11}, although each generator affects only one token. ZONOGPT stores the same input without the cross-token zeros:

$$x_{11} = x_{11}^0 + r\epsilon_1 + 0\epsilon_2 \qquad x_{12} = x_{12}^0 + 0\epsilon_1 + r\epsilon_2 \qquad x_{21} = x_{21}^0 + r\epsilon_3 + 0\epsilon_4 \qquad x_{22} = x_{22}^0 + 0\epsilon_3 + r\epsilon_4 \tag{6}$$

ZONOGPT stores eight coefficients (instead of the 16). For the GPT-2-small input with $S = 512$ and $d = 768$, ZONOGPT uses $Sd^2 \cdot 8/2^{30} = 2.25$ GiB, while DEEPZ uses $S^2d^2 \cdot 8/2^{30} = 1152$ GiB. For hidden states after each residual component, ZONOGPT stores generators at the source token:

$$Z_s = c_s + \sum_{j=1}^{m} G_{j,s}\epsilon_j + L_s\epsilon_s^{\text{loc}} + [-b_s, b_s] \tag{7}$$

ZONOGPT uses G for shared generators, L_s for generators local to token s, and b_s for the remaining independent values. With at most m shared and q local generators per token, these terms require $O(Sd(m + q + 1))$ coefficients independent of network depth (§4.5). In contrast, DEEPZ adds $O(Sd)$ dense generators per block. Each dense generator contains Sd coefficients, so one block adds $O(S^2d^2)$ coefficients and L blocks add $O(LS^2d^2)$.

Precision. We use the network in Fig. 1 with the following fixed parameters:

$$W_{qkv}^\top = \begin{bmatrix} 3.1 & -0.2 & 1.0 & 0.1 & 0.8 & -3.0 \\ 0.7 & -1.1 & -4.3 & 1.2 & -0.5 & 0.6 \end{bmatrix}, \qquad W_o = \begin{bmatrix} 0.112 & 0.029 \\ 0.014 & 0.153 \end{bmatrix} \tag{8}$$

$b_{qkv} = (-0.1, 0.1, 0, 0.2, 0.2, -0.2)$, $b_o = (0.32122, 0.95628)$, $\gamma = (1.6, 1.8)$, $\beta = (0.2, 0.3)$, and $\tau = 10^{-5}$. We represent the input interval by the zonotope

$$X = \begin{bmatrix} 0 & 0 \\ 0.7 & -0.8 \end{bmatrix} + 0.02 \begin{bmatrix} \epsilon_{11} & \epsilon_{12} \\ \epsilon_{21} & \epsilon_{22} \end{bmatrix}, \qquad c = \begin{bmatrix} 0 & 0 \\ 0.7 & -0.8 \end{bmatrix}, \qquad \epsilon_{ij} \in [-1, 1] \tag{9}$$

Alg. 1: Structured-zonotope bound propagation

Input: $F = \Phi_T \circ \cdots \circ \Phi_1$; input center and radius (x_0, r); m shared and q local generators
Output: $(\underline{y}, \overline{y})$ bounding $F(x)$ for every $\|x - x_0\|_\infty \le r$

1 $Z \leftarrow \mathcal{Z}(x_0, 0, \{\mathrm{diag}(r_s)\}_{s=1}^{S}, 0)$ // Initialize local input generators (§4.1)
2 **for** $t \leftarrow 1$ **to** T **do**
3 **for** $s \leftarrow 1$ **to** S **do** // Represent $|e_s| \le b_s$ with local generators (§4.4)
4 $I_s \leftarrow \{i : (b_s)_i > 0\}$; $L_s \leftarrow [L_s \mid \mathrm{diag}(b_s)_{:,I_s}]$; $b_{s,I_s} \leftarrow 0$
5 **if** Φ_t *is an attention block* **then**
6 $Z \leftarrow \mathsf{FusedAttentionTransform}(\Phi_t, Z)$ // Bound Attention residual (§4.2)
7 **else**
8 $Z \leftarrow \mathsf{SequentialMLPTransform}(\Phi_t, Z)$ // Bound MLP residual (§4.3)
9 $K \leftarrow \mathsf{SelectShared}(G, m)$ // Select at most m shared generators (§4.4)
10 $b \leftarrow b + \sum_{j \notin K} |G_j|$; $G \leftarrow G_K$
11 **for** $s \leftarrow 1$ **to** S **do** // Select at most q local generators (§4.4)
12 $K_s \leftarrow \mathsf{SelectLocal}(L_s, q)$; $b_s \leftarrow b_s + \sum_{j \notin K_s} |L_{s,j}|$; $L_s \leftarrow L_{s,K_s}$
13 **return** $\mathrm{concretize}(Z)$

The top row of Fig. 2 follows DEEPZ through LayerNorm, $QK^\top$, and softmax. Each nonlinear operator bounds its input with fresh generators and the attention output has growth to 34 generators:

$$[A(X)]_{1,1} = z_0 + \textstyle\sum_{i=1}^{34} g_i \epsilon_i, \qquad z_0 \approx 0.36, \qquad \textstyle\sum_i |g_i| \approx 1.61 \tag{10}$$

Concretizing Eq. 10 gives $[A(X)]_{1,1} \in [-1.24, 1.98]$ (endpoints rounded outward), and cannot verify the property of $[A(X)]_{1,1} > 0$ (Eq. 4).

ZONOGPT instead bounds the attention-residual component as one *fused* block (§4.2):

$$A(x) \in A(c) + J_A(c)G\epsilon + [-R_A, R_A], \qquad \|x - x_0\|_\infty \le r \tag{11}$$

The bottom row of Fig. 2 separates the three contributions in Eq. 11. The output center $A(c)$ computes the attention output at the input center $x = c$. The Jacobian image $J_A(c)G$ carries each input generator through both skip and attention branch, preserving correlations between the two branches. The fused generators encode the nonlinear remainder bounded by R_A. Adding the three terms together gives the final bound, $[A(X)]_{1,1} \in [0.11, 0.69]$, and verifies $[A(X)]_{1,1} > 0$.

4 THE ZONOGPT APPROACH

Alg. 1 describes how ZONOGPT computes bounds for an input interval (x_0, r) through a Transformer F. First, ZONOGPT represents the input interval with local generators (line 1). For each residual component, ZONOGPT represents $|e| \le b$ with zonotope generators (line 4). It applies fused transformation to the attention block (line 6) and propagates layer-by-layer through the MLP block (line 8). ZONOGPT limits shared generators to m (line 9 to line 10). It also limits local generators to q per token (line 12). Finally, ZONOGPT returns the output bounds (line 13).

The key idea is to preserve dependencies that can tighten later bounds. During bound propagation, ZONOGPT uses *shared generators* for cross-token relations (expensive) and *local generators* for token-specific relations (linear cost in the sequence length), while *intervals* keep the number of generators from growing with network depth.

4.1 STRUCTURED REPRESENTATION

ZONOGPT represents the input using local generators (line 1) because each input coordinate initially affects only its token. Attention later mixes tokens, so ZONOGPT uses shared generators to preserve cross-token relations and an interval to bound the remaining terms independently.

Definition 1 (Structured Zonotope)**.** *For S tokens, let c be the center, m shared generators G, q_s local generators L_s at token s, and $b \ge 0$ the interval radius. The represented set is:*

$$\mathcal{Z}(c, G, L, b) = \Big\{ c + \sum_{j=1}^{m} G_j \epsilon_j + \sum_{s=1}^{S} \mathrm{emb}_s(L_s \epsilon_s^{\mathrm{loc}}) + e : \|\epsilon\|_\infty \le 1, \|\epsilon_s^{\mathrm{loc}}\|_\infty \le 1,\ |e| \le b \Big\} \tag{12}$$

$\mathrm{emb}_s(L_s\epsilon_s^{\mathrm{loc}})$ *forms a matrix whose s-th row is* $L_s\epsilon_s^{\mathrm{loc}}$ *and whose other rows are zero. A generator* G_j *can change several tokens through the same* ϵ_j*, while a generator in* L_s *changes only token s.*

The coordinate radius of $Z = \mathcal{Z}(c, G, L, b)$ is $r(Z)$ and interval bounds $[c - r(Z), c + r(Z)]$:

$$r(Z) = \sum_{j=1}^{m} |G_j| + \sum_{s=1}^{S} \mathrm{emb}_s\Big(\sum_{j=1}^{q_s} |L_{s,j}|\Big) + b \tag{13}$$

4.2 Fused Attention Block Transform

For input x, let $A(x)$ denote the attention output with the skip connection. Let n be the LayerNorm output. For head h, let Q_h, K_h, and V_h be the projected queries, keys, and values. Let U_h be the score matrix and P_h the softmax output:

$$\begin{aligned} n &= \mathrm{LN}(x), \qquad Q\|K\|V = W_{qkv}n + b_{qkv}, \qquad U_h = Q_h K_h^\top/\sqrt{d_h}, \qquad P_h = \mathrm{softmax}(U_h) \\ A(x) &= x + W_o \underbrace{[\,P_1V_1 \mid \cdots \mid P_HV_H\,]}_{\mathrm{concat}(P_hV_h)} + b_o \end{aligned} \tag{14}$$

Note that attention contains the related products $QK^\top$ and PV, while softmax couples every key in a row of U_h. Bounding these operations separately loses relations among their terms before adding to the skip branch. ZonoGPT instead propagates each input generator through the complete attention residual and bounds only the remaining nonlinear error.

To construct output bounds for A, ZonoGPT computes the three terms in Eq. 3, *e.g.*, $A(c)$, $H_j + J_B(c)H_j$, and $e_A(c, v)$. First, ZonoGPT computes $A(c)$ simply by evaluating Eq. 14 at $x = c$.

Second, ZonoGPT computes how A changes when the coefficient of H_j varies from zero:

$$x(t) = c + tH_j, \qquad \dot{x} = \left.\frac{dx(t)}{dt}\right|_{t=0} = H_j, \qquad \dot{A} = \left.\frac{dA(x(t))}{dt}\right|_{t=0} = J_A(c)H_j \tag{15}$$

Applying the chain rule along this path gives:

$$\begin{aligned} &\dot{n} = J_{\mathrm{LN}}(c)\dot{x}, \qquad \dot{Q}\|\dot{K}\|\dot{V} = W_{qkv}\dot{n}, \qquad \dot{U} = (\dot{Q}K^\top + Q\dot{K}^\top)/\sqrt{d_h}, \qquad \dot{P} = J_{\mathrm{SM}}(U)\dot{U} \\ &\dot{A} = \dot{x} + W_o\,\mathrm{concat}(\dot{P}_hV_h + P_h\dot{V}_h) \end{aligned} \tag{16}$$

where each undotted quantity (*e.g.*, Q, K, V, U, and P) denotes its value evaluated at $x = c$. Eq. 15 and Eq. 16 give $J_A(c)H_j = H_j + J_B(c)H_j = \dot{x} + W_o\,\mathrm{concat}(\dot{P}_hV_h + P_h\dot{V}_h)$.

Third, Taylor expansion identifies error term $e_A(c, v)$ as $A(c+v) - A(c) - J_A(c)v$ (Eq. 2). ZonoGPT then computes R_A satisfying:

$$|e_A(c, v)| = |A(c + v) - A(c) - J_A(c)v| \le R_A \tag{17}$$

Intuitively, R_A bounds the nonlinear terms introduced by $QK^\top$, softmax, and PV. ZonoGPT computes a bound R_A for the nonlinear attention error e_A using Thm. 1.

Theorem 1 (Attention nonlinear-error bound)**.** *Let* A *be the attention residual in Eq. 14, and the input zonotope be* $Z = \{c + \sum_{j=1}^{m} H_j\epsilon_j : |\epsilon_j| \le 1\}$*. Let* $P = P(c)$ *and* $V = V(c)$*. Assume nonnegative tuples* (f_P, R_P, f_V, R_V) *denote the bounds on the first-order change (*f*) and the Taylor remainder (*R*) of* P *and* V*, respectively. satisfy* $|J_P(c)(x - c)| \le f_P$*,* $|P(x) - P - J_P(c)(x - c)| \le R_P$*,* $|J_V(c)(x - c)| \le f_V$*, and* $|V(x) - V - J_V(c)(x - c)| \le R_V$ *for every* $x \in Z$*, then*

$$\underbrace{|A(x) - A(c) - J_A(c)(x - c)|}_{e_A} \le \underbrace{|W_o|\,\mathrm{concat}\,(R_P|V| + PR_V + (f_P + R_P)(f_V + R_V))}_{R_A} \tag{18}$$

Proof sketch. Expanding the attention-value product gives the three terms on the right-hand side of Eq. 18. The triangle inequality bounds these terms by $R_P|V|$, PR_V, and $(f_P + R_P)(f_V + R_V)$, respectively. ZonoGPT derives f_P, R_P, f_V, and R_V in §A.4. □

Thm. 1 bounds the nonlinear attention error with an interval. ZonoGPT improves precision by representing the part caused by the softmax error (*e.g.*, $|W_o|\,\mathrm{concat}(R_P|V|)$) with a zonotope. Applying the residual Taylor expansion in Eq. 2 to P gives $P(x) = P(c) + J_P(c)(x - c) + e_P$, hence

$e_P = P(x) - P(c) - J_P(c)(x-c)$. The entry $(e_P)_{ihj}$ is the softmax error for query token i, attention head h, and key token j. Each row of $P(x)$ and $P(c)$ sums to one (softmax normalizes each row), and each row of $J_P(c)(x-c)$ sums to zero (the derivative of this fixed row sum is zero). Thus, $\sum_j (e_P)_{ihj} = 1 - 1 - 0 = 0$.

Let $\bar{v}_{hj} = W_o^{(h)} V_{hj}(c)$ be the projected center value at key token j. Because the coefficients $(e_P)_{ihj}$ sum to zero, subtracting a_{ih} from every $\bar{v}_{hj}$ does not change their weighted sum. Since $|e_P| \le R_P$, we write $(e_P)_{ihj} = \xi_{ihj}(R_P)_{ihj}$ with $|\xi_{ihj}| \le 1$, which gives

$$\begin{aligned} \sum_j (e_P)_{ihj} \bar{v}_{hj} &= \sum_j (e_P)_{ihj} (\bar{v}_{hj} - a_{ih}) \\ &= \sum_j \xi_{ihj} (R_P)_{ihj} (\bar{v}_{hj} - a_{ih}) \end{aligned} \tag{19}$$

Thus, the softmax term is a combination of the generators $(R_P)_{ihj}(\bar{v}_{hj} - a_{ih})$ with coefficients ξ_{ihj}, and their radius $\sum_j (R_P)_{ihj} |\bar{v}_{hj} - a_{ih}|$ depends on a_{ih}. ZONOGPT chooses a_{ih} as an elementwise weighted median of $\bar{v}_{hj}$ with weights $(R_P)_{ihj}$, which minimizes this radius in output.

Finally, ZONOGPT computes a structured zonotope $Z_A = \mathcal{Z}(c_A, G_A, L_A, b_A)$ (Def. 1) for A:

$$\begin{aligned} c_A &= A(c), \qquad (G_A)_j = J_A(c) G_j, \\ (L_A)_i &= [\, (J_A(c)\, \mathrm{emb}_i(L_{i,k}))_i \,]_{k=1}^{q_i} \;\big|\; [\, (R_P)_{ihj} (\bar{v}_{hj} - a_{ih}) \,]_{h,j}, \\ (b_A)_i &= [|W_o|\, \mathrm{concat}\,(P R_V + (f_P + R_P)(f_V + R_V))]_i + \sum_{s \ne i} \sum_{k=1}^{q_s} |(J_A(c)\, \mathrm{emb}_s(L_{s,k}))_i| \end{aligned} \tag{20}$$

Thm. 2 establishes soundness of the output zonotope in Eq. 20.

Theorem 2 (Fused attention soundness)**.** *Let the input zonotope be $Z = \mathcal{Z}(c, G, L, 0)$ with local generators $L_i = [L_{i,1}, \ldots, L_{i,q_i}]$, and let A be the attention in Eq. 14. Suppose a nonnegative tuple (f_P, R_P, f_V, R_V) satisfies the four bounds in Thm. 1 for every $x \in Z$. For each query token i and head h, choose a_{ih} shared across all key tokens j. Then $Z_A = \mathcal{Z}(c_A, G_A, L_A, b_A)$ defined by Eq. 20 satisfies $A(Z) \subseteq Z_A$.*

Proof sketch. Eq. 19 represents the softmax error with the local generators in L_A. The other nonlinear terms satisfy the bounds in Thm. 1, while b_A bounds the cross-token parts of the local input generator images. The full proof is in §A.5. □

4.3 MLP BLOCK TRANSFORM

For input x, the MLP block computes

$$M(x) = x + W_2 g(W_1\, \mathrm{LN}(x) + b_1) + b_2 \tag{21}$$

The MLP contains two nonlinear operations with different structures. LayerNorm couples the features within each token through their mean and variance, so ZONOGPT bounds the complete LayerNorm as one fused transform (§4.3.1). GELU acts independently on each coordinate, and the surrounding affine maps are exact (§4.3.2). After LayerNorm, the MLP has no bilinear products such as $QK^\top$ and PV in attention (§4.2). ZONOGPT therefore propagates the affine maps and GELU sequentially before combining the block with the skip connection.

4.3.1 LAYERNORM

The mean and variance in LayerNorm depend on the same token features. Bounding these quantities separately loses this relation, so ZONOGPT treats the complete LayerNorm as one fused transform using the Taylor expansion in Eq. 3. At token s, the input zonotope has the form:

$$x_s = c_s + \sum_j (G_j)_s \epsilon_j + L_s \epsilon_s^{\mathrm{loc}} \tag{22}$$

Applying the Taylor expansion to the complete LayerNorm gives

$$\mathrm{LN}(x_s) = \mathrm{LN}(c_s) + \sum_j J_{\mathrm{LN}}(c_s)(G_j)_s \epsilon_j + J_{\mathrm{LN}}(c_s) L_s \epsilon_s^{\mathrm{loc}} + e_{N,s}, \qquad |e_{N,s}| \le (R_N)_s \tag{23}$$

Thus, ZONOGPT computes the LayerNorm output in the form of a structured zonotope as:

$$Z_N = \mathcal{Z}\Big(\mathrm{LN}(c), \{J_{\mathrm{LN}}(c)G_j\}_j, \{J_{\mathrm{LN}}(c_s)L_s\}_{s=1}^{S}, R_N\Big) \tag{24}$$

where $c = (c_1, \ldots, c_S)$ is the center, and $\mathrm{LN}(c) = (\mathrm{LN}(c_1), \ldots, \mathrm{LN}(c_S))$. ZONOGPT derives the remaining term R_N and proves its soundness (Thm. 3) in §A.1.

4.3.2 GELU

The MLP applies tanh-GELU elementwise:

$$g(t) = \frac{t}{2}\Big[1 + \tanh\Big(\sqrt{\frac{2}{\pi}}\left(t + 0.044715t^3\right)\Big)\Big] \tag{25}$$

For a GELU input zonotope $Z = \mathcal{Z}(c, G, L, b)$, ZONOGPT first computes the coordinate interval $[\ell, u]$. For each coordinate with $\ell_k < u_k$, ZONOGPT separates GELU into the linear term $a_k t$ and the remaining error $g(t) - a_k t$.

$$a_k = \frac{g(u_k) - g(\ell_k)}{u_k - \ell_k}, \qquad r_k^{\mathrm{lb}} \le g(t) - a_k t \le r_k^{\mathrm{ub}}, \qquad t \in [\ell_k, u_k] \tag{26}$$

where the error bounds r^{lb} and r^{ub} are computed in §A.2. With $m = (r^{\mathrm{lb}} + r^{\mathrm{ub}})/2$ and $d = (r^{\mathrm{ub}} - r^{\mathrm{lb}})/2$, Eq. 26 gives the affine GELU bound:

$$g(z) \in a \odot z + m + \{d \odot \epsilon^g : |\epsilon^g| \le 1\} \tag{27}$$

Applying Eq. 27 to Z gives the GELU output:

$$Z_g = \mathcal{Z}\Big(a \odot c + m, \{a \odot G_j\}_j, \{[\,\mathrm{diag}(a_s)L_s \mid \mathrm{diag}(d_s)\,]\}_{s=1}^{S}, |a| \odot b\Big) \tag{28}$$

GELU bounds are proved sound in Thm. 4 (§A.2).

4.4 ZONOTOPE CONVERSION AND REDUCTION

In Def. 1, $|e| \le b$ term means that every coordinate of e varies independently. Before applying block transform, ZONOGPT writes these values as local zonotope generators (line 4). The skip and branch then use the same ϵ for each value while maintaining the original set Z (Lem. 1).

Lemma 1 (Interval-to-zonotope conversion)**.** *A structured zonotope with S tokens satisfies*

$$\mathcal{Z}(c, G, L, b) = \mathcal{Z}\Big(c, G, \{[L_s \mid \mathrm{diag}(b_s)]\}_{s=1}^{S}, 0\Big) \tag{29}$$

Proof. The condition $|e_s| \le b_s$ is equivalent to $e_s = \mathrm{diag}(b_s)\epsilon_s$ for some $|\epsilon_s| \le 1$. □

After each block transform, ZONOGPT retains at most m shared and q local generators per token (line 9 to line 12). ZONOGPT ranks the generators by summed absolute coefficient magnitude. The selected shared generators remain in G, while $|G_j|$ is added to b for every discarded generator G_j. The resulting zonotope contains the original zonotope (Lem. 2).

Lemma 2 (Generator reduction)**.** *For any set K of retained shared-generator indices,*

$$\mathcal{Z}(c, G, L, b) \subseteq \mathcal{Z}\Big(c, G_K, L, b + \sum_{j \notin K} |G_j|\Big) \tag{30}$$

Proof. For discarded index j, $|G_j \epsilon_j| \le |G_j|$ because $|\epsilon_j| \le 1$. Summing these gives Eq. 30. □

4.5 COMPLEXITY

Space complexity. Between components, line 9 to line 12 retain at most m shared and q local generators per token for S tokens of width d. The center, b, shared and local generators thus require $O(Sd) + O(mSd) + O(Sqd) = O(Sd(m+q+1))$. For an attention, Eq. 29 adds $O(Sd^2)$ temporary coefficients before reduction, while the softmax for H heads adds $O(HS^2d)$ and dominates the $O(HS^2)$ attention matrices. The space complexity of ZONOGPT is therefore:

$$O\big(Sd(m + q + 1) + Sd^2 + HS^2d\big) \tag{31}$$

Time complexity. Let T_Φ be the time for one full-component Jacobian vector product (JVP) , T_{tok} the time for one source-token attention JVP, and T_R the time for computing the nonlinear remainder bound. ZONOGPT has the following time complexity:

$$T_{\mathrm{A}} = O\big(mT_\Phi + S(q + d)T_{\mathrm{tok}} + T_R\big), \qquad T_{\mathrm{MLP}} = O\big((m + q + d)T_\Phi + T_R\big) \tag{32}$$

Tab. 1: Benchmark models and instances.

Type	Model	Blocks	Width	Heads	Parameters	Instances
Vision	GPT-2 Small	$\{2, 4, 8, 12\}$	768	12	14.2M - 85.1M	720
	GPT-2 Medium	$\{2, 4, 8, 24\}$	1024	16	25.3M - 302.4M	720
Text	GPT-2 Small	$\{2, 4, 8, 12\}$	768	12	52.8M - 123.7M	720
	GPT-2 Medium	$\{2, 4, 8, 24\}$	1024	16	76.7M - 353.8M	720
Total						**2880**

Tab. 2: Verification results (runtime in seconds). A dash (-) denotes out-of-memory errors.

Dataset	Model	Block	IBP		DeepZ		$\alpha\beta$-CROWN		CoVeNN		ZonoGpt-M		ZonoGpt-D		ZonoGpt-F		ZonoGpt-A	
			Solved	Time	Solved	Time	Solved	Time	Solved	Time	Solved	Time	Solved	Time	Solved	Time	Solved	Time
MNIST	GPT-2 Small	2	0	3.6	-	-	-	-	0	5769.8	36	215.3	36	826.2	108	544.5	**144**	1507.2
		4	0	6.5	-	-	-	-	0	11256.8	0	464.1	0	1423.8	**108**	2463.5	**108**	4327.8
		8	0	12.2	-	-	-	-	0	22244.4	0	824.2	0	2942.5	63	5609.2	**90**	10621.9
		12	0	18.3	-	-	-	-	0	33440.1	0	1232.9	0	4435.2	54	6688.8	**72**	15862.0
	GPT-2 Medium	2	0	3.6	-	-	-	-	0	10472.1	36	291.9	36	729.3	108	788.3	**126**	1463.0
		4	0	7.0	-	-	-	-	0	20718.1	0	592.3	0	1506.0	99	2531.2	**108**	3783.4
		8	0	13.2	-	-	-	-	0	41476.1	0	876.4	0	3048.2	54	4800.8	**81**	8493.9
		24	0	39.2	-	-	-	-	0	71257.8	0	2612.2	0	9595.7	**9**	5105.6	**9**	28909.9
SST	GPT-2 Small	2	0	50.2	-	-	-	-	0	8122.4	24	112.5	50	581.9	87	790.2	**94**	1962.7
		4	0	77.6	-	-	-	-	0	28785.1	0	343.9	0	1307.2	67	3815.7	**81**	5260.0
		8	0	61.1	-	-	-	-	0	28194.8	0	649.3	1	2626.8	46	7849.6	**61**	11036.0
		12	0	68.5	-	-	-	-	0	41694.3	0	1022.4	0	3992.4	72	12677.0	**84**	17096.6
	GPT-2 Medium	2	0	66.5	-	-	-	-	0	15233.8	42	143.5	66	729.9	91	1027.2	**108**	2105.2
		4	0	70.3	-	-	-	-	0	27953.7	0	259.7	10	1548.2	74	3018.8	**80**	4919.5
		8	0	78.8	-	-	-	-	0	53430.5	1	510.1	2	2955.6	56	6984.3	**68**	10102.5
		24	0	112.2	-	-	-	-	0	149360.1	0	1507.0	0	8472.1	24	18784.8	**25**	29873.8
Total			0	688.8	-	-	-	-	0	569409.8	139	11657.6	201	46720.8	1120	83479.4	**1339**	157325.4

5 Evaluation

5.1 Experimental Design

Benchmark. We evaluate ZonoGpt on GPT-2-based classifiers that we trained on the MNIST and SST datasets. We adopted the exact configuration of GPT-2 model (Radford et al., 2019) and utilize its reference implementation from HuggingFace[1] In total, we generated 2880 instances across networks and perturbation radii (Tab. 1). For MNIST, image rows become tokens, and we perturb pixels within an ℓ_∞ ball. For SST, we perturb up to three token embeddings within an ℓ_∞ ball.

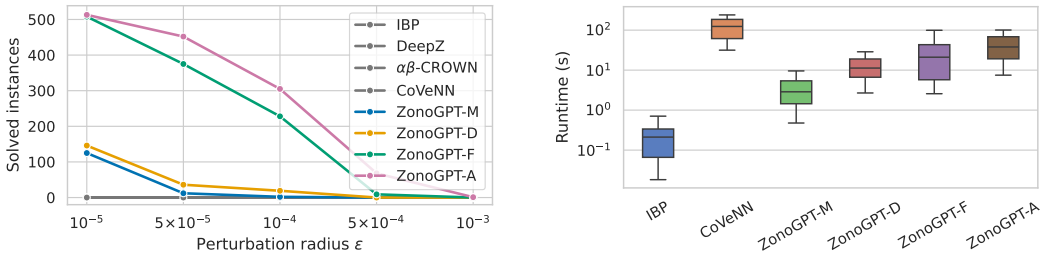
Baselines. We compare ZonoGpt with IBP (Wang et al., 2018) (interval), DeepZ (Singh et al., 2018) (zonotope), $\alpha\beta$-CROWN (Wang et al., 2021; Zhou et al., 2025) (polytope), and CoVeNN (Duong et al., 2025b) (polytope). Among ZonoGpt's variants, ZonoGpt-D fuses neither blocks (*D*ecomposed), ZonoGpt-A fuses only *A*ttention (§4.2), ZonoGpt-M fuses only the *M*LP (§A.3), and ZonoGpt-F fuses both (*F*ull). We exclude DeepT (Bonaert et al., 2021) and PBVerifier (Huang et al., 2026) as they do not support GELU and their post-LayerNorm attention residual (BERT-style) differs from our GPT-based target. All methods run on an NVIDIA GeForce RTX 4090 (24 GB VRAM), an AMD Ryzen Threadripper PRO 5975WX (128 GB RAM).

5.2 RQ1: Overall Verification Performance

Tab. 2 reports verified instances for 2880 instances across MNIST and SST benchmarks. ZonoGpt successfully verifies properties across all network depths, with ZonoGpt-A certifying 46.5% (1339 of 2880) of all instances. In contrast, baseline methods verify none, as they either exceed memory limits (DeepZ, $\alpha\beta$-CROWN) or produce loose bounds (IBP, CoVeNN).

The results validate our block-specific transformations. For the attention block, the fused transform provides the precision necessary to scale to the 24-block GPT-2 Medium. On models with more than two blocks, ZonoGpt-A verifies 40.1% of instances (867 of 2160), whereas unfused-attention variants solve fewer than 1% (ZonoGpt-D solved 13, ZonoGpt-M solved one of 2160). For the MLP block, sequential processing yields higher verified instances than block-level fusion, support-

[1]Hugging Face is a widely adopted open-source library that provides standardized reference implementations for SOTA transformer models. We specifically use their `transformers` package (Wolf et al., 2020).



(a) Verified instances by perturbation radius.

(b) Per-instance runtime (log scale).

Fig. 3: Verification and runtime statistics.

ing our analysis in §4.3. Computing the exact affine and elementwise GELU layers sequentially increases verified instances by 19.6% (ZONOGPT-A's 1339 vs. ZONOGPT-F's 1120).

The runtimes reflect the computational cost of maintaining precision. Baselines such as IBP and COVENN run quickly, but these loose abstractions fail to verify any instances. In contrast, ZONOGPT incurs more computational overhead to compute tighter bounds, *e.g.*, ZONOGPT-A accrues a substantially higher total runtime (157325s) compared to IBP (689s).

5.3 RQ2: FUSED TRANSFORMATION EFFECTIVENESS

To isolate the impact of our fused abstractions, we evaluate performance across varying perturbation radii ($\epsilon \in [10^{-5}, 10^{-3}]$), as shown in Fig. 3a. While all baselines verify zero instances across all tested radii, the ablation of ZONOGPT variants highlights the necessity of the fusion. Variants lacking a fused attention block (ZONOGPT-D and ZONOGPT-M) verify only 146 and 125 instances at the tightest radius ($\epsilon = 10^{-5}$) and drop to near zero by $\epsilon = 10^{-4}$.

Preserving the fused attention allows ZONOGPT-A and ZONOGPT-F to scale to much larger perturbations. Although both perform similarly at $\epsilon = 10^{-5}$, ZONOGPT-A performance is consistently better than ZONOGPT-F across all radii. In particular, at $\epsilon = 10^{-4}$, ZONOGPT-A verifies 305 instances compared to ZONOGPT-F's 228, and at $\epsilon = 5 \times 10^{-4}$, ZONOGPT-A certifies 68 instances while ZONOGPT-F drops to 9. This highlights the importance of the precision of ZONOGPT-A's sequential MLP bounds, which are critical as the radius grows.

5.4 RQ3: RUNTIME ANALYSIS

We evaluate computational efficiency using runtime distributions in Fig. 3b. IBP finishes instantly (under 1s) by relying on cheap but imprecise interval bounds, whereas COVENN incurs massive overhead (median $>$ 150s) due to computationally expensive CROWN analysis within each block. Nonetheless, both methods fail to maintain the tight bounds required to scale to deep networks.

Within the ZONOGPT variants, runtime predictably scales with abstraction granularity. Fusing the MLP block reduces execution time compared to tracking it sequentially, making ZONOGPT-M (3s) faster than ZONOGPT-D (11s), and ZONOGPT-F (21s) faster than ZONOGPT-A (38s). However, the fused attention abstraction requires slightly more computation to preserve precision, *e.g.*, ZONOGPT-F/ZONOGPT-A run slower than ZONOGPT-M/ZONOGPT-D, respectively. This is because the fused mechanism for attention must capture interactions across multiple paths (including queries, keys, values, and skip connections) into a unified abstract structure.

6 CONCLUSION

We presented ZONOGPT, an abstract domain for verifying large transformers through structured zonotopes and block-specific fused transformations, and successfully verified standard HuggingFace GPT-2 Medium. Future work includes jointly over-approximating cached attention states for autoregressive decoding (Pope et al., 2023) and adapting to Diffusion (Peebles & Xie, 2023).

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of Lean mechanized proofs for Thm. 1 (attention nonlinear-error bound), Thm. 2 (fused attention soundness), Lem. 1 (interval-to-zonotope conversion), Lem. 2 (generator reduction), Thm. 3 (LayerNorm soundness), Thm. 4 (affine GELU soundness), Thm. 5 (MLP nonlinear-error bound), and Thm. 6 (fused MLP soundness).

We have not used generative AI tools for helping developing theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, interpret results, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to create scientific figures: the Python code for Fig. 1 and the Python plotting code for Fig. 3. We have reviewed all AI-assisted work, *e.g.*, we read the Python code the AI tools generated for the figures and checked the plotted values against the raw data files. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All authors have read and adhere to the ICLR 2027 Code of Ethics.

REPRODUCIBILITY STATEMENT

ZONOGPT is available at `https://anonymous.4open.science/r/ZonoGPT`.

REFERENCES

Stanley Bak. nnenum: Verification of ReLU Neural Networks with Optimized Abstraction Refinement. In *NASA Formal Methods Symposium*, pp. 19–36. Springer, 2021. doi: 10.1007/978-3-030-76384-8_2.

Gregory Bonaert, Dimitar I Dimitrov, Maximilian Baader, and Martin Vechev. Fast and precise certification of transformers. In *Proceedings of the 42nd ACM SIGPLAN international conference on programming language design and implementation*, pp. 466–481, 2021.

Rudy Bunel, P Mudigonda, Ilker Turkaslan, P Torr, Jingyue Lu, and Pushmeet Kohli. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21(2020), 2020. URL `https://dl.acm.org/doi/10.5555/3455716.3455758`.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pp. 238–252, 1977.

Hai Duong, Dong Xu, Thanhvu Nguyen, and Matthew B. Dwyer. Harnessing neuron stability to improve dnn verification. *Proc. ACM Softw. Eng.*, 1(FSE), 2024. doi: 10.1145/3643765.

Hai Duong, ThanhVu Nguyen, and Matthew B Dwyer. Neuralsat: A high-performance verification tool for deep neural networks. In *International Conference on Computer Aided Verification*, pp. 409–423. Springer, 2025a.

Hai Duong, David Shriver, ThanhVu Nguyen, and Matthew B Dwyer. Compositional neural network verification via assume-guarantee reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Claudio Ferrari, Mark Niklas Mueller, Nikola Jovanović, and Martin Vechev. Complete Verification via Multi-Neuron Relaxation Guided Branch-and-Bound. In *International Conference on Learning Representations*, 2022. doi: 10.48550/arXiv.2205.00263.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *Computational linguistics*, 50(3):1097–1179, 2024.

Pei Huang, Dennis Wei, Omri Isac, Haoze Wu, Min Wu, and Clark Barrett. Parameterized abstract interpretation for transformer verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35500–35508, 2026.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4129–4142, 2019.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117. Springer, 2017. doi: 10.1007/978-3-319-63387-9_5.

Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *International Conference on Computer Aided Verification*, pp. 443–452. Springer, 2019. doi: 10.1007/978-3-030-25540-4_26.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: a calculus for reasoning about deep neural networks. *Formal Methods in System Design*, 60(1):87–116, 2022. doi: 10.1007/s10703-021-00363-7.

Konstantin Kaulen, Tobias Ladner, Stanley Bak, Christopher Brix, Hai Duong, Thomas Flinkow, Taylor T. Johnson, Lukas Koller, Edoardo Manino, ThanhVu H Nguyen, and Haoze Wu. The 6th international verification of neural networks competition (vnn-comp 2025): Summary and results, 2025. URL https://arxiv.org/abs/2512.19007.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. Robustness verification for transformers. *arXiv preprint arXiv:2002.06622*, 2020.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in Neural Information Processing Systems*, 31, 2018.

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):1–30, 2019.

Hoang-Dung Tran, Diago Manzanas Lopez, Patrick Musau, Xiaodong Yang, Luan Viet Nguyen, Weiming Xiang, and Taylor T Johnson. Star-based reachability analysis of deep neural networks. In *International symposium on formal methods*, pp. 670–686. Springer, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in {GPT} models. 2023.

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Formal security analysis of neural networks using symbolic intervals. In *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1599–1614, 2018. URL https://dl.acm.org/doi/10.5555/3277203.3277323.

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 *Advances in Neural Information Processing Systems*, volume 34, pp. 29909–29921, 2021.

Dennis Wei, Haoze Wu, Min Wu, Pin-Yu Chen, Clark Barrett, and Eitan Farchi. Convex bounds on the softmax function with applications to robustness verification. In *International Conference on Artificial Intelligence and Statistics*, pp. 6853–6878. PMLR, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Haoze Wu, Omri Isac, Aleksandar Zeljić, Teruhiro Tagomori, Matthew Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, et al. Marabou 2.0: a versatile formal analyzer of neural networks. In *International Conference on Computer Aided Verification*, pp. 249–264. Springer, 2024.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, volume 31, pp. 4939–4948, Montréal, Canada, 2018. Curran Associates, Inc.

Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. General cutting planes for bound-propagation-based neural network verification. *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022. URL https://dl.acm.org/doi/10.5555/3600270.3600391.

Yunruo Zhang, Lujia Shen, Shanqing Guo, and Shouling Ji. Galileo: General linear relaxation framework for tightening robustness certification of transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21797–21805, 2024.

Duo Zhou, Christopher Brix, Grani A Hanasusanto, and Huan Zhang. Scalable neural network verification with branch-and-bound inferred cutting planes. *arXiv preprint arXiv:2501.00200*, 2024.

Duo Zhou, Jorge Chavez, Hesun Chen, Grani A. Hanasusanto, and Huan Zhang. Clip-and-verify: Linear constraint-driven domain clipping for accelerating neural network verification. In *Advances in Neural Information Processing Systems*, 2025.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. Promptbench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254):1–22, 2024.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A GPT TRANSFORMATIONS

A.1 FUSED LAYERNORM TRANSFORM

Theorem 3 (LayerNorm soundness). *For $\Pi = I - \mathbf{1}\mathbf{1}^\top/d$ and $\tau > 0$, LayerNorm is*

$$\mathrm{LN}(x) = \gamma \odot \frac{\Pi x}{\sqrt{\tau + \|\Pi x\|_2^2/d}} + \beta. \tag{33}$$

For $Z = \mathcal{Z}(c, G, L, 0)$, let H_j range over its shared and embedded local generators. ZONOGPT *computes nonnegative vectors r_N and R_N such that every $x \in Z$ satisfies*

$$|J_{\mathrm{LN}}(c)(x-c)| \le r_N, \qquad |\,\mathrm{LN}(x) - \mathrm{LN}(c) - J_{\mathrm{LN}}(c)(x-c)| \le R_N. \tag{34}$$

The resulting zonotope

$$Z_N = \mathcal{Z}\Big(\mathrm{LN}(c), \{J_{\mathrm{LN}}(c)G_j\}_j, \{J_{\mathrm{LN}}(c_s)L_s\}_{s=1}^S, R_N\Big) \tag{35}$$

satisfies $\mathrm{LN}(Z) \subseteq Z_N$.

Proof. LayerNorm acts independently on each token, so the proof considers one token. For $x \in Z$, its change from the center and the corresponding centered change are

$$\Delta = x - c = \sum_j H_j \epsilon_j, \qquad \bar{\Delta} = \Pi\Delta = \sum_j \underbrace{\Pi H_j}_{\bar{H}_j} \epsilon_j. \tag{36}$$

The centered input and its squared norm at c are

$$z_0 = \Pi c, \qquad a_0 = \tau + \|z_0\|_2^2/d. \tag{37}$$

Consequently, LayerNorm at $x = c + \Delta$ is

$$\mathrm{LN}(c+\Delta) = \gamma \odot (z_0 + \bar{\Delta})\rho(a_0 + \delta a_{\mathrm{lin}} + \delta a_{\mathrm{quad}}) + \beta, \tag{38}$$

where

$$\rho(a) = a^{-1/2}, \qquad \delta a_{\mathrm{lin}} = 2z_0^\top \bar{\Delta}/d, \qquad \delta a_{\mathrm{quad}} = \|\bar{\Delta}\|_2^2/d. \tag{39}$$

Linear change computation. Differentiating Eq. 38 at $\Delta = 0$ gives

$$J_{\mathrm{LN}}(c)\Delta = \gamma \odot \left(\bar{\Delta}\rho(a_0) + z_0\rho'(a_0)\delta a_{\mathrm{lin}}\right). \tag{40}$$

Because the expression is linear in Δ, substituting Eq. 36 into Eq. 40 gives

$$J_{\mathrm{LN}}(c)\Delta = \sum_j J_{\mathrm{LN}}(c) H_j \epsilon_j. \tag{41}$$

Thus ZONOGPT computes

$$|J_{\mathrm{LN}}(c)\Delta| = |J_{\mathrm{LN}}(c)(x-c)| \le r_N, \qquad r_N = \sum_j |J_{\mathrm{LN}}(c)H_j|, \tag{42}$$

where the inequality follows from $|\epsilon_j| \le 1$ and the triangle inequality.

Nonlinear error computation. Subtracting $\mathrm{LN}(c)$ and $J_{\mathrm{LN}}(c)\Delta$ from Eq. 38 gives exactly

$$\mathrm{LN}(c+\Delta) - \mathrm{LN}(c) - J_{\mathrm{LN}}(c)\Delta = \gamma \odot \left(z_0 e_\rho + \bar{\Delta} d_\rho\right), \tag{43}$$

where

$$\begin{aligned} e_\rho &= \rho(a_0 + \delta a_{\mathrm{lin}} + \delta a_{\mathrm{quad}}) - \rho(a_0) - \rho'(a_0)\delta a_{\mathrm{lin}}, \\ d_\rho &= \rho(a_0 + \delta a_{\mathrm{lin}} + \delta a_{\mathrm{quad}}) - \rho(a_0). \end{aligned} \tag{44}$$

The input generators bound the quantities in Eq. 44 by

$$\begin{aligned} r_{\mathrm{cen}} &= \sum_j |\bar{H}_j|, & |\bar{\Delta}| &\le r_{\mathrm{cen}}, \\ r_{\mathrm{lin}} &= \sum_j |2z_0^\top \bar{H}_j/d|, & |\delta a_{\mathrm{lin}}| &\le r_{\mathrm{lin}}, \\ r_{\mathrm{quad}} &= \|r_{\mathrm{cen}}\|_2^2/d, & 0 \le \delta a_{\mathrm{quad}} &\le r_{\mathrm{quad}}, \\ r_a &= r_{\mathrm{lin}} + r_{\mathrm{quad}}, & |\delta a_{\mathrm{lin}} + \delta a_{\mathrm{quad}}| &\le r_a. \end{aligned} \tag{45}$$

Taylor's theorem also requires a positive lower bound for the argument of ρ. For $\widehat{z} = z_0/\|z_0\|_2$ when $z_0 \neq 0$ and $\widehat{z} = 0$ otherwise, the generator bounds give

$$\begin{aligned} r_{\|} &= \sum_j |\widehat{z}^\top \bar{H}_j|, \\ a_{\min} &= \tau + \max\left\{\max(0, \|z_0\|_2^2/d - r_{\mathrm{lin}}), (\|z_0\|_2 - r_{\|})_+^2/d\right\}. \end{aligned} \tag{46}$$

This construction ensures

$$\tau + \|z_0 + t\bar{\Delta}\|_2^2/d \geq a_{\min} \qquad \text{for every } t \in [0,1]. \tag{47}$$

Using $\rho'(a) = -\frac{1}{2}a^{-3/2}$ and $\rho''(a) = \frac{3}{4}a^{-5/2}$, Taylor's theorem and the mean value theorem yield

$$\begin{aligned} |e_\rho| &\leq \tfrac{1}{2}a_0^{-3/2} r_{\mathrm{quad}} + \tfrac{3}{8}a_{\min}^{-5/2} r_a^2, \\ |d_\rho| &\leq \tfrac{1}{2}a_{\min}^{-3/2} r_a. \end{aligned} \tag{48}$$

Applying these inequalities and $|\bar{\Delta}| \leq r_{\mathrm{cen}}$ to Eq. 43 gives

$$R_N = |\gamma| \odot \left[|z_0|\left(\tfrac{1}{2}a_0^{-3/2} r_{\mathrm{quad}} + \tfrac{3}{8}a_{\min}^{-5/2} r_a^2\right) + \tfrac{1}{2} r_{\mathrm{cen}} a_{\min}^{-3/2} r_a\right]. \tag{49}$$

Therefore, Eq. 42 gives the first inequality in Eq. 34, and Eq. 49 gives the second. For every $x \in Z = \mathcal{Z}(c, G, L, 0)$, Eq. 34 gives

$$\begin{aligned} \mathrm{LN}(x) = \mathrm{LN}(c) &+ \sum_j J_{\mathrm{LN}}(c) G_j \epsilon_j \\ &+ \sum_{s=1}^{S} \mathrm{emb}_s\left(J_{\mathrm{LN}}(c_s) L_s \epsilon_s^{\mathrm{loc}}\right) + e_N, \qquad |e_N| \leq R_N \end{aligned} \tag{50}$$

Therefore,

$$\mathrm{LN}(Z) \subseteq Z_N = \mathcal{Z}\Big(\mathrm{LN}(c), \{J_{\mathrm{LN}}(c) G_j\}_j, \{J_{\mathrm{LN}}(c_s) L_s\}_{s=1}^S, R_N\Big) \tag{51}$$

The Jacobian images retain every input symbol, while only the nonlinear error e_N enters the interval. □

Remark 1 (Lean mechanization of LayerNorm soundness)**.** *Thm. 3 is fully mechanized in Lean 4 as* `layerNorm_image_subset_lnZN` *in* `ZonoGpt/LayerNorm.lean`*, whose only hypothesis is* $\tau > 0$*. The mechanization follows the proof above. It first shows that the closed-form Jacobian* $J_{\mathrm{LN}}(c)$ *is the Fréchet derivative of LayerNorm (*`ZonoGpt/LayerNormDeriv.lean`*). For a single token, it then bounds the first-order change by* r_N *and the Taylor remainder by* R_N*, using second-order Taylor and mean-value bounds for* $a \mapsto a^{-1/2}$ *(*`ZonoGpt/Rho.lean`*,* `ZonoGpt/Calculus.lean`*); the hypothesis* $\tau > 0$ *serves only to guarantee* $a_{\min} > 0$*. Lifting these bounds to all tokens yields the symbol-preserving expansion Eq. 50 (*`layerNorm_szono_expansion`*), from which the inclusion* $\mathrm{LN}(Z) \subseteq Z_N$ *follows directly. The proofs proceed step by step, mainly using the tactics* `have`*,* `rw`*,* `simp`*, and* `ring`*, with* `linarith` *and* `positivity` *closing the inequalities. They contain no* `sorry` *and introduce no axioms.*

A.2 Affine GELU Transform

The MLP requires bounds on tanh-GELU

$$g(t) = \tfrac{1}{2}t\,(1 + \tanh h(t)), \qquad h(t) = \sqrt{2/\pi}\,(t + 0.044715t^3) \tag{52}$$

over each preactivation interval $[\ell, u]$. ZonoGpt bounds g on $[\ell, u]$ by a linear lower bound $at + r^{\mathrm{lb}}$ and a linear upper bound $at + r^{\mathrm{ub}}$ that share the coefficient a, where r^{lb} and r^{ub} bound the residual $q(t) = g(t) - at$ (Eq. 59). These bounds hold for every a; for $\ell < u$, ZonoGpt uses

$$a = \frac{g(u) - g(\ell)}{u - \ell}. \tag{53}$$

The second derivative of g satisfies

$$|g''(t)| \leq \mathrm{sech}^2(h(t))\left(|h'(t)| + |t|\left(|h'(t)|^2 + \tfrac{1}{2}|h''(t)|\right)\right), \tag{54}$$

where $h'(t) = \sqrt{2/\pi}\,(1 + 0.134145t^2)$ and $h''(t) = 0.26829\sqrt{2/\pi}\,t$. Partition $[\ell, u]$ into $\ell = t_0 < \cdots < t_K = u$, and let C_i bound the right-hand side of Eq. 54 on $[t_i, t_{i+1}]$. Linear interpolation of q on this interval gives

$$\begin{aligned} q_i^{\mathrm{lb}} &= \min\{q(t_i), q(t_{i+1})\} - \frac{C_i(t_{i+1} - t_i)^2}{8}, \\ q_i^{\mathrm{ub}} &= \max\{q(t_i), q(t_{i+1})\} + \frac{C_i(t_{i+1} - t_i)^2}{8} \end{aligned} \tag{55}$$

and therefore $q(t) \in [r_{\mathrm{part}}^{\mathrm{lb}}, r_{\mathrm{part}}^{\mathrm{ub}}]$, where

$$r_{\mathrm{part}}^{\mathrm{lb}} = \min_i q_i^{\mathrm{lb}}, \qquad r_{\mathrm{part}}^{\mathrm{ub}} = \max_i q_i^{\mathrm{ub}} \tag{56}$$

ZONOGPT also uses the global relation $g(t) = \mathrm{ReLU}(t) - e_{\mathrm{relu}}(t)$, where $e_{\mathrm{relu}}(t) = |t|/\big(1 + e^{2h(|t|)}\big)$ because h is odd, and

$$0 \le e_{\mathrm{relu}}(t) \le e_{\max}. \tag{57}$$

ZONOGPT computes $e_{\max} \approx 0.1703$ numerically. For $|t| \le 8$, it splits $[0, 8]$ into 8192 equal pieces and bounds e_{relu} on each piece $[t_-, t_+]$ by $t_+/\big(1 + e^{2h(t_-)}\big)$, since $|t|$ increases and $1/\big(1 + e^{2h(|t|)}\big)$ decreases in $|t|$. For $|t| \ge 8$, it uses $e_{\mathrm{relu}}(t) \le 8e^{-2h(8)}$, since $|t|e^{-2h(|t|)}$ decreases there. This computation runs in floating point with a small outward margin rather than directed rounding. A coarser closed-form value follows from $h(|t|) \ge \sqrt{2/\pi}\,|t|$: $e_{\mathrm{relu}}(t) \le |t|e^{-2\sqrt{2/\pi}\,|t|} \le \sqrt{\pi/2}/(2e) \approx 0.231$. Because $\mathrm{ReLU}(t) - at$ is piecewise affine, its extrema on $[\ell, u]$ occur at ℓ, u, or zero when the interval contains zero. For this set of points E, define

$$r_{\mathrm{global}}^{\mathrm{lb}} = \min_{t \in E}\big(\mathrm{ReLU}(t) - at\big) - e_{\max}, \qquad r_{\mathrm{global}}^{\mathrm{ub}} = \max_{t \in E}\big(\mathrm{ReLU}(t) - at\big) \tag{58}$$

Intersecting the two sound bounds gives

$$r^{\mathrm{lb}} = \max\{r_{\mathrm{part}}^{\mathrm{lb}}, r_{\mathrm{global}}^{\mathrm{lb}}\}, \qquad r^{\mathrm{ub}} = \min\{r_{\mathrm{part}}^{\mathrm{ub}}, r_{\mathrm{global}}^{\mathrm{ub}}\} \tag{59}$$

The implementation uses $K = 32$ equal subintervals.

Set $m = (r^{\mathrm{lb}} + r^{\mathrm{ub}})/2$ and $d = (r^{\mathrm{ub}} - r^{\mathrm{lb}})/2$ elementwise. Then every $t \in [\ell, u]$ satisfies

$$g(t) = at + m + d\epsilon^g, \qquad |\epsilon^g| \le 1 \tag{60}$$

For a preactivation zonotope $Z_z = \mathcal{Z}(c_z, G_z, L_z, b_z)$, applying Eq. 60 elementwise produces

$$\begin{aligned} Z_g &= \mathcal{Z}(c_g, G_g, L_g, b_g), \qquad & c_g &= a \odot c_z + m, \quad (G_g)_j = a \odot (G_z)_j, \\ (L_g)_s &= [\,a \odot (L_z)_s \mid \mathrm{diag}(d_s)\,], \quad & (b_g)_s &= |a_s| \odot (b_z)_s \end{aligned} \tag{61}$$

The new columns $\mathrm{diag}(d_s)$ are local because GELU acts independently on every token and coordinate.

Theorem 4 (Affine GELU soundness)**.** *Let $Z_z = \mathcal{Z}(c_z, G_z, L_z, b_z)$ be a GELU input zonotope, and let $[\ell, u]$ contain every coordinate of Z_z. The bounds in Eq. 59 and the output in Eq. 61 satisfy*

$$\{g(z) : z \in Z_z\} \subseteq Z_g. \tag{62}$$

Moreover, the output keeps the input noise symbols: every $z = c_z + \sum_{j=1}^{m}(G_z)_j\epsilon_j + \sum_{s=1}^{S}\mathrm{emb}_s\big((L_z)_s\epsilon_s^{\mathrm{loc}}\big) + e \in Z_z$ satisfies $g(z) = c_g + \sum_{j=1}^{m}(G_g)_j\epsilon_j + \sum_{s=1}^{S}\mathrm{emb}_s\big((L_g)_s(\epsilon_s^{\mathrm{loc}}, \epsilon_s^g)\big) + a \odot e$ for some $\|\epsilon^g\|_\infty \le 1$.

Proof of Thm. 4. On each partition interval, the linear-interpolation error is at most $C_i(t_{i+1} - t_i)^2/8$. The interpolant lies between its endpoint values, which proves Eq. 55. The identity $g = \mathrm{ReLU} - e_{\mathrm{relu}}$ with Eq. 57 proves Eq. 58. Their intersection contains $q(t)$, so centering $[r^{\mathrm{lb}}, r^{\mathrm{ub}}]$ gives Eq. 60. Applying this scalar identity to every coordinate of z gives the symbol-preserving form with the fields in Eq. 61, where $|a \odot e| \le |a| \odot b_z = b_g$. Therefore, every $g(z)$ belongs to Z_g, which proves Eq. 62. □

Remark 2 (Lean mechanization of affine GELU soundness)**.** *Thm. 4 is mechanized as* `gelu_soundness` *in* `ZonoGpt/GeluSound.lean`*. The mechanized bound holds for any $K \ge 1$ partition nodes that cover $[\ell, u]$; the implementation's $K = 32$ equal subintervals and the*

coefficient a in (53) *serve only to tighten the linear lower and upper bounds. The symbol-preserving statement of Thm. 4 is proved as* `gelu_szono_symbolic`*, which skip connections require because they share noise symbols with earlier generators (*`set_inclusion_not_compositional`*). The mechanization follows the GELU bound of §A.2 step by step. It first verifies the formulas for g' and g'', the curvature bound* (54)*, and the closed-form ReLU gap bound $e_{\max} = \sqrt{\pi/2}/(2e)$ (*`ZonoGpt/GeluCalc.lean`*). On each subinterval, it bounds the residual $q(t) = g(t) - at$ by the linear interpolation error in* (55)*; combining these bounds with the global ReLU bound gives the residual interval $[r^{\mathrm{lb}}, r^{\mathrm{ub}}]$, and hence an affine bound that keeps the noise symbols of the input. The proofs proceed step by step, mainly using the tactics* `have`*,* `rw`*,* `exact`*, and* `unfold`*, with* `linarith` *and* `positivity` *closing the inequalities and* `split_ifs` *handling the case analysis in the* min *and* max *bounds.*

A.3 Fused MLP Transform

For input x, the MLP block computes $M(x) = x + W_2 g(W_1 \,\mathrm{LN}(x) + b_1) + b_2$. To construct output bounds for M, ZonoGpt computes the three terms of Eq. 3 for the complete MLP residual: $M(c)$, $J_M(c)H_j$, and R_M. First, ZonoGpt computes $M(c)$ by evaluating the MLP residual at $x = c$. Second, ZonoGpt computes $J_M(c)H_j$ along the path $x(t) = c + tH_j$ from Eq. 15:

$$\begin{aligned} &n = \mathrm{LN}(c), \qquad z_c = W_1 n + b_1, \qquad \dot{n} = J_{\mathrm{LN}}(c)H_j, \qquad \dot{z} = W_1 \dot{n} \\ &J_M(c)H_j = H_j + W_2\big(g'(z_c) \odot \dot{z}\big) = H_j + J_B(c)H_j \end{aligned} \tag{63}$$

Third, ZonoGpt computes the nonlinear-error bound R_M by Thm. 5.

Theorem 5 (MLP nonlinear-error bound)**.** *Let $M(x) = x + W_2 g(W_1 \,\mathrm{LN}(x) + b_1) + b_2$ act tokenwise, let $Z = \{c + \sum_j H_j \epsilon_j : |\epsilon_j| \le 1\}$, and set $z_c = W_1 \,\mathrm{LN}(c) + b_1$. Suppose a nonnegative tuple (r_N, R_N, C_g) satisfies $|J_{\mathrm{LN}}(c)(x - c)| \le r_N$ and $|\,\mathrm{LN}(x) - \mathrm{LN}(c) - J_{\mathrm{LN}}(c)(x - c)| \le R_N$ for every $x \in Z$. Suppose $|g''(u)| \le C_g$ elementwise for every u between $z_c - |W_1|(r_N + R_N)$ and $z_c + |W_1|(r_N + R_N)$. Then, for every $x \in Z$,*

$$\underbrace{|M(x) - M(c) - J_M(c)(x - c)|}_{e_M} \le \underbrace{|W_2 \,\mathrm{diag}(g'(z_c))W_1| R_N + \tfrac{1}{2}|W_2| \left(C_g \odot \left(|W_1|(r_N + R_N)\right)^2\right)}_{R_M} \tag{64}$$

Proof sketch. Taylor's theorem splits the omitted error into a transported LayerNorm error and a GELU curvature term. §A.3.1 gives the computation and full proof. □

Finally, ZonoGpt computes the MLP output zonotope $Z_M = \mathcal{Z}(c_M, G_M, L_M, b_M)$ (Def. 1):

$$c_M = M(c), \quad (G_M)_j = J_M(c)G_j, \quad (L_M)_s = [\,(J_M(c)\,\mathrm{emb}_s(L_{s,k}))_s\,]_{k=1}^{q_s}, \quad (b_M)_s = (R_M)_s \tag{65}$$

Thm. 6 establishes soundness of the output zonotope in Eq. 65.

Theorem 6 (Fused MLP soundness)**.** *Let $Z = \mathcal{Z}(c, G, L, 0)$, with $L_s = [L_{s,1}, \ldots, L_{s,q_s}]$, and let M be the tokenwise MLP residual above. Suppose a nonnegative tuple (r_N, R_N, C_g) satisfies the hypotheses of Thm. 5 for every $x \in Z$. Then Z_M defined by Eq. 65 satisfies $M(Z) \subseteq Z_M$.*

Proof. The Jacobian images in Eq. 63 retain the input symbols, and the tokenwise MLP keeps each local image at its source token. The nonlinear error satisfies $|e_M| \le R_M = b_M$ by Thm. 5, so the four fields in Eq. 65 contain every output $M(x)$ for $x \in Z$. □

Remark 3 (Lean mechanization of fused MLP soundness)**.** *Thm. 6 is mechanized as* `MlpParams.fused_mlp_sound` *in* `ZonoGpt/Mlp.lean`*, and the variant with LayerNorm pre-composed as* `MlpParams.fused_mlp_sound_ln`*. The mechanized theorem assumes only the three bounds of Thm. 5 on r_N, R_N, and C_g. The mechanization also proves end-to-end soundness of the sequential MLP, covering the full transform used by the implementation (*`MlpParams.Mseq_image_subset_seqMlpOut` *in* `ZonoGpt/Sequential.lean`*). The proof combines two facts. First, the JVP $J_M(c)$ is linear and acts on each token separately, so each input generator maps to exactly one output generator with the same noise symbol, and each local generator stays at its own token. Second, Thm. 5 bounds the remaining error by $R_M = b_M$. Together, they give the symbol-preserving expansion of $M(x)$*

(`MlpParams.fused_mlp_symbolic`), from which $M(Z) \subseteq Z_M$ follows. Instantiating r_N and R_N with the LayerNorm radii of Thm. 3, and C_g with $\sup|g''|$, leaves $\tau > 0$ as the only hypothesis (`MlpParams.fused_mlp_sound_ln_sup`). The proofs mainly use the tactics `have`, `rw`, `simp only`, and `exact`, with `linarith` closing the inequalities.

A.3.1 MLP NONLINEAR-ERROR COMPUTATION

Proof of Thm. 5. Fix $x \in Z$ and set $v = x - c$. Set $z_c = W_1\,\mathrm{LN}(c) + b_1$. The LayerNorm construction gives the following bounds (Thm. 3):

$$\mathrm{LN}(x) = \mathrm{LN}(c) + J_{\mathrm{LN}}(c)v + e_N, \qquad |J_{\mathrm{LN}}(c)v| \le r_N, \qquad |e_N| \le R_N$$

The change in the GELU preactivation is

$$\begin{aligned} z_{\mathrm{change}} &= W_1\big(\mathrm{LN}(x) - \mathrm{LN}(c)\big) \\ &= \underbrace{W_1 J_{\mathrm{LN}}(c)v}_{\text{linear LayerNorm change}} + \underbrace{W_1 e_N}_{\text{LayerNorm nonlinear error}} \end{aligned}$$

Therefore,

$$|z_{\mathrm{change}}| \le |W_1| r_N + |W_1| R_N = |W_1|(r_N + R_N)$$

For tanh-GELU, set

$$s(u) = \sqrt{2/\pi}\,(u + 0.044715u^3), \qquad g(u) = \tfrac{1}{2}u\big(1 + \tanh s(u)\big)$$

The derivatives of the inner function are

$$s'(u) = \sqrt{2/\pi}\,(1 + 0.134145u^2), \qquad s''(u) = 0.26829\sqrt{2/\pi}\,u$$

Differentiating GELU once gives

$$g'(u) = \tfrac{1}{2}\big(1 + \tanh s(u)\big) + \tfrac{1}{2}u\,\mathrm{sech}^2(s(u))s'(u)$$

Differentiating again gives

$$g''(u) = \mathrm{sech}^2(s(u))\left[s'(u) + \tfrac{1}{2}us''(u) - u\tanh(s(u))(s'(u))^2\right]$$

Since $|\tanh(s(u))| \le 1$,

$$|g''(u)| \le \mathrm{sech}^2(s(u))\left(|s'(u)| + |u|\left(|s'(u)|^2 + \tfrac{1}{2}|s''(u)|\right)\right)$$

The preactivation bound places every coordinate of $z_c + z_{\mathrm{change}}$ in

$$[\,z_c - |W_1|(r_N + R_N),\ z_c + |W_1|(r_N + R_N)\,]$$

Bounding the preceding expression over each coordinate interval gives

$$C_g = \sup_{u \in [\,z_c - |W_1|(r_N+R_N),\, z_c + |W_1|(r_N+R_N)\,]} |g''(u)|$$

Apply Taylor's theorem to GELU at z_c:

$$g(z_c + z_{\mathrm{change}}) = g(z_c) + g'(z_c) \odot z_{\mathrm{change}} + e_g$$

The curvature assumption on C_g gives

$$|e_g| \le \tfrac{1}{2}C_g \odot |z_{\mathrm{change}}|^2 \le \tfrac{1}{2}C_g \odot \big(|W_1|(r_N + R_N)\big)^2$$

Expanding the complete MLP residual gives

$$\begin{aligned} M(x) - M(c) &= v + W_2\big(g(z_c + z_{\mathrm{change}}) - g(z_c)\big) \\ &= v + W_2\,\mathrm{diag}(g'(z_c))W_1 J_{\mathrm{LN}}(c)v \\ &\quad + W_2\,\mathrm{diag}(g'(z_c))W_1 e_N + W_2 e_g \end{aligned}$$

The first two terms form $J_M(c)v$. Consequently,

$$M(x) - M(c) - J_M(c)v = W_2\,\mathrm{diag}(g'(z_c))W_1 e_N + W_2 e_g$$

Taking magnitudes after composing the linear factors yields

$$\begin{aligned}|M(x) - M(c) - J_M(c)v| &\le |W_2 \operatorname{diag}(g'(z_c))W_1|R_N + |W_2||e_g| \\ &\le |W_2 \operatorname{diag}(g'(z_c))W_1|R_N \\ &\quad + \tfrac{1}{2}|W_2|\left(C_g \odot \left(|W_1|(r_N + R_N)\right)^2\right)\end{aligned}$$

The right-hand side is R_M in Eq. 64. □

Remark 4 (Lean mechanization of the MLP nonlinear-error bound)**.** *The computation above is mechanized as* `MlpParams.mlp_remainder` *in* `ZonoGpt/Mlp.lean` *using a second-order Taylor bound (*`taylor2_bound` *in* `ZonoGpt/Calculus.lean`*). The curvature bound C_g needs to hold only on the interval centered at z_c with radius $|W_1|(r_N + R_N)$. The mechanization defines a computable elementwise curvature bound,* `MlpParams.CgSup`*, and proves it valid using the global curvature estimate $|g''(u)| \le C_{\sup}$ from* `ZonoGpt/GeluCalc.lean`*. The mechanization follows the computation step by step: it writes the preactivation change as W_1 applied to the first-order LayerNorm change plus the LayerNorm remainder, bounds this change elementwise by $|W_1|(r_N + R_N)$, applies the second-order Taylor bound to g on the segment between z_c and the perturbed preactivation, and combines the two error terms into R_M. This Taylor bound requires only that g and g' be differentiable. The bound holds for every τ and for r_N, R_N, and C_g of any sign.*

A.4 Fused Attention Transform

Proof of Thm. 1. Fix $x \in Z$ with $\Delta = x - c$. For each intermediate quantity X, its total change, linear change, and nonlinear error are

$$\delta X = X(x) - X(c), \qquad \dot{X} = J_X(c)\Delta, \qquad e_X = \delta X - \dot{X} \tag{66}$$

Thus $\dot{X}$ is the linear change retained by the Jacobian, and e_X is the nonlinear error to be bounded.

First, ZonoGpt computes the LayerNorm linear-change and nonlinear-error bounds

$$|\dot{N}| \le r_N, \qquad |e_N| \le R_N \tag{67}$$

using the construction in §A.1. The affine QKV projection then gives (f_Q, f_K, f_V) from $|W_{qkv}|r_N$ and (R_Q, R_K, R_V) from $|W_{qkv}|R_N$. Hence

$$|\delta Q| \le T_Q = f_Q + R_Q, \qquad |\delta K| \le T_K = f_K + R_K, \qquad |\delta V| \le T_V = f_V + R_V \tag{68}$$

Next, expand the score matrix $U = QK^\top/\sqrt{d_h}$. Its linear change and nonlinear error are

$$\begin{aligned}\dot{U} &= (\dot{Q}K^\top + Q\dot{K}^\top)/\sqrt{d_h} \\ e_U &= (e_Q K^\top + Qe_K^\top + \delta Q \delta K^\top)/\sqrt{d_h}\end{aligned} \tag{69}$$

The three required bounds follow directly:

$$\begin{aligned}f_U &= (f_Q|K|^\top + |Q|f_K^\top)/\sqrt{d_h} \\ R_U^N &= (R_Q|K|^\top + |Q|R_K^\top)/\sqrt{d_h} \\ R_U^\times &= T_Q T_K^\top/\sqrt{d_h}\end{aligned} \tag{70}$$

Therefore $|\dot{U}| \le f_U$, $|e_U| \le R_U^N + R_U^\times$, and $|\delta U| \le f_U + R_U^N + R_U^\times$.

The score bounds must now pass through softmax. For every nonnegative array r, define

$$[\mathcal{J}_P(r)]_{ij} = P_{ij}\left(r_{ij} + \sum_k P_{ik} r_{ik}\right) \tag{71}$$

so $|J_{\mathrm{SM}}(U)\Delta U| \le \mathcal{J}_P(|\Delta U|)$. In particular, $f_P = \mathcal{J}_P(f_U)$ bounds $|\dot{P}|$.

To bound the nonlinear softmax error, consider one row $p(t) = \mathrm{softmax}(U + t\delta U)$ and set $M = \|f_U + R_U^N + R_U^\times\|_\infty$. Direct differentiation gives

$$p_j''(t) = p_j(t)\left[(\delta U_j - \mathbb{E}_{p(t)}\delta U)^2 - \mathrm{Var}_{p(t)}(\delta U)\right] \tag{72}$$

The bracket has magnitude at most $4M^2$, and $p_j(t) \le P_j e^{2M}$. Taylor's integral remainder therefore gives

$$|\operatorname{softmax}(U+\delta U) - P - J_{\mathrm{SM}}(U)\delta U| \le 2P \odot e^{2M} M^2 \tag{73}$$

The retained derivative is $J_{\mathrm{SM}}(U)\dot{U}$. The omitted score error $e_U = \delta U - \dot{U}$ contributes at most $\mathcal{J}_P(R_U^N) + \mathcal{J}_P(R_U^\times)$. Thus the Taylor bound is

$$R_P^T = \mathcal{J}_P(R_U^N) + \mathcal{J}_P(R_U^\times) + 2P \odot e^{2M} M^2 \tag{74}$$

Softmax outputs lie in $[0,1]$, which independently gives $R_P^I = \max\{P, 1-P\} + f_P$. Consequently,

$$R_P = \min\{R_P^T, R_P^I\}, \qquad T_P = f_P + R_P \tag{75}$$

satisfy $|e_P| \le R_P$ and $|\delta P| \le T_P$.

Finally, expand the attention-value product after subtracting its center and linear change:

$$\begin{aligned} e_A &= A(x) - A(c) - J_A(c)\Delta \\ &= W_o \operatorname{concat}\big(\underbrace{e_P V}_{\text{softmax error}} + \underbrace{P e_V}_{\text{value error}} + \underbrace{\delta P \delta V}_{\text{product of both changes}} \big) \end{aligned} \tag{76}$$

The three terms are bounded by

$$|e_P V| \le R_P |V|, \qquad |P e_V| \le P R_V, \qquad |\delta P \delta V| \le (f_P + R_P)(f_V + R_V) \tag{77}$$

because $P \ge 0$ and $T_V = f_V + R_V$. Concatenating the heads and applying $|W_o|$ gives exactly the bound R_A in Eq. 18. The residual skip and output bias introduce no nonlinear error because both are affine. □

Remark 5 (Lean mechanization of the attention nonlinear-error bound)**.** *Thm. 1 is mechanized as* `AttnParams.attention_radius_recipe` *in* `ZonoGpt/AttentionRecipe.lean`*, whose only hypothesis is* $\tau > 0$*. The quantity* M *in* (73) *must be taken per row of the score matrix (*`elementwise_M_unsound` *in* `ZonoGpt/SoftmaxTaylor.lean`*). The mechanization follows the computation above step by step: it propagates the LayerNorm radii of Thm. 3 through the projections via* $|W|$*, through the scores via the exact expansion of* $QK^\top$*, and through softmax via the radius map* $\mathcal{J}_P$ *and a row-wise Taylor bound based on the second derivative* (72)*. Each step is an exact identity followed by the triangle inequality; the only analytic inputs are the LayerNorm bounds, which need* $\tau > 0$*, and the softmax Taylor bound. The computed tuple* (f_P, R_P, f_V, R_V) *is also proved nonnegative, so it satisfies the nonnegativity condition of Thm. 1. The proofs mainly use the tactics* `have`*,* `rw`*,* `exact`*,* `simp only`*, and* `ring`*, with* `linarith` *and* `Finset.sum_le_sum` *closing the inequalities.*

A.5 Fused attention soundness

Proof of Thm. 2. Fix $x \in Z$. At the center, $P = P(c)$ and $V = V(c)$. For $x \in Z$,

$$P(x) = P + \delta P, \qquad V(x) = V + \delta V$$

The four assumptions give

$$\delta P = J_P(c)(x-c) + e_P, \qquad \delta V = J_V(c)(x-c) + e_V, \qquad |e_P| \le R_P, \qquad |e_V| \le R_V \tag{78}$$

Expanding $(P+\delta P)(V+\delta V)$ and subtracting its center and Jacobian terms gives

$$A(x) - A(c) - J_A(c)(x-c) = W_o \operatorname{concat}\big(e_P V + P e_V + \delta P \delta V\big) \tag{79}$$

Consider the first term at query token i and head h. Each row of e_P sums to zero, so Eq. 19 gives

$$\sum_j (e_P)_{ihj} \bar{v}_{hj} = \sum_j (e_P)_{ihj} (\bar{v}_{hj} - a_{ih}) \tag{80}$$

For every i, h, j, the bound $|(e_P)_{ihj}| \le (R_P)_{ihj}$ expresses $(e_P)_{ihj}$ as a coefficient in $[-1,1]$ multiplied by $(R_P)_{ihj}$. Hence the generators $(R_P)_{ihj}(\bar{v}_{hj} - a_{ih})$ in $(L_A)_i$ contain the complete contribution of $W_o \operatorname{concat}(e_P V)$ at token i.

The remaining two terms satisfy

$$|P e_V + \delta P \delta V| \le P R_V + (f_P + R_P)(f_V + R_V) \tag{81}$$

because $P \geq 0$, $|\delta P| \leq f_P + R_P$, and $|\delta V| \leq f_V + R_V$. Concatenating the heads and applying W_o bounds their output by

$$|W_o| \operatorname{concat}\big(PR_V + (f_P + R_P)(f_V + R_V)\big) \tag{82}$$

which is the nonlinear-error term in b_A.

It remains to place the linear term $J_A(c)(x - c)$ in the output zonotope. The definition of Z gives

$$J_A(c)(x-c) = \sum_j J_A(c) G_j \epsilon_j + \sum_{s=1}^{S} \sum_{k=1}^{q_s} J_A(c) \operatorname{emb}_s(L_{s,k}) \epsilon_{s,k}^{\mathrm{loc}} \tag{83}$$

The first sum is represented exactly by $(G_A)_j = J_A(c)G_j$ with the original coefficients ϵ_j. For a local input generator at token s, $(L_A)_s$ retains the row at token s. At every token $i \neq s$, the triangle inequality bounds the same generator image by $|(J_A(c) \operatorname{emb}_s(L_{s,k}))_i|$. Summing these bounds over $s \neq i$ and k gives the cross-token term in $(b_A)_i$.

Therefore $c_A = A(c)$ represents the center, G_A and L_A represent the retained terms, and b_A bounds all remaining terms. Thus every $A(x)$ belongs to Z_A, which proves $A(Z) \subseteq Z_A$. □

Remark 6 (Lean mechanization of fused attention soundness)**.** *Thm. 2 is mechanized as* `AttnParams.fused_attention_sound_recipe` *in* `ZonoGpt/AttentionRecipe.lean`*, whose only hypothesis is* $\tau > 0$*. The argument for* (19)*, which builds on the exact expansion* `AttnParams.A_sub_sub_dA`*, uses the zero-sum property* $\sum_j (e_P)_{ihj} = 0$ *to replace* $\bar{v}_{hj}$ *by* $\bar{v}_{hj} - a_{ih}$ *for any* a_{ih}*, and writes each* $(e_P)_{ihj}$ *as* $\xi_{ihj}(R_P)_{ihj}$ *with* $|\xi_{ihj}| \leq 1$*. The theorem requires* $b = 0$ *for the input zonotope, because the linear expansion covers only the generator part of the input (*`fused_attention_needs_zero_interval`*). The proof separates the algebra from the bounds. It first writes the nonlinear error as the algebraic identity* $W_o \operatorname{concat}(e_P V + P e_V + \delta P\, \delta V)$*; the zero-sum property then places the softmax term* $e_P V$ *into local generators, and* b_A *bounds the remaining terms. This yields the symbol-preserving form (*`AttnParams.fused_attention_symbolic_recipe`*), from which* $A(Z) \subseteq Z_A$ *follows for every choice of* a_{ih}*; the weighted median serves only to tighten the bound. The proofs mainly use the tactics* `simp only`*,* `have`*,* `rw`*,* `exact`*, and* `refine`*, with* `funext` *for equalities between sequences.*

B Related Work

DNN verification tools. State-of-the-art DNN verifiers, such as those evaluated in VNN-COMP (Kaulen et al., 2025), combine techniques to improve scalability. $\alpha\beta$-CROWN (Zhou et al., 2024; 2025; Zhang et al., 2022; Wang et al., 2021) and NeuralSat (Duong et al., 2024; 2025a) split verification problems and refine bounds on the resulting subproblems. Marabou (Wu et al., 2024; Katz et al., 2017; 2022; 2019) encodes each NNV instance as a constraint-solving problem and solves with a customized solver, while NNEnum (Bak, 2021) enumerates reachable regions using star sets (Tran et al., 2019). Note that every NNV tool uses some forms of abstract domains to quickly compute bounds of DNNs given input regions and thus verify problems efficiently.

Abstraction domains. Existing abstractions form a spectrum of speed-accuracy tradeoffs, IBP (Wang et al., 2018) propagates interval bounds layer by layer, making it the fastest method but also the loosest. DeepZ (Singh et al., 2018) refines this by tracking affine error terms through linear layers, yielding tighter bounds at the cost of higher computational cost. CROWN (Zhang et al., 2018) and DeepPoly (Singh et al., 2019) apply linear relaxation with full back-substitution to the inputs, producing tighter bounds in a single forward-backward pass. α-CROWN (Xu et al., 2020) extends CROWN by optimizing the slopes of the linear relaxations via gradient ascent, while GCP-CROWN (Zhang et al., 2022) augments α-CROWN with cutting planes on intermediate neurons, pruning infeasible regions, further tightening bounds at the cost of very high computation. However, none of these methods scale to very large and deep networks with millions of parameters, *e.g.*, GPT-2 Medium (Radford et al., 2019).

Transformer verification. Existing methods adapt abstract domains to smaller Transformer models or tighten bounds for individual operations. Shi et al. (2020) verifies small models with simplified LayerNorm. DeepT extends zonotopes to Transformer operations, but evaluates deeper models

with smaller configurations and restricted perturbations (Bonaert et al., 2021). PBVERIFIER (Huang et al., 2026) and Zhang et al. (2024) improve attention-product bounds, while Wei et al. (2023) improves softmax bounds. COVENN (Duong et al., 2025b) scales through decomposition, whose interval bounds between components lose precision quickly.

Our approach. Prior work remains three limitations: (i) DEEPZ accumulates generators with network size; (ii) existing work bounds softmax and attention products separately, while COVENN's interval bounds lose relations between components; and (iii) Shi et al. (2020) uses simplified LayerNorm, while DEEPT evaluates smaller configurations with restricted perturbations. ZONOGPT addresses these limitations by (i) reducing shared and local generators after each block to keep space independent of network depth (§4.1, §4.4, and §4.5); (ii) retaining relations between components in its zonotope and bounding the attention residual as one fused transform (§4.1 and §4.2); and (iii) bounding standard LayerNorm and GELU to verify GPT-2 architectures (§4.3.1, §4.3.2, and §5).

C BENCHMARK DETAILS

Tab. 3: Benchmark configuration.

	MNIST	**SST**
Task	10-class image classification	Binary sentiment classification
Input sequence	28 projected image rows	GPT-2 tokens, length 32 or 48
Classifier input	Mean-pooled tokens	Final token
Training	Cross-entropy	IBP, from scratch
Perturbation	ℓ_∞ over all pixels	ℓ_∞ over 1 or 3 token embeddings
Perturbation dimension	$28 \times 28 = 784$	768 or 2304 (Small), 1024 or 3072 (Medium)
Radii ε	$10^{-3},\ 5{\times}10^{-4},\ 10^{-4},\ 5{\times}10^{-5},\ 10^{-5}$	
Instances	$8 \times 5 \times 4 \times 9 = 1440$	$8 \times 5 \times 18 \times 2 = 1440$

Both tasks use causal GPT-2 blocks at Small (width 768, 12 heads) and Medium (width 1024, 16 heads) sizes. The block counts are 2, 4, 8, and 12 for Small, and 2, 4, 8, and 24 for Medium. For MNIST, a learned projection maps each image row to a token. We select four correctly classified test images per model and radius, clip pixel perturbations to $[0, 1]$, and write one specification per alternative class. For SST, we perturb token embeddings rather than token IDs. We select 18 correctly classified validation sentences per model and radius. For each sentence, we generate two specifications, perturbing either one or three token embeddings. All sampling uses seed 0.

In all experiments, ZONOGPT bounds the combined generator pool by 512 MiB (2^{29} bytes). The shared and local tensors store mSd and qSd float32 values, respectively, and therefore require $4Sd(m + q)$ bytes. The storage limit gives $m + q \leq \lfloor 2^{29}/(4Sd) \rfloor$. The implementation retains at least 64 generators, giving $m + q \leq \max\{64, \lfloor 2^{29}/(4Sd) \rfloor\}$.

Let $C = \max\{64, \lfloor 2^{29}/(4Sd) \rfloor\}$ denote this combined limit, and let $\hat{m}$ and $\hat{q}$ denote the numbers of shared and local generators before each reduction. The values of C are 6,241 and 4,681 for MNIST Small and Medium, and 5,461 and 4,096 for SST Small and Medium, respectively. ZONOGPT computes m and q dynamically during propagation, while keeping $m + q \leq C$ after each reduction. In particular, ZONOGPT retains $m = \min\{\hat{m}, C\}$ shared generators and $q = \min\{\hat{q}, C - m\}$ local generators per token.