
Tropical Attention: Neural Algorithmic Reasoning for Combinatorial Algorithms

Baran Hashemi
Origins Data Science Lab
Technical University of Munich
Munich, Germany
baran.hashemi@tum.de

Kurt Pasque
Naval Postgraduate School
Monterey, California
kurt.pasque@nps.edu

Chris Teska
Naval Postgraduate School
Monterey, California
christopher.teska@nps.edu

Ruriko Yoshida
Naval Postgraduate School
Monterey, California
ryoshida@nps.edu

Abstract

Can algebraic geometry enhance the sharpness, robustness, and interpretability of modern neural reasoning models by equipping them with a mathematically grounded inductive bias? To answer this, we introduce Tropical Attention, an attention mechanism grounded in tropical geometry that lifts the attention kernel into tropical projective space, where reasoning is piecewise-linear and 1-Lipschitz, thus preserving the polyhedral decision structure inherent to combinatorial reasoning. We prove that Multi-Head Tropical Attention (MHTA) stacks universally approximate tropical circuits and realize tropical transitive closure through composition, achieving polynomial resource bounds without invoking recurrent mechanisms. These guarantees explain why the induced polyhedral decision boundaries remain sharp and scale-invariant, rather than smoothed by Softmax. Empirically, we show that Tropical Attention delivers stronger out-of-distribution generalization in both length and value, with high robustness against perturbative noise, and substantially faster inference with fewer parameters compared to Softmax-based and recurrent attention baselines, respectively. For the first time, we push the domain of neural algorithmic reasoning beyond **P**TIME problems to **NP-hard/complete** problems, paving the way toward sharper and more expressive Large Reasoning Models (LRMs) capable of tackling complex combinatorial challenges in Phylogenetics, Cryptography, Particle Physics, and Mathematical Discovery. The code is available at <https://github.com/Baran-phys/Tropical-Attention/>.

1 Introduction

The *tropical semiring* $\mathbb{T} := (\mathbb{R} \cup \{-\infty\}, \max, +)$ (or its “min-plus” variant) replaces ordinary addition by maximum and multiplication by addition [1]. Polynomials over this semiring evaluate to *piecewise-linear*, polyhedral functions. These are the main objects of study in *tropical geometry* that translates algebraic geometry into combinatorics, turning varieties into polyhedral complexes, with wide applications across the intersection of matroid theory, combinatorial optimization, auction theory, enumerative geometry [1–5], and recently Machine Learning [6–13]. Because it analyses the entire polyhedral structure of solutions rather than a single Euclidean point, tropical geometry is a

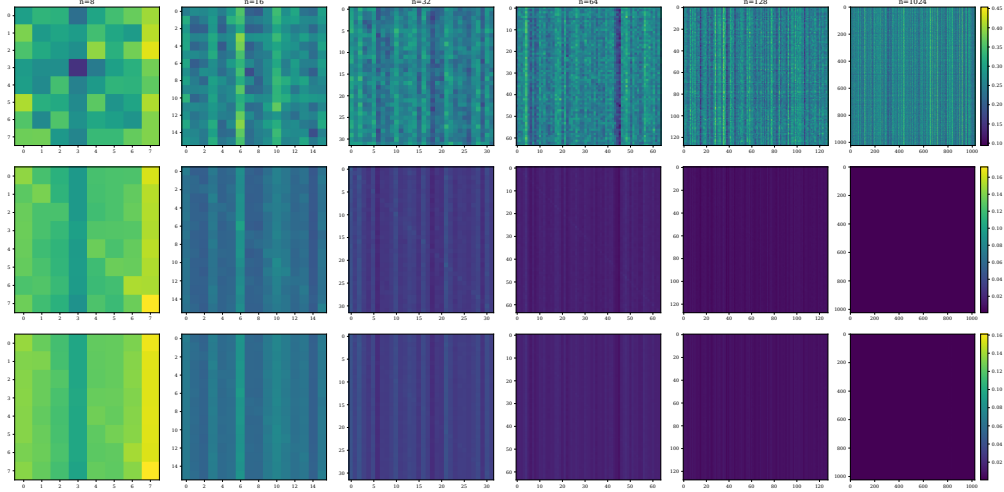


Figure 1: (top) **Tropical Attention** with sharp attention maps on learning the notorious¹ **QUICKSELECT** algorithm, showcasing a size-invariance and OOD lengths generalization behavior far beyond training ($8 \rightarrow 1024$). In contrast, both (middle) adaptive-softmax and (bottom) vanilla-softmax heads dilute and uniformly disperse as sequence length grows, failing to generalize. Each column evaluates the models on a new batch of independently drawn inputs of increasing length. Since the position of the target k -th element is different in each batch, the pattern of attention naturally changes to reflect the new data.

Combinatorial Tasks	Baseline Transformers		Universal Transformers (UT)		Tropical Transformer
	<i>Vanilla</i>	<i>Adaptive</i>	<i>Vanilla w/ ACT</i>	<i>Adaptive w/ ACT</i>	
CONVEXHULL	42.75 ± 2.06	48.25 ± 0.96	43.37	53.83	97.00 ± 1.15
KNAPSACK	41.06 ± 1.76	39.18 ± 2.59	54.57	55.04	60.00 ± 2.09
QUICKSELECT	4.66 ± 5.98	22.89 ± 2.49	37.05	40.44	77.06 ± 3.78
BINPACKING	60.75 ± 2.49	64.25 ± 1.09	64.07	63.28	66.01 ± 1.55
SCC	51.30 ± 3.91	56.50 ± 2.22	74.68	70.81	89.25 ± 3.49
SUBSETSUM	21.13 ± 2.45	22.75 ± 5.25	41.43	42.05	87.50 ± 6.45
BALANCEDPARTITION	80.55 ± 2.91	91.90 ± 5.52	80.01	91.13	96.73 ± 3.50
3SUM	80.00 ± 0.82	79.75 ± 0.50	81.12	81.67	82.75 ± 1.59
MINCOINCHANGE	9.25 ± 1.86	17.98 ± 2.29	17.33	23.67	42.52 ± 1.47
FLOYD-WARSHALL	12.81 ± 4.03	1.31 ± 0.36	7.59	0.97	0.81 ± 0.08
FRACTIONALKNAPSACK	0.88 ± 0.06	0.86 ± 0.08	0.83	0.85	0.66 ± 0.10

Table 1: Out-of-distribution Micro- F_1 score (top) and MSE for regression tasks (bottom) under **Length OOD** test. The **Tropical Transformer** outperforms all baselines, across combinatorial tasks while delivering $3 \times - 9 \times$ faster inference and using $\sim 20\%$ fewer parameters than Universal Transformer (UT) [17] baselines with iterative attention that approximate the closure.

natural mathematical language for algorithms that must reason over *families* of inputs, particularly those generating such polyhedral structures. Dynamic programming (DP) exemplifies this connection. It is a cornerstone for numerous combinatorial optimization problems. The structure of these problems allows DP value functions to be described as piecewise-linear functions, forming polyhedral structures. They are just recursively constructed circuits over tropical semirings. Each such circuits, namely **tropical circuits**, compute monomials as feasible solutions over the underlying semiring making the DP update step effectively linear within this algebraic framework [14, 15].

¹The challenge by Michael Galkin [16]

Many combinatorial optimization problems are specified by a family of these optimal solutions. For example, shortest-path algorithms such as Floyd-Warshall explicitly manifest this as tropical matrix products and closures, and their feasible solutions trace the faces of a polyhedral complex in parameter space [18]. More broadly, if candidate solutions to combinatorial algorithms are monomials (linear segments) in tropical space, then the computation is just a circuit of tropical gates. Several combinatorial algorithms like shortest paths [19, 20], change making [21, 22], knapsack [23–25], are nothing but recursively constructed tropical circuits. So the **tropical circuit model can give us a bridge between combinatorial optimization and neural architectures to perform sharp, piecewise-linear, and polyhedral computations**, offering a representation around which we can align the attention mechanism at the core of Transformer reasoning.

However, vanilla Transformers [26] are misaligned to this objective. Softmax-normalized dot-product attention lives in Euclidean geometry and produces smooth, quadratic decision boundaries. This smoothness blurs the hard $\arg \max / \arg \min$ structure on which combinatorial algorithms rely. As input length grows, softmax distributions can disperse, resulting in an increasingly flat probability distribution, a.k.a, dispersion [27] or attention fading [28]. Moreover, the exponential sensitivity of softmax makes logits vulnerable to small ℓ_∞ perturbations, harming adversarial robustness. As a result, Transformers equipped with softmax attention fail to extrapolate beyond the training regime of input length or magnitude on combinatorial tasks. For non-algorithmic reasoning tasks, even though injecting positional information [29–31] can alleviate the length extrapolation issue, we believe that the core of the issue lies within the attention mechanism itself.

Our Contribution: As a result, we propose *Tropical Attention*, a novel attention mechanism that incorporates tropical algebraic geometry to perform reasoning and information routing in tropical projective space. In tropical projective space, the attention scores are governed by the tropical Hilbert projective metric; consequently, the induced map is piecewise-linear, idempotent in the aggregation, and non-expansive (1-Lipschitz). In particular, Tropical Attention preserves the polyhedral structure characteristic of combinatorial value functions while inheriting the projective invariance and shortest-path geometry captured by the Hilbert metric [5]. Our viewpoint is aligned with Neuroalgebraic Geometry [32], in which neural computations are modeled by algebraic and tropical structures acting on polyhedral complexes. The mechanism is drop-in for standard Transformer blocks where reasoning occurs in tropical projective space and then returns to Euclidean coordinates for subsequent layers, so training and architectures remain unchanged.

We show that multi-head Tropical Attention (MHTA) realizes the target class of dynamic programs and approximates tropical circuits with *tropical transitive closure*. In particular, our expressivity theorems construct MHTA stacks that simulate tropical circuits without super-polynomial blow-ups in network size or the requirement for recurrent reasoning [17]. In other words, MHTA has sufficient capacity to approximate closure by absorbing the feasible solutions into its tropical polyhedral stratification, thus avoiding recurrent hidden state refinement [33].

Empirically on combinatorial tasks, including **NP-hard/complete** problems, this alignment translates into stronger out-of-distribution generalization together with substantial inference-time and parameter efficiencies relative to recurrent-style baselines. This study, also for the first time extends the applicability of Neural Algorithmic Reasoning [34] beyond the polynomial-time solvable problems to NP-hard/complete combinatorial optimization problems, such as KNAPSACK, BIN PACKING, and BALANCED PARTITION problems.

2 Background and Related Work

2.1 Out-of-Distribution Generalization

An important measure of a supervised learning model’s reasoning is its ability to generalize to inputs that differ fundamentally from those encountered during training. This is known as out-of-distribution (OOD) generalization. Following [35] notations formally, let \mathcal{X} denote the feature space and \mathcal{Y} be the label set. A model $h : \mathcal{X} \rightarrow \mathcal{Y}$ is learned from training examples drawn independently and identically distributed from a training distribution D_{tr} over $\mathcal{X} \times \mathcal{Y}$. Given a distinct test distribution D_{te} on the same space, we define the OOD risk as, $\mathcal{R}_{D_{\text{te}}}(h) := \mathbb{E}_{(x,y) \sim D_{\text{te}}}[\ell(h(x), y)]$, where ℓ

is a loss function. Its empirical estimate on a finite sample S drawn from D_{te} (i.e., $S \sim D_{te}^{|S|}$) is, $\widehat{\mathcal{R}}_S(h) := \frac{1}{|S|} \sum_{(x,y) \in S} \ell(h(x), y)$.

We say that the model h OOD-generalizes from D_{tr} to D_{te} if its OOD risk $\mathcal{R}_{D_{te}}(h)$ remains comparable to its in-distribution risk $\mathcal{R}_{D_{tr}}(h)$, indicating minimal performance degradation despite the distributional shift. In the context of neural algorithmic reasoning, three main types of deviation between D_{tr} and D_{te} are important in measuring a model’s capabilities:

1. Length Generalization Both distributions draw their numerical entries from the same range but the test sequences are strictly longer, $D_{te}(\mathcal{X}) \subsetneq (\mathbb{R}^{>0})^{n_{\max}}$ with $n_{\max} > n_{tr}$. Here, a good performance indicates that the network has learned a parallel or recursive scheme that scales with input size rather than memorizing a fixed shallow circuit.

2. Value generalization The two distributions share the same support with respect to sequence length but $\text{supp}(D_{te}(\mathcal{X}))$ contains magnitudes never encountered during training, i.e. $\text{supp}(D_{te}(\mathcal{X})) \setminus \text{supp}(D_{tr}(\mathcal{X})) \neq \emptyset$. For arithmetic or DP-style tasks, value generalization is the clearest evidence that the model has learned the *rule* rather than the lookup table of seen inputs.

3. Perturbative-noise generalization Noisy data, whether arising from measurement error, adversarial attack, or any other source, often causes models to make mistakes and must be accounted for in model design. To test noise robustness, D_{te} is obtained from D_{tr} by an ℓ_p -bounded, perturbation map $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{X}$ such that $x_{ptb} = \mathcal{A}(x)$ with $\|x_{ptb} - x\|_p \leq \varepsilon$. Robust generalization demands that the risk remains low even under the worst allowed \mathcal{A} . This regime probes the stability and smoothness of the learned function of the architecture. The perturbative noise robustness of Neural Algorithmic Reasoning models is very important for many real-world systems, especially for cryptographic schemes [36].

Length, value, and perturbative noise generalization stress complementary facets of algorithmic competence [37, 38]. Thus, a model as a true reasoning circuit [39, 40] that excels simultaneously in all three regimes offers strong evidence of having internalized the underlying combinatorial procedure rather than a brittle statistical surrogate.

2.2 Softmax Self-Attention Mechanism

Given an input sequence $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d_x}$, let $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q^\top$, $\mathbf{K} = \mathbf{X}\mathbf{W}_K^\top$, $\mathbf{V} = \mathbf{X}\mathbf{W}_V^\top$, where the parameter matrices satisfy $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d \times d_x}$ and $\mathbf{W}_V \in \mathbb{R}^{d_v \times d_x}$. Denote by \mathbf{q}_i^\top and \mathbf{k}_j^\top the i -th and j -th rows of \mathbf{Q} and \mathbf{K} , respectively, and $\tau > 0$ for a temperature parameter. Vanilla self-attention computes, for every token i ,

$$\mathbf{h}_i = \sum_{j=1}^N \alpha_{ij} \mathbf{v}_j, \quad \alpha_{ij} = \text{softmax}_\tau(\langle \mathbf{q}_i, \mathbf{k}_j \rangle) := \frac{\exp(\langle \mathbf{q}_i, \mathbf{k}_j \rangle / \tau)}{\sum_{t=1}^N \exp(\langle \mathbf{q}_i, \mathbf{k}_t \rangle / \tau)}, \quad i = 1, \dots, N, \quad (1)$$

where the softmax is applied independently to each row of the score matrix $\mathbf{Q}\mathbf{K}^\top$. The temperature τ modulates the sharpness of the resulting probability vector, as $\tau \rightarrow 0$ the weights approach a one-hot selection, whereas large τ yields an almost uniform mixture. Equation 1 measures similarity with the Euclidean inner product, which is spherically invariant, meaning that every coordinate contributes equally, regardless of its algorithmic significance. Despite its success in many tasks [26, 41], its geometric and numerical properties are ill-suited to algorithmic reasoning [35, 42]. We summarize the main shortcomings.

1. Inherent blurriness The exponential map assigns a non-zero weight to *every* token; even at low temperatures the second-largest term remains strictly positive. As problem size grows, the gap between the top two logits often decreases (e.g. when costs are drawn from a common distribution), so the resulting distribution cannot converge to a one-hot vector. In practice this leads to *soft* rather than decisive selections, hampering tasks that require exact order statistics [27, 28]. Recent diagnostic suites show that large language models fail on simple tasks of finding minima and second-minima even within In Distribution (ID) length tests [43, 44]. The attention kernel’s inability to sharpen with scale is a primary culprit.

2. Sensitivity to small perturbations Because $\text{softmax}(z) \propto e^z$, a perturbation of size δ in the largest logit changes the corresponding weight by a multiplicative factor e^δ . An adversary who

can alter a single entry of \mathbf{QK}^\top/τ by $\mathcal{O}(\log N)$ may invert the ranking of two tokens, propagating an $\mathcal{O}(1)$ error to downstream activations [45, 46]. This ℓ_∞ -fragility persists even after common stabilisers such as temperature scaling or normalization layers [45].

3. Mismatch with polyhedral decision boundaries In a combinatorial optimization the value function is a tropical polynomial—piecewise linear with faces aligned to coordinate hyperplanes [5, 47]. The quadratic forms generated by Euclidean dot products carve the domain into spherical caps [48] rather than polyhedral cones; reproducing a DP recurrence therefore demands exponentially many heads or layers unless the desired structure is injected by hand.

4. Temperature–gradient dilemma Driving the distribution toward a hard arg max necessitates lowering the temperature parameter τ . Yet as $\tau \rightarrow 0$ the Jacobian of the softmax grows like τ^{-1} , causing gradient explosion/vanishing [49]. Careful schedule tuning or gradient clipping becomes mandatory [45], adding hyper-parameter overhead.

2.3 Neural Algorithmic Reasoning

The problem of bridging symbolic algorithms and differentiable models has become known as *Neural Algorithmic Reasoning* (NAR). Neural Algorithmic Reasoning involves developing neural models and learning procedures to facilitate the *internalization* of algorithms directly in models’ weights. Starting from early work [34] that aimed to demonstrate the applicability of Graph Neural Networks (GNNs) to approximate classical algorithms [50], the community has subsequently developed and expanded further in different directions [42, 51–58]. Some notable applications are constructing [59] and enumerating [60] combinatorial structures of a particular type [59], and dynamic programming [40, 61]. A fundamental objective of NAR is to achieve robust out-of-distribution (OOD) generalization through *algorithmic alignment*. Typically, models are trained and validated on small sets/sequences/graphs and tested on larger sets/sequences/graphs. This is inspired by classical algorithms’ *size-invariance*, where correctness of the solution is maintained irrespective of the input size. Our work pursues this objective from a fresh, tropical geometric angle and provides universality guarantees and expressivity for an attention core within the NAR framework.

Recent work exclusively tried to quantify NAR failures when test sequences are longer [37, 38] or numerically larger [35], but have not assessed noise robustness scenarios. Perturbative noise itself is also of importance since real-world deployments must withstand the worst-case inputs and noise. Robustness in this setting is a test for whether a model has internalized genuine algorithmic structure rather than superficial statistical cues. Hence, we introduce *perturbative-noise generalization* as a third pillar for NAR benchmarking and show that Tropical Transformer demonstrates systematic gains across all three axes.

Based on the past related works, one can establish a demand for an attention mechanism that (i) is expressive and respects the underlying geometric structure of combinatorial algorithms, (ii) mitigates softmax dispersion that hampers OOD generalization, and (iii) delivers OOD noise robustness benefits. Tropical Attention positions itself at this intersection, drawing on a decade of tropical geometric insights to advance neural algorithmic reasoning. **To best of our knowledge, this level of incorporating Tropical algebraic geometry (not just pre-post processing arithmetic [13, 40, 62]) to define a new mathematically grounded reasoning architecture has not been done before.**

3 Tropical Attention: Reasoning in Tropical projective space

Tropical Attention arises from *combinatorial algorithm alignment*, that is information exchange is governed by order statistics, namely maxima, minima, and interval widths rather than absolute magnitudes. These operations live naturally in the tropical semiring, whose idempotence, translation covariance, and projective scale invariance match the properties of tropical geometry and it’s computational mirror, tropical circuits. By contrast, the dot-product–softmax kernel depends on absolute scale and temperature, yields smooth (quadratic) decision boundaries, and exhibits dispersion as sequence length grows, thereby blurring the sharp structure required for decisive reasoning.

Our goal is to replace the *dot-product*, Softmax-based kernel of vanilla self-attention with a reasoning core that (i) takes place in the tropical polyhedral complex (ii) preserves the piecewise-linear geometry of combinatorial problems, and (iii) inherits the *1-Lipschitz* robustness of tropical linear maps. To do so, we perform a **tropicalization** map of queries, keys, and values to piecewise-linear projective

spaces carved out by polyhedral constraints, compute attention weights with the *tropical Hilbert projective metric* (See Appendix B and [5] for details on tropical algebraic geometry.), aggregate by a tropical matrix–vector product, and finally map the result back to Euclidean space so that the rest of the original algorithm (e.g Transformer) modular stack is untouched. We present the framework relating robustness and piecewise-linearity of maps and show how our proposed scheme offers improvements on OOD generalization tasks.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the token embedding of the input. We define the *tropicalization map* by going to an amoeba representation of the input with a learnable valuation map, $\Phi : \mathbb{R}^{N \times d} \rightarrow (\mathbb{TP}^{d-1})^N$

$$\Phi_\lambda(\mathbf{X})_i = \mathbf{U}_i - \max_{1 \leq r \leq d} \mathbf{U}_{ir} \cdot \mathbf{1}_d, \text{ where } \mathbf{U} = \log(\max(\mathbf{0}, \mathbf{X})) \in \mathbb{R}^{N \times d} \quad (2)$$

for each row $i \in 1, \dots, N$. The constant shift enforces $\max_i \phi_\lambda(\mathbf{x})_i = \epsilon$, so the output of ϕ_λ always lies in the tropical simplex, $\Delta^{d-1} := \{z \in \mathbb{R}^d \mid \max_i z_i = \epsilon\}$, where every vector is projectively equivalent to exactly one point in the tropical simplex. In other words, Φ is a section of the quotient $\mathbb{R}^d / \mathbb{R}\mathbf{1}$.

Lemma 3.1. *For every embedded coordinate $i \in [N]$, the function*

$$v_\lambda(x) := [\phi_\lambda(x)]_i = \begin{cases} \log(x) - \lambda, & x > 0, \\ -\infty, & x \leq 0, \end{cases}$$

where ϕ_λ is a (projective) valuation map. Hence the shifted map $\tilde{v}(x) = v_\lambda(x) + \lambda = \log(\max(0, x))$ is an Archimedean valuation in the classical sense, and Φ is a matrix-valued valuation modulo tropical scalars; its image lies in the tropical simplex.

After mapping each input token to tropical projective space, $\mathbf{Z} = \phi_\lambda(\mathbf{X}) \in \mathbb{TP}^{N \times d-1}$, we compute attention independently across H heads.

Definition 3.1 (Multi-head Tropical Attention (MHTA)). *Let $d_k = d/H$ be a fixed head dimension. Then, for every head ($h \in [H]$) one can choose learnable matrices $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)} \in \mathbb{R}^{d_k \times d}$ and define the tropical linear projections [7] $\mathbf{Q}^{(h)} = \mathbf{Z} \odot \mathbf{W}_Q^{(h)\top}, \mathbf{K}^{(h)} = \mathbf{Z} \odot \mathbf{W}_K^{(h)\top}, \mathbf{V}^{(h)} = \mathbf{Z} \odot \mathbf{W}_V^{(h)\top}$ where \odot denotes max-plus matrix multiplication, $(\mathbf{A} \odot \mathbf{B})_{ij} = \max_t \{\mathbf{A}_{it} + \mathbf{B}_{tj}\}$. Then, using $d_{\mathbb{H}}$ the tropical Hilbert projective metric, defined in B.4, we will have the tropical attention score*

$$\mathbf{S}_{ij}^{(h)} = -d_{\mathbb{H}}(\mathbf{q}_i^{(h)}, \mathbf{k}_j^{(h)}), \quad i, j \in [N],$$

that comes with Projective Invariance and Non-expansiveness condition (discussed in B.4). Thereafter, the head outputs are aggregated via tropical matrix–vector product,

$$\mathbf{C}_i^{(h)} = \bigoplus_{j=1}^N \mathbf{S}_{ij}^{(h)} \odot \mathbf{v}_j^{(h)} = \max_j \{\mathbf{S}_{ij}^{(h)} + \mathbf{v}_j^{(h)}\}, \quad i \in [N].$$

The tropical context picks the value that best aligns projectively with the query. Then, the contexts per head, will be mapped to the Euclidean domain via a smooth inverse map (devaluation) $\psi(z) = \exp(z)$, and concatenated back to the original dimension, $\mathbf{H} = [\psi(\mathbf{C}^{(1)}) \parallel \dots \parallel \psi(\mathbf{C}^{(H)})] \in \mathbb{R}^{N \times d}$.

Why Tropical Attention? Every operation inside MHTA is piecewise linear and aligned with tropical geometry (not just tropical arithmetic). Hence the entire network computes a tropical polygonal map whose cells are polyhedral cut out by hyperplanes. This is aligned with combinatorial algorithms, whose solutions correspond to a vertex of a polytope and every decision boundary is a facet. Training a transformer with Tropical attention therefore starts from a hypothesis space that already mirrors the solution structure of a combinatorial algorithms. That is what we call a **polyhedral inductive bias**. By contrast, Euclidean softmax attention inserts an exponential map, blurring the sharp decisions on their input data. Classical transformers modulate the entropy–versus–sharpness trade-off through a temperature parameter in the softmax; MHTA sharpness is built in and temperature-free. Moreover, since every intermediate representation of MHTA lies in the projective simplex Δ^{d-1} , going through $\arg \max$ is well-defined (no equal maxima except on a set of measure 0) and is stable by global scaling meaning that shifting the entire vector by $\lambda \in \mathbb{R}^d$ does not alter which index attains the maximum. In other words, only relative relations between inputs matter, thus Tropical attention is inherently robust against distribution shifts.

Furthermore, each MHTA head can function as a tropical gate in a **tropical circuit**. A tropical circuit is a finite acyclic digraph whose input vertices store either a variable or a non-negative real constant, while every internal vertex has in-degree two and outputs the *maximum* or the *sum* of its two predecessors. The circuit’s size is the number of internal gates. Classical pure DP algorithms are recursive tropical circuits of this kind; consequently, lower bounds for tropical circuits translate directly into limits for such DP schemes. An MHTA head can also be interpreted as a single tropical gate. A single head implements the composite transformation $(u, v) \mapsto \max_j \{ S_{ij} + v_j \}$, where the score S_{ij} itself is obtained through several applications of max and $+$ gates. The outer maximization provides the \oplus -gate, while the summand v_j furnishes a \odot -gate acting on two variable inputs. Thus every head is a compact, differentiable wrapper around the two tropical primitives, and a full multi-head layer is simply a collection of such gates operating in parallel on a shared input tape, creating a **tropical transitive closure**. Training a multi-layer MHTA therefore amounts to discovering how these gates should be wired together, rather than coaxing a Euclidean softmax kernel to emulate max-plus algebra indirectly. As a result of developing MHTA, we prove that it is a universal approximator of max-plus dynamic programming for combinatorial optimization with closure (Theorem C.3, Corollary C.3.1, and Theorem 3.2).

Theorem 3.2 (Simulation of max-plus dynamic programs). *Let (S, E) be a finite directed acyclic graph with $|S| = N$ vertices and edge weights $\{w_{uv}\}_{(u,v) \in E} \subset \mathbb{T}$. Fix a source vertex $v_0 \in S$ and consider the max-plus Bellman recursion*

$$d_v(t+1) = \bigoplus_{u: (u,v) \in E} (w_{uv} \odot d_u(t)), \quad d_v(0) = \delta_{v,v_0}, \quad t \in \mathbb{N}.$$

Theorem 3.2, Theorem C.3, and Theorems C.4 and C.5 show *upper bounds*, i.e., sufficient conditions, of T and N such that a stack of T MHTA layers and N heads can approximate any horizon tropical circuit for a dynamic program.

Remark 3.1. *A shallow MHTA is sufficiently expressive to approximate the tropical transitive-closure map, and thereby avoids explicit recurrence, to encode a very rich polyhedral geometry of combinatorial algorithmic reasoning into attention and provide a non-recurrent (one-shot) solution. However, in the worst-case scenario, the non-recurrent representation requires head-width proportional to the number of active path monomials up to $\mathcal{O}(n^2 2^k)$. Depth- (T) stacks realize the same computation with polynomial resources by distributing the computation across layers.*

In other words, Tropical Attention has sufficient capacity to approximate closure by absorbing the feasible solutions into its tropical polyhedral stratification, thus avoiding explicit recurrence. **Consequently, a MHTA module learns the closure directly rather than implementing the recurrence step by step.** By contrast, Recurrent Transformers such as the Universal Transformer [17] incorporate a depth-wise recurrence to learn the hidden state representation per token. The recurrent function evolves in parallel across token positions while exchanging information through self-attention. In the next section, Section 4, we show that, Tropical Transformer provides a stronger out-of-distribution generalization than Universal Transformer with Dynamic Halting mechanism, while delivering substantially faster inference with much fewer parameters.

4 Experiments

We evaluate Tropical transformers on eleven combinatorial tasks (see E), several of which are NP-hard/complete problems, and the the Long Range Arena (LRA) benchmark [63], a standard for testing transformers on long-sequence tasks across text, image, and math domains. For each combinatorial task we measure three complementary forms of out-of-distribution (OOD) generalization: Length OOD (longer inputs), Value OOD (unseen magnitudes), and noise robustness (perturbed inputs). A procedure to compare between vanilla attention and Tropical Attention is described in Appendix D. For our experiments we custom generated both train and test datasets following procedures from the canonical algorithmic reasoning benchmark CLRS [43]. This decision was due largely to the absence of NP-hard and NP-complete problems in the CLRS benchmark, but also because our framework is designed for sequence and set-based data modalities and our OOD evaluation includes two extra new evaluation pipelines, adversarial perturbations and value generalization. All datasets, generation scripts, and OOD protocols are described in Appendix E and F.

For our experiment we consider three variants, (i) **Vanilla**: Standard transformer encoder with softmax dot-product attention. (ii) **Adaptive**: Transformer equipped with adaptive softmax attention

from [27]. (iii) **Tropical**, which every attention block is replaced by MHTA. For length OOD tests, we also compare with 32-step Universal Transformer (UT) with dynamic halting, as a recurrent attention model, under the vanilla softmax and adaptive-temperature softmax kernels. To ensure a fair comparison, all variants share identical backbone hyperparameters: depth, width, and number of heads. The only architectural difference is the attention kernel. Crucially, no model sees OOD examples during optimization. We follow a uniform procedure in which each model is trained from scratch under the same training regime with task specific fixed input sequence lengths and value ranges.

Out-of-Distribution Protocols In order to assess OOD generalization, we construct three stress tests: (i) **Length OOD** – inputs drawn from the same value range but with longer input sequence lengths. (ii) **Value OOD** – the input sequence lengths are fixed and the values are sampled from an increasingly large range (for example, if the models trained on inputs sampled from the range $[-5, 5]$ an out of distribution evaluation would be inputs sampled from the range $[-10, 10]$). (iii) **Perturbative noise OOD** – the input sequence lengths are fixed and the values are from the same input range, but a subset of the input values are perturbed randomly.

5 Results and Discussion

Section 2.2 elaborated why and how softmax self-attention - and its descendants - are incapable of generalizing to OOD inputs in combinatorial problems, and Section 3 discussed *why* Tropical Attention can generalize in the combinatorial regime. With our experimentation, we seek to show *if* and, if so, *how* Tropical Attention generalizes in this domain.

To answer *if* Tropical Attention generalizes, we report the numerical results of our experimentation in Tables 1 and 3. The Tropical attention architecture achieves superior OOD performance to both the Vanilla and Adaptive softmax attention. Notably, this out performance can be seen in both regression and classification combinatorial tasks and across OOD protocols, validating our theoretical results from Section 3. The Tropical architecture’s ability to generalize well across OOD protocols and problem sets, especially the notorious Quickselect, suggests that instead of simply learning the specific data it is trained on, these purpose-built models learn the underlying polyhedral structure of the combinatorial algorithm. The results, from Tables 1 and 2, demonstrate two key findings. MHTA shows a strong performance, even when compared against an iterative attention class of transformers with a dynamic Adaptive Computation Time (ACT) mechanism where they can approximate an algorithmic closure, the Tropical Attention model still achieves better OOD performance across all algorithmic tasks. The more interesting results are that our model achieves these results while being on average $3 \times -9 \times$ **faster at inference** and using **20% fewer parameters** than the Universal Transformer baselines. Evaluating on the Long Range Arena (LRA) benchmark as shown in Table 4, Tropical Transformer achieves highly competitive, State-of-the-art (SOTA) results, placing *second* overall in average accuracy across the benchmark’s tasks. This performance shows that the benefits of Tropical Attention stand as a viable and powerful mechanism for general-purpose sequence modeling.

Table 2: Average inference time per sample across all tasks and parameter count.

Model	CPU (ms)	GPU (ms)	Params.
Vanilla UT w/ ACT	6.285	0.027	50,242
Adaptive UT w/ ACT	7.898	0.018	50,242
Tropical Transformer	1.949	0.003	40,961

In order to understand *how* Tropical Attention outperforms, we explore the tropical Attention maps relative to vanilla and adaptive attention maps for both Quickselect and Knapsack. On the QUICKSELECT task the goal is to find the k -th smallest elements. For such tasks, maintaining focus means the ability to allocate a high attention score to the correct items, creating sharp spikes in the heatmap. Contrarily, losing focus means the attention uniformly disperses across elements, with no single item receiving a high score. Figure 1, which depicts attention maps as sequence length increases for all models, shows that the vanilla and adaptive models exhibit attention fading. In contrast, the Tropical Attention consistently shows bright, distinct bands, indicating it continues to allocate sharp attention even at larger length sequences. Figure 2, replicating visualizations from [27], depicts a normalized attentional head for the Quickselect task for a batch of 32 sets, over the 8 items with the largest keys by the ℓ_2 -norm. If the head operates correctly, it must allocate sharp attention to the position of k -th smallest element. Again, we see that the attention on both softmax models quickly dilute/disperse as sequence length grows OOD while the tropical Attention maintains focus.

Table 3: Out-of-distribution performance under **Value OOD** and **Perturbative Noise** tests. Top: Micro- F_1 for classification tasks; Bottom: MSE for regression tasks.

Algorithmic Tasks	Value OOD			Perturbative Noise		
	Vanilla	Adaptive	Tropical	Vanilla	Adaptive	Tropical
CONVEXHULL	22.75 ± 3.59	23.77 ± 3.10	34.25 ± 1.71	90.75 ± 2.22	91.00 ± 2.16	96.00 ± 2.16
KNAPSACK	38.87 ± 3.43	26.92 ± 1.33	49.67 ± 2.01	67.85 ± 3.19	68.36 ± 3.51	74.67 ± 3.13
QUICKSELECT	74.22 ± 2.30	74.30 ± 1.99	71.10 ± 3.11	33.87 ± 7.11	34.82 ± 4.79	57.22 ± 5.01
BINPACKING	67.26 ± 3.70	74.23 ± 1.51	78.54 ± 1.89	55.38 ± 5.10	60.64 ± 3.92	61.19 ± 4.33
SCC	78.51 ± 3.08	81.38 ± 2.62	74.86 ± 5.01	70.00 ± 5.98	71.33 ± 1.96	69.86 ± 4.17
SUBSETSUM	34.75 ± 6.60	28.50 ± 10.12	79.25 ± 5.38	3.75 ± 1.50	3.00 ± 1.63	72.75 ± 10.01
BALANCEDPARTITION	63.40 ± 4.29	56.57 ± 1.18	55.76 ± 5.63	51.06 ± 2.66	57.06 ± 1.08	57.29 ± 1.33
3SUM	26.00 ± 3.16	26.25 ± 3.50	22.00 ± 2.16	47.50 ± 9.47	49.25 ± 9.22	65.25 ± 3.59
MINCOINCHANGE	23.64 ± 4.07	2.20 ± 1.13	33.18 ± 5.64	22.12 ± 2.75	18.44 ± 4.49	33.75 ± 4.89
FLOYD-WARSHALL	87.68 ± 5.65	56.30 ± 3.04	55.30 ± 4.36	7.54 ± 3.63	5.29 ± 2.56	4.39 ± 1.62
FRACTIONALKNAPSACK	0.24 ± 0.12	0.17 ± 0.03	0.08 ± 0.03	0.05 ± 0.02	0.03 ± 0.01	0.02 ± 0.01

Similarly, Figure 3 depicts length OOD on the full attention head for the KNAPSACK problem, a classic dynamic program corresponding to tropical circuits. Each model begins sharp in distribution, but the Tropical Attention head maintains the same activation pattern across each input length, strongly suggesting it has learned and internalized the underlying combinatorial problem and polyhedral structure vice the specific training data indicated by the vanilla and adaptive fading attention patterns.

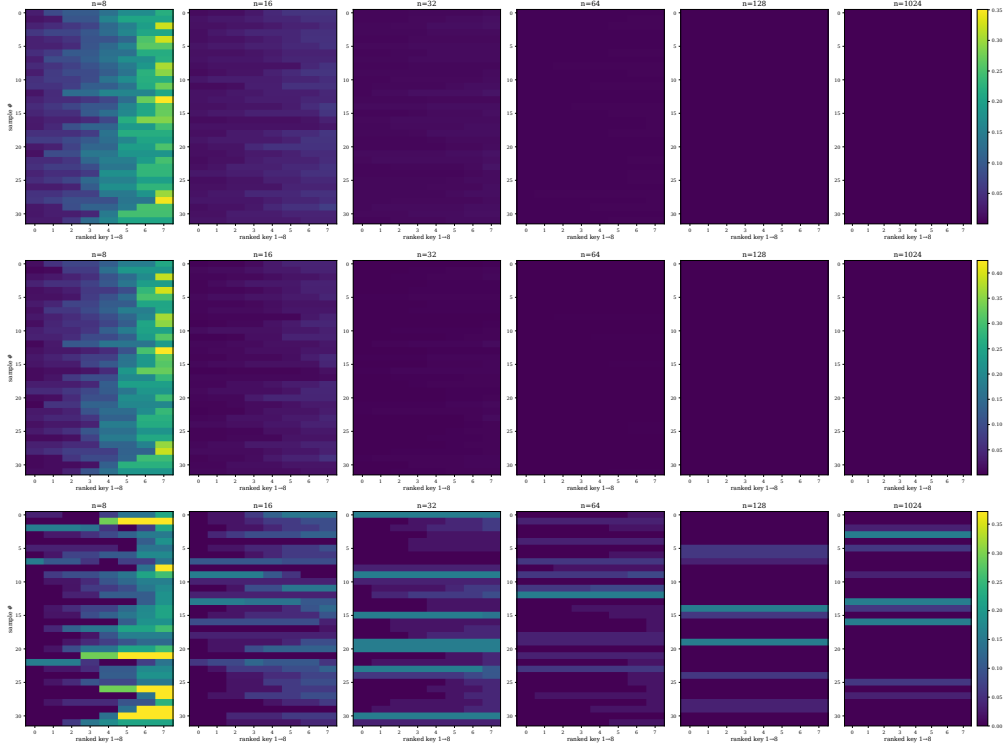


Figure 2: Stacked attention head representations for QUICKSELECT under (a) Vanilla, (b) Adaptive, and (c) **Tropical** models. Each model was trained on length 8 sequences and was evaluated from Left to Right on length 16 to 1024 sequences. Each image was generated by a batch of 32 inputs. The columns are the 8 largest keys by ℓ_2 -norm. Heatmap values are the attention of the row item at the column key.

Table 4: We report classification accuracy for each task and the average accuracy across all tasks. All results are taken from their respective papers, except for Adaptive Softmax, which we re-implemented and evaluated.

Models	ListOps	Text	Retrieval	Image	Pathfinder	Avg.	Complexity
Transformer [26]	36.37	64.27	57.46	42.44	71.40	54.39	$\mathcal{O}(n^2)$
Longformer [64]	35.63	62.85	56.89	42.22	69.71	53.46	$\mathcal{O}(n)$
Linformer [65]	35.70	53.94	52.27	38.56	76.34	51.36	$\mathcal{O}(n)$
Performer [66]	18.01	65.40	53.82	42.77	77.50	51.41	$\mathcal{O}(n)$
Elliptical [67]	37.8	65.6	80.3	40.2	73.2	61.24	$\mathcal{O}(n^2)$
Fourierformer [68]	40.73	75.02	<u>85.35</u>	53.17	83.43	67.54	$\mathcal{O}(n \log n)$
MEGA [69]	<u>63.14</u>	90.43	91.25	90.44	<u>96.01</u>	86.25	$\mathcal{O}(n \log n)$
AdaptiveSoft. [27]	47.15	<u>75.52</u>	79.56	51.58	80.94	66.95	$\mathcal{O}(n^2)$
Tropical Transformer	68.65	70.13	64.82	<u>60.04</u>	97.33	<u>72.79</u>	$\mathcal{O}(n^2)$

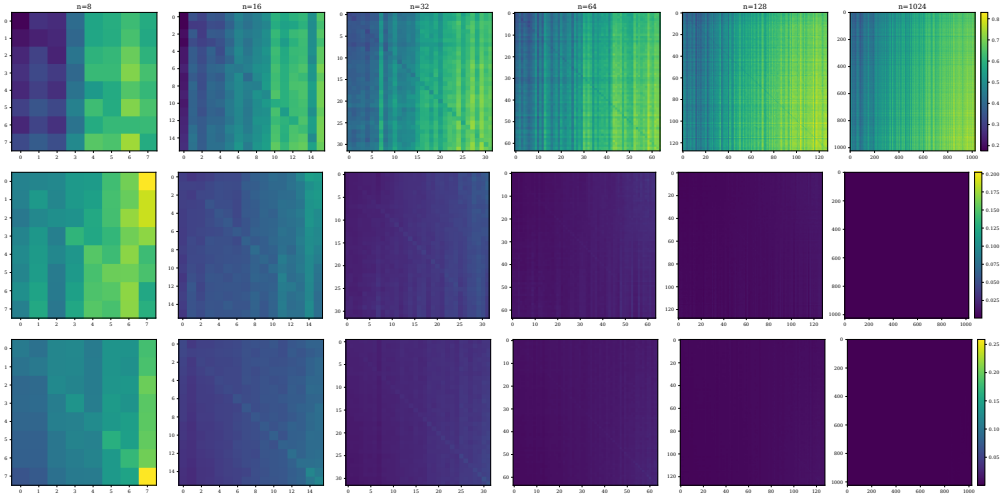


Figure 3: (top) **Tropical Attention** with sharp attention maps on learning the KNAPSACK algorithm, showcasing a size-invariance and OOD lengths generalization behavior far beyond training (16 \rightarrow 1024). In contrast, both (middle) adaptive-softmax and (bottom) vanilla-softmax heads dilute and disperse as sequence length grows, failing to generalize.

6 Conclusion

We introduced Tropical Attention, replacing softmax-normalized dot-product attention with an attention mechanism that operates in tropical projective space. On the theory side, we showed that multi-head Tropical Attention (MHTA) simulates tropical circuits and realizes tropical transitive closure via finite-depth compositions, with polynomial resource bounds (Theorem C.3, Corollary C.3.1, Theorem 3.2). These guarantees provide a principled account of scale-invariance and sharp, polyhedral decision boundaries, properties that are essential for reasoning models expected to generalize beyond their training distribution. Empirically, across various combinatorial problems, Tropical transformer achieved SOTA out-of-distribution generalization, and delivered stronger noise robustness, while being much faster at inference with fewer parameters than the recurrent/iterative attention baselines. These findings carry an important message for both neural algorithmic reasoning (NAR) and Large Reasoning Model (LRM) communities: **tropicalization of reasoning and going beyond softmax not only enriches the algorithmic power of attention mechanisms but also yields tangible gains on reasoning tasks.** We believe Tropical Attention opens compelling avenues for hybrid semiring architectures and for leveraging tropical geometry to reason over discrete structures within deep learning systems. Future work will explore sparse tropical kernels and applications to graph-theoretic domains, aiming for ever-stronger generalization guarantees in neural algorithm and reasoning synthesis.

Acknowledgments and Disclosure of Funding

This research was supported by the Excellence Cluster ORIGINS, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC-2094-390783311. B.H extends his gratitude to the organizers and the wonderful instructors, Marta Panizzut and Margarida Melo, of the 2024 Trieste Algebraic Geometry Summer School (TAGSS) on Tropical Geometry, where the idea of the project was sparked. K.P., C.T. and R.Y. are partially supported by NSF Division of Mathematical Sciences: Statistics Program DMS 2409819.

References

- [1] Diane Maclagan and Bern Sturfs. *Introduction to Tropical Geometry*. American Mathematical Society, 2015. [1](#), [B.3](#)
- [2] Federico Ardila and Mike Develin. Tropical hyperplane arrangements and oriented matroids, 2007.
- [3] Federico Ardila-Mantilla, Christopher Eur, and Raul Penaguiao. The tropical critical points of an affine matroid. *SIAM Journal on Discrete Mathematics*, 38(2):1930–1942, 2024.
- [4] Alex Fink and Felipe Rincón. Stiefel tropical linear spaces, 2015.
- [5] Michael Joswig. *Essentials of Tropical Convexity*. American Mathematical Society, 2021. [1](#), [1](#), [2.2](#), [3](#)
- [6] Kurt Pasque, Christopher Teska, Ruriko Yoshida, Keiji Miura, and Jefferson Huang. Tropical decision boundaries for neural networks are robust against adversarial attacks, 2024. [1](#)
- [7] Ruriko Yoshida, Georgios Aliatimis, and Keiji Miura. Tropical neural networks and its applications to classifying phylogenetic trees. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2024. [3.1](#)
- [8] Ruriko Yoshida, Leon Zhang, and Xu Zhang. Tropical principal component analysis and its application to phylogenetics, 2017.
- [9] Petros Maragos, Vasileios Charisopoulos, and Emmanouil Theodosios. Tropical geometry and machine learning. *Proceedings of the IEEE*, 109(5):728–755, 2021.
- [10] Marie-Charlotte Brandenburg, Georg Loho, and Guido Montúfar. The real tropical geometry of neural networks. *arXiv preprint arXiv:2403.11871*, 2024.
- [11] Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5824–5832. PMLR, 10–15 Jul 2018.
- [12] Christoph Hertrich and Leon Sering. Relu neural networks of polynomial size for exact maximum flow computation. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 187–202. Springer, 2023.
- [13] Davide Bacciu, Francesco Landolfi, and Danilo Numeroso. A tropical view of graph neural networks. *ESANN 2023 proceedings*, 2023. [1](#), [2.3](#)
- [14] Stéphane Gaubert. Tropical considerations in dynamic programming. Presentation at Optimization, Games, and Dynamics, Institut Henri Poincaré, Paris, November 2011. Based on joint work with Akian, Guterman, Allamigeon, Katz, Vigeral, McEneaney, and Qu. [1](#), [B](#)
- [15] Stasys Jukna. Lower bounds for tropical circuits and dynamic programs. *Theory of Computing Systems*, 57(1):160–194, October 2014. [1](#), [C.1](#)
- [16] Michael Galkin. Graph & Geometric ML in 2024: Where we are and what’s next, 2024. Medium Article. [1](#)
- [17] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018. [1](#), [1](#), [3](#)
- [18] Michael Joswig and Benjamin Schröter. Parametric shortest-path algorithms via tropical geometry. *Mathematics of Operations Research*, 47(3):2065–2081, August 2022. [1](#)
- [19] Mehryar Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002. [1](#)

- [20] Peter Höfner and Bernhard Möller. Dijkstra, floyd and warshall meet kleene. *Formal Aspects of Computing*, 24(4):459–476, 2012. 1
- [21] Marek Cygan, Marcin Mucha, Karol Węgrzycki, and Michał Włodarczyk. On problems equivalent to $(\min, +)$ -convolution. *ACM Transactions on Algorithms (TALG)*, 15(1):1–25, 2019. 1
- [22] Timothy M Chan and Qizheng He. More on change-making and related problems. *Journal of Computer and System Sciences*, 124:159–169, 2022. 1
- [23] Silvano Martello, David Pisinger, and Paolo Toth. New trends in exact algorithms for the 0–1 knapsack problem. *European Journal of Operational Research*, 123(2):325–332, 2000. 1
- [24] Karl Bringmann and Alejandro Cassis. Faster 0-1-knapsack via near-convex min-plus-convolution. *arXiv preprint arXiv:2305.01593*, 2023.
- [25] Kyriakos Axiotis and Christos Tzamos. Capacitated dynamic programming: Faster knapsack and graph algorithms. *arXiv preprint arXiv:1802.06440*, 2018. 1
- [26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1, 2.2, 4
- [27] Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. softmax is not enough (for sharp out-of-distribution), 2024. 1, 2.2, 4, 5, 4
- [28] Ken M. Nakanishi. Scalable-softmax is superior for attention, 2025. 1, 2.2
- [29] Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer, 2022. 1
- [30] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.
- [31] Shaoxiong Duan, Yining Shi, and Wei Xu. From interpolation to extrapolation: Complete length generalization for arithmetic transformers, 2024. 1
- [32] Giovanni Luca Marchetti, Vahid Shahverdi, Stefano Mereta, Matthew Trager, and Kathlén Kohn. An invitation to neuroalgebraic geometry. *arXiv e-prints*, pages arXiv–2501, 2025. 1
- [33] Matteo Tiezzi, Michele Casoni, Alessandro Betti, Tommaso Guidi, Marco Gori, and Stefano Melacci. On the resurgence of recurrent models for long sequences—survey and research opportunities in the transformer era. *arXiv preprint arXiv:2402.08132*, 2024. 1
- [34] Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. Neural execution of graph algorithms. In *International Conference on Learning Representations*, 2020. 1, 2.3
- [35] Artur Back de Luca, George Giapitzakis, Shenghao Yang, Petar Veličković, and Kimon Fountoulakis. Positional attention: Expressivity and learnability of algorithmic computation, 2025. 2.1, 2.2, 2.3
- [36] Pawan Kumar Pradhan, Sayan Rakshit, and Sujoy Datta. Lattice based cryptography : Its applications, areas of interest & future scope. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 988–993, 2019. 2.1
- [37] Robert R. Nerem, Samantha Chen, Sanjoy Dasgupta, and Yusu Wang. Graph neural networks extrapolate out-of-distribution for shortest paths, 2025. 2.1, 2.3
- [38] Sadegh Mahdavi, Kevin Swersky, Thomas Kipf, Milad Hashemi, Christos Thrampoulidis, and Renjie Liao. Towards better out-of-distribution generalization of neural algorithmic reasoning tasks, 2023. 2.1, 2.3
- [39] Xihan Li, Xing Li, Lei Chen, Xing Zhang, Mingxuan Yuan, and Jun Wang. Circuit transformer: A transformer that preserves logical equivalence, 2025. 2.1
- [40] Andrew Joseph Dudzik and Petar Veličković. Graph neural networks are dynamic programmers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2.1, 2.3
- [41] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2.2
- [42] Gleb Rodionov and Liudmila Prokhorenkova. Discrete neural algorithmic reasoning, 2025. 2.2, 2.3

- [43] Larisa Markeeva, Sean McLeish, Borja Ibarz, Wilfried Bounsi, Olga Kozlova, Alex Vitvitskyi, Charles Blundell, Tom Goldstein, Avi Schwarzschild, and Petar Veličković. The clrs-text algorithmic reasoning language benchmark, 2024. 2.2, 4
- [44] Euan Ong and Petar Veličković. Learnable commutative monoids for graph neural networks, 2022. 2.2
- [45] Hao Xuan, Bokai Yang, and Xingyu Li. Exploring the impact of temperature scaling in softmax for classification and adversarial robustness, 2025. 2.2
- [46] Gihyun Kim, Juyeop Kim, and Jong-Seok Lee. Exploring adversarial robustness of vision transformers in the spectral perspective, 2023. 2.2
- [47] Stasys Jukna. Tropical circuit complexity. *Limits of Pure Dynamic Programming/by Stasys Jukna.-*, 2023. 2.2
- [48] Stefan K. Nielsen, Laziz U. Abdullaev, Rachel S. Y. Teo, and Tan M. Nguyen. Elliptical attention, 2024. 2.2
- [49] Akhil Kedia, Mohd Abbas Zaidi, Sushil Khyalia, Jungho Jung, Harshith Goka, and Haejun Lee. Transformers get stable: An end-to-end signal propagation theory for language models, 2024. 2.2
- [50] Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7):100273, July 2021. 2.3
- [51] Hefei Li, Chao Peng, Chenyang Xu, and Zhengfeng Yang. Open-book neural algorithmic reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2.3
- [52] Gleb Rodionov and Liudmila Prokhorenkova. Neural algorithmic reasoning without intermediate supervision. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [53] Dobrik Georgiev Georgiev, Danilo Numeroso, Davide Bacciu, and Pietro Lio. Neural algorithmic reasoning for combinatorial optimisation. In *The Second Learning on Graphs Conference*, 2023.
- [54] Dobrik Georgiev Georgiev, JJ Wilson, Davide Buffelli, and Pietro Lio. Deep equilibrium algorithmic reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [55] Montgomery Bohde, Meng Liu, Alexandra Saxton, and Shuiwang Ji. On the markov property of neural algorithmic reasoning: Analyses and methods. In *The Twelfth International Conference on Learning Representations*, 2024.
- [56] Kaijia Xu and Petar Veličković. Recurrent aggregators in neural algorithmic reasoning. In *The Third Learning on Graphs Conference*, 2024.
- [57] Wilfried Bounsi, Borja Ibarz, Andrew Joseph Dudzik, Jessica B Hamrick, Larisa Markeeva, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. Transformers meet neural algorithmic reasoners, 2025.
- [58] Borja Ibarz, Vitaly Kurin, George Papamakarios, Kyriacos Nikiforou, Mehdi Bennani, Róbert Csordás, Andrew Joseph Dudzik, Matko Bošnjak, Alex Vitvitskyi, Yulia Rubanova, Andreea Deac, Beatrice Bevilacqua, Yaroslav Ganin, Charles Blundell, and Petar Veličković. A generalist neural algorithmic learner. In *The First Learning on Graphs Conference*, 2022. 2.3
- [59] Yunhui Jang, Dongwoo Kim, and Sungsoo Ahn. Graph generation with \mathcal{K}^2 -trees. In *The Twelfth International Conference on Learning Representations*, 2024. 2.3
- [60] Baran Hashemi, Roderic Guigo Corominas, and Alessandro Giacchetto. Can transformers do enumerative geometry? In *The Thirteenth International Conference on Learning Representations*, 2025. 2.3
- [61] Borja Ibarz, Vitaly Kurin, George Papamakarios, Kyriacos Nikiforou, Mehdi Bennani, Róbert Csordás, Andrew Joseph Dudzik, Matko Bošnjak, Alex Vitvitskyi, Yulia Rubanova, Andreea Deac, Beatrice Bevilacqua, Yaroslav Ganin, Charles Blundell, and Petar Veličković. A generalist neural algorithmic learner. In Bastian Rieck and Razvan Pascanu, editors, *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pages 2:1–2:23. PMLR, 09–12 Dec 2022. 2.3
- [62] Andrew Joseph Dudzik, Tamara von Glehn, Razvan Pascanu, and Petar Veličković. Asynchronous algorithmic alignment with cocycles. In *Learning on Graphs Conference*, pages 3–1. PMLR, 2024. 2.3
- [63] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2020. 4
- [64] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 4

- [65] Sinong Wang, Belinda Z Li, Madian Khabisa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. **4**
- [66] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. **4**
- [67] Stefan Nielsen, Laziz Abdullaev, Rachel SY Teo, and Tan Nguyen. Elliptical attention. *Advances in Neural Information Processing Systems*, 37:109748–109789, 2024. **4**
- [68] Tan Nguyen, Minh Pham, Tam Nguyen, Khai Nguyen, Stanley J Osher, and Nhat Ho. Transformer with fourier integral attentions. *arXiv preprint arXiv:2206.00206*, 2022. **4**
- [69] Xueze Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: Moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*, 2022. **4**
- [70] Roan Talbut, Daniele Tramontano, Yueqi Cao, Mathias Drton, and Anthea Monod. Probability metrics for tropical spaces of different dimensions, 2024. **2**
- [71] MARIANNE AKIAN, STÉPHANE GAUBERT, and ALEXANDER GUTERMAN. Tropical polyhedra are equivalent to mean payoff games. *International Journal of Algebra and Computation*, 22(01):1250001, February 2012. **B**
- [72] Roger D. Nussbaum. Convexity and log convexity for the spectral radius. *Linear Algebra and its Applications*, 73:59–122, 1986. Department of Mathematics, Rutgers University. **B**
- [73] Michael Joswig and Benjamin Schröter. Parametric shortest-path algorithms via tropical geometry. *Mathematics of Operations Research*, 47(3):2065–2081, 2022. **C.5.1**
- [74] 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, Zach 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, 2019. **F**
- [75] Wes McKinney. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, SciPy '10, pages 51 – 56. SciPy, 2010. **F**
- [76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. **F**
- [77] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. **F**

A Limitations

Although Tropical Attention is out performing in almost all algorithmic tasks, this study was conducted on combinatorial algorithms and we have not yet demonstrated how Tropical transformers can scale to perform on generative and autoregressive next-token prediction language tasks. In particular, the computational and memory overhead introduced by tropical operations and the tropical Hilbert metric could incur nontrivial runtime costs or scaling challenges.

B Tropical Geometry

The most fundamental component of tropical algebraic geometry is the *tropical semiring* $\mathbb{T} := (\mathbb{R} \cup \{-\infty\}, \oplus, \odot)$. The two operations \oplus and \odot , called *tropical addition* and *tropical multiplication* respectively, are defined as follows.

Definition B.1. For $x, y \in \mathbb{R}$, their tropical sum is $x \oplus y := \max\{x, y\}$; their tropical product is $x \odot y := x + y$; the tropical quotient of x over y is $x \oslash y := x - y$.

For any $x \in \mathbb{R}$, we have $-\infty \oplus x = 0 \odot x = x$ and $-\infty \odot x = -\infty$. Thus $-\infty$ is the tropical additive identity and 0 is the tropical multiplicative identity. Furthermore, these operations satisfy the usual laws of arithmetic, namely associativity, commutativity, and distributivity. The set $\mathbb{R} \cup \{-\infty\}$ is therefore a semiring under the operations \oplus and \odot . While it is not a ring since it lacks an additive inverse, one may nonetheless generalize many algebraic objects over the tropical semiring, the study of these, in a nutshell, constitutes the subject of tropical algebra. In order to have a transition from classical arithmetic to tropical arithmetic we need a series of transition maps, which is referred to as *tropicalization*.

Definition B.2. (*The valuation map*) Let $d \in \mathbb{N}$ and write \mathbb{R}^d the field of real numbers. A valuation on \mathbb{R} is a function $\text{val}: \mathbb{R} \rightarrow \mathbb{R} \cup -\infty$ satisfying the following three axioms:

1. $\text{val}(a) = -\infty \iff a = 0$;
2. $\text{val}(ab) = \text{val}(a) + \text{val}(b)$;
3. $\text{val}(a + b) \leq \max\{\text{val}(a), \text{val}(b)\} \forall a, b \in \mathbb{R}$.

One approach to tropical geometry, is to define a tropical variety as a shadow of an algebraic variety that involves logarithmic limit sets. Classically, the *amoeba* of a variety is its image under taking the coordinate-wise logarithm of the absolute value of any point on the variety [14]. The logarithm turns ordinary multiplication into tropical addition:

$$\text{val}(x \odot y) = \text{val}(x) + \text{val}(y), \quad x, y \in \mathbb{R}_{>0},$$

and satisfies the sub-additive inequality $\text{val}(x + y) \leq \max\{\text{val}(x), \text{val}(y)\} + \log 2$. Hence val is a Archimedean log map up to a harmless additive constant. For an input $X \subset \mathbb{R}^d$ we call

$$\text{Trop}(X) := \text{val}(X) \subset \mathbb{T}^d$$

its *tropicalization*. All subsequent reasoning, including attention weight computations, will take place in this max-plus space. When X is a smooth manifold, $\text{Trop}(X)$ is typically a curved domain whose “tentacles” encode asymptotic directions of X . Passing to the max-plus algebra straightens those curves into polyhedral pieces, providing the piecewise-linear structure on which our Tropical Attention operates.

Definition B.3. (*The tropical projective space [1].*) We regard \mathbb{T}^d as a semimodule over the tropical semiring by coordinate-wise operations. Introduce

$$\mathbf{1}_{d+1} := (1, \dots, 1) \in \mathbb{R}^{d+1}, \quad \mathbb{T}^{d+1} := (\mathbb{R} \cup \{-\infty\})^{d+1} \setminus \{-\infty\}^{d+1}.$$

Declare two points $x, y \in \mathbb{T}^{d+1}$ projectively equivalent, written $x \sim y$, if there is a scalar $\lambda \in \mathbb{R}$ such that $y = x + \lambda \mathbf{1}_{d+1}$. The quotient

$$\mathbb{TP}^d := \mathbb{T}^{d+1} / \sim$$

is the tropical projective space. See [1] for more details on tropical geometry.

Every class has a unique representative with maximal coordinate equal to 0, so \mathbb{TP}^d identifies with the standard simplex $\Delta^d := \{w \in \mathbb{R}^{d+1} \mid \max_i w_i = 0\}$. Attention weights produced by the softmax surrogate live in the Euclidean simplex; Tropical Attention will instead output points of Δ^d interpreted tropically, guaranteeing sharp $\arg \max$ behavior.

Definition B.4. (*The tropical Hilbert projective metric.*) For $x := (x_1, \dots, x_{d+1}), y := (y_1, \dots, y_{d+1}) \in \mathbb{T}^{d+1}$ put

$$d_{\mathbb{H}}(x, y) := \left(\max_i (x_i - y_i) \right) - \left(\min_i (x_i - y_i) \right) = \text{diam}(x \oslash y),$$

where $x \oslash y$ denotes the coordinate-wise tropical quotient $(x_1 - y_1, \dots, x_{d+1} - y_{d+1})$ and diam its range.

The metric descends to \mathbb{TP}^d and enjoys two key properties:

1. **Projective invariance.** $d_{\mathbb{H}}(x + c\mathbf{1}_{d+1}, y + c\mathbf{1}_{d+1}) = d_{\mathbb{H}}(x, y)$ for all $c \in \mathbb{R}$.
2. **Non-expansiveness of max-plus-affine maps [70].** Every tropical linear map $A: \mathbb{T}^{d+1} \rightarrow \mathbb{T}^{m+1}$ is 1-Lipschitz: $d_{\mathbb{H}}(Ax, Ay) \leq d_{\mathbb{H}}(x, y)$.

These facts, due to Nussbaum and further developed by Akian–Gaubert, furnish tight robustness guarantees, perturbing the inputs by ϵ in Hilbert distance changes the output of any compositional stack of tropical linear layers by at most ϵ [71, 72].

C Proofs

Proof of lemma 3.1. If ϕ_λ is a valuation map, hence for all $a, b \in \mathbb{R}$, we have

1. $v_\lambda(0) = -\infty$;
2. $v_\lambda(ab) = v_\lambda(a) + v_\lambda(b) - \lambda$;
3. $v_\lambda(a + b) \leq \max\{v_\lambda(a), v_\lambda(b)\}$.

Property (i) is immediate from the definition. For $a, b > 0$, (ii) follows from $\log(ab) = \log a + \log b$:

$$v_\lambda(ab) = \log(ab) - \lambda = (\log a - \lambda) + (\log b - \lambda) + \lambda = v_\lambda(a) + v_\lambda(b) - \lambda.$$

If either factor is non-positive, both sides equal $-\infty$. For (iii), note that when $a, b > 0$ we have

$$\log(a + b) \leq \max\{\log a, \log b\} + \log 2,$$

so subtracting λ preserves the inequality; if $a \leq 0$ or $b \leq 0$ the claim is trivial. Adding back the constant λ to v_λ eliminates the offset in (ii) while leaving (i)–(iii) unchanged, yielding the classical valuation \tilde{v} . Collecting the d coordinate-wise maps gives the vector-valued projection $\phi_\lambda : \mathbb{R}^d \rightarrow \Delta^{d-1}$, which is therefore a valuation map up to the projective (constant-shift) equivalence native to tropical geometry. \square

The main theorem here establishes that MHTA is an expressive tropically universal approximator of max-plus dynamic programming for combinatorial optimization such that every function that can be computed by a finite max-plus circuit admits a realization by a finite-depth MHTA stack. The proof proceeds in three stages. First we show that a single head can act as a tropical max gate. Second, we demonstrate that an H -head block can realize a tropical map by computing finitely many such maxima in parallel. Finally, we prove by structural induction that stacking a finite number of blocks suffices to emulate an arbitrary max-plus circuit. With the first lemma we want to show that a single head can realize a *weighted tropical max gate*.

Lemma C.1 (Head-level Weighted \oplus gate). *Let J be a finite index set and let $\{x_j\}_{j \in J} \subset \mathbb{T}$ and $\{w_j\}_{j \in J} \subset \mathbb{T}$. There exists an attention head $h^* \in [H]$, a query-token index $i^* \in [N] \setminus \{t(j) \mid j \in J\}$, and distinct seq indices $t(j) \in [N]$ such that, after one forward pass, the context returned at i^* is equal to*

$$c_{i^*}^{(h^*)} = \bigoplus_{j \in J} (x_j \odot w_j) = \max_{j \in J} \{x_j + w_j\}. \quad (3)$$

Proof. For a fix h^* and i^* , for every $j \in J$, let's select a distinct token position $t(j)$. Then one can define the *value* vectors by $\mathbf{v}_{t(j)}^{(h^*)} := x_j \odot w_j$ and $\mathbf{v}_r^{(h^*)} := -\infty$ for all $r \notin \{t(j)\}$. To enforce (3) it suffices to make $s_{i^* t(j)}^{(h^*)} = 0$ for $j \in J$ and $s_{i^* r}^{(h^*)} = -\infty$ otherwise, because then $c_{i^*}^{(h^*)} = \bigoplus_{j \in J} (0 \odot (x_j \odot w_j)) = \max_{j \in J} (x_j + w_j)$.

One can write every query / key vector in block form $u = (u^{(1)}, u^{(2)}) \in \mathbb{T}^{d_k-1} \times \mathbb{T}$. Fix arbitrary first blocks $u^{(1)}$ and arrange

$$\mathbf{q}_{i^*}^{(h^*)} = (0, \dots, 0, 0), \quad \mathbf{k}_{t(j)}^{(h^*)} = (0, \dots, 0, 0), \quad j \in J,$$

so that $d_{\mathbb{H}}(\mathbf{q}_{i^*}^{(h^*)}, \mathbf{k}_{t(j)}^{(h^*)}) = 0$ and hence $s_{i^* t(j)}^{(h^*)} = 0$. For every *irrelevant* token $r \notin \{t(j)\}$ set

$$\mathbf{k}_r^{(h^*)} = (0, \dots, 0, -\Gamma_r), \quad \Gamma_r \gg 0,$$

so that the last coordinate differs from that of the query by Γ_r ; consequently $d_{\mathbb{H}}(\mathbf{q}_{i^*}^{(h^*)}, \mathbf{k}_r^{(h^*)}) = \Gamma_r$ and $s_{i^* r}^{(h^*)} = -\Gamma_r$. Choosing Γ_r large enough drives the score to $-\infty$ in the semiring, ensuring that irrelevant tokens do not influence the context. Equation (3) follows. \square

Lemma C.2 (Tropical affine layer). *Let $A \in \mathbb{T}^{M \times N}$ and $b \in \mathbb{T}^M$. Embed $x = (x_1, \dots, x_N) \in \mathbb{T}^N$ as the values of tokens $t(1), \dots, t(N)$ and add one bias token i_b whose value is fixed to 0. There exists an MHTA layer with $H = M$ heads and $d_k = 2$ such that, for each $m \in [M]$,*

$$c_{i_m}^{(m)} = \bigoplus_{j=1}^N (A_{mj} \odot x_j) \oplus b_m,$$

where i_m is the query token of head m .

Proof. For $m \in [M]$ and head $h = m$, we can apply Lemma C.1 with $J = \{1, \dots, N\}$, input x_j and weights A_{mj} to obtain $\bigoplus_j (A_{mj} \odot x_j)$. Let the bias relevant to every head by assigning its key identical to the query, whence $s_{i_m i_b}^{(m)} = 0$ for all m . Then, we give it value b_m in head m alone via $\mathbf{W}_V^{(m)}$. The context becomes the maximum of $\bigoplus_j (A_{mj} \odot x_j)$ and b_m , completing the proof. \square

Definition C.1 (Tropical circuit [15]). A tropical circuit is a finite directed acyclic graph whose source nodes are labelled by variables $z_1, \dots, z_n \in \mathbb{T}$ and whose internal nodes are labelled either by the operation tropical addition $(u, v) \mapsto u \oplus v = \max\{u, v\}$ or by the operation tropical multiplication $(u, v) \mapsto u \odot v = u + v$. The circuit computes a map $f : \mathbb{T}^n \rightarrow \mathbb{T}^m$ whose m outputs are designated sinks. A circuit is layered if every edge points from layer ℓ to layer $\ell + 1$ for some topological layering $\{\mathcal{L}_0, \dots, \mathcal{L}_L\}$. We write $\text{depth}(\mathcal{C}) = L$ and $\text{size}(\mathcal{C}) = |\mathcal{C}|$ for the number of internal gates.

Because tropical multiplication distributes over tropical addition, every such circuit computes a *tropical polynomial*, namely a tropical sum \oplus of finitely many monomials, each monomial being a tropical product \odot (classical summation) of a subset of the indeterminates plus a constant. A *tropical polynomial* in variables $z = (z_1, \dots, z_n)$ has an expression of the form

$$P(z) = \bigoplus_{k=1}^K \left(c_k \odot \bigodot_{j=1}^n z_j^{\odot e_{kj}} \right) = \max_{k \leq K} \left\{ c_k + \sum_{j=1}^n e_{kj} z_j \right\},$$

where $c_k \in \mathbb{T}$ and $e_{kj} \in \mathbb{N}$. Thus P is *already* the maximum of finitely many affine forms in z . Lemma C.2 therefore applies directly.

Theorem C.3 (Single-layer universality for tropical polynomials). Let $P : \mathbb{T}^n \rightarrow \mathbb{T}^m$ be a vector-valued tropical polynomial map whose m coordinates are $P_\ell(z) = \bigoplus_{k \leq K_\ell} (A_{\ell k} \odot z) \oplus b_{\ell k}$. There exists a single MHTA layer with $H = \sum_{\ell=1}^m K_\ell$ heads and $d_k \geq 2$ whose tropical output (the collection of all head contexts before the de-valuation $\psi = \exp$) equals $P(z)$.

Proof. For each output coordinate ℓ one can allocate K_ℓ heads, one per affine term $A_{\ell k} \odot z \oplus b_{\ell k}$. Lemma C.2 shows that affine map in head (ℓ, k) , depositing its value at a fresh query token $i_{\ell k}$. Because the score of an irrelevant head is $-\infty$, the contexts written to those tokens are ignored by all other heads. Finally, putting an aggregation head per output ℓ whose query token reads all tokens $i_{\ell k}$ with *score* 0 and returns their \oplus , namely $\max_k (A_{\ell k} \odot z \oplus b_{\ell k}) = P_\ell(z)$. No de-valuation is applied inside the tropical computation, so the result equals $P(z)$ in the max-plus semiring. \square

Corollary C.3.1 (Depth- L universality). Let $F : \mathbb{T}^n \rightarrow \mathbb{T}^m$ be the output of a layered tropical circuit of depth L . Then, there exists an MHTA stack of L successive layers which, on every $x \in (\mathbb{R}_{>0})^n$, produces

$$\mathbf{C}^{(L)}(x) = F(\text{val}(x)).$$

Proof. We can apply Theorem C.3 to each $P^{(i)}$ in succession, feeding the contexts of layer i (still in tropical form) as the inputs to layer $i + 1$. Because no Euclidean de-valuation occurs after all MHTA layers, the tropical composition is preserved. \square

Theorem C.4 (Simulation of max-plus Dynamic Programs). Let (S, E) be a finite directed acyclic graph with $|S| = N$ nodes and weighted edges $\{w_{uv}\}_{(u,v) \in E} \subset \mathbb{T}$. For $t \in \mathbb{N}$ define

$$d_v(t+1) = \bigoplus_{u: (u,v) \in E} \left(w_{uv} \odot d_u(t) \right), \quad d_v(0) = \delta_{v,v_0},$$

where $v_0 \in S$ is the source node. For every finite horizon T there exists a MHTA of depth T and N heads per layer such that the token values at layer t equal the vector $(d_v(t))_{v \in S}$ for all $t \leq T$.

Proof. If we label the tokens by the vertices of S , at layer t we store $d_v(t)$ in the value field of token v . To obtain $d_v(t+1)$ let head $h = v$ whose query token is v . Then, one can apply Lemma C.1 with index set $J = \{u \mid (u, v) \in E\}$, input scalars $x_u = d_u(t)$ and weights w_{uv} , thereby producing $d_v(t+1)$ as context at token v . Since every head acts on em disjoint query tokens, all $v \in S$ are updated in parallel. Repeating for T layers unrolls the dynamic program, hence layer T realizes the horizon- T value vector. \square

Let $(\Gamma = (V, E))$ be a directed graph with $(|V| = n)$ and tropical weighted adjacency matrix $(D \in \mathbb{T}^{n \times n})$. For $k \geq 0$ the tropical power $D^{\odot k}$ encodes path weights of length k . Fix a horizon $T \in \mathbb{N}$ and set the finite Kleene star

$$D^{*(T)}; :=; \bigoplus_{k=0}^{T-1} D^{\odot k} \in \mathbb{T}^{n \times n}.$$

For $v_0 \in \mathbb{T}^n$ the horizon- T value vector is $v_T = D^{*(T)} \otimes v_0$.

Theorem C.5 (MHTA learns tropical transitive closure). *Given $\Gamma = (V, E)$ with $|V| = n$ and adjacency $D \in \mathbb{T}^{n \times n}$, for every $T \in \mathbb{N}$ there exists an MHTA stack of depth T with n heads per layer and head dimension d_k such that the token values at layer t equal $v_t \in \mathbb{T}^n$ and*

$$v_{t+1}; =; D \otimes v_t \quad (t = 0, 1, \dots, T-1).$$

In particular, the output at layer T equals $v_T = D^{(T)} \otimes v_0$.*

Proof. Indexing tokens by (V) and store $(v_t(u))$ as the value on token (u) at depth (t) , we fix $(v \in V)$ and in layer (t) , we assign head $(h = v)$ whose query is token (v) . Then, we apply Lemma B.1 to the index set $(J_v = u \in V : (u, v) \in E)$ with inputs $(x_u = v_t(u))$ and weights $(w_{uv} = D_{vu})$. The head returns at its query token

$$\bigoplus_{u \in J_v} (w_{uv} \odot x_u); =; \max_{u: (u, v) \in E} D_{vu} + v_t(u); =; (D \otimes v_t)(v); =; v_{t+1}(v).$$

All $(v \in V)$ update in parallel because heads use disjoint query tokens; thus (n) heads suffice in each layer. Iterating over $t = 0, \dots, T-1$ yields (v_T) , and by the path-semiring semantics we have $(v_T = D^{*(T)} \otimes v_0)$. \square

Corollary C.5.1. *In a tropical semiring and under the assumption that no negative cycle exist, shortest paths exist and are cycle-free, so their lengths are realized by paths of at most $(n-1)$ arcs; consequently the infinite Kleene star truncates at $(n-1)$ [73]. Thus, every shortest path uses at most $(n-1)$ arcs, hence*

$$D^*; =; I \oplus D \oplus \dots \oplus D^{\odot(n-1)}; =; D^{*(n)}.$$

Therefore the MHTA stack of with depth $T \geq n-1$ learns $(D^ \otimes v_0)$.*

In the embedding space of MHTA, the feasible domain is a polyhedron in parameter space on which shortest paths exist and the Kleene star is well-defined. The MHTA stack thus implements Bellman steps in the parameter space, thus the combinatorics of feasible solutions are absorbed in the polyhedral stratification. In particular, the map $(v_0, \theta) \mapsto D^{*(T)}(\theta) \otimes v_0$ is a tropical polynomial, and the Tropical Attention has a sufficient ability to learn such maps without recurrence. Hence, in principle, MHTA can learn the closure map itself rather than the step-by-step recurrence.

D Comparison between vanilla attention and Tropical Attention

In this section, we compare the algorithmic view between vanilla attention and Tropical Attention.

Algorithm 1 Comparison between vanilla attention and Tropical Attention

<pre> function ATTENTION($\mathbf{X} : n \times d$) $\mathbf{Q}, \mathbf{K}, \mathbf{V} \leftarrow \text{linear}(\mathbf{X}).\text{chunk}(3)$ $\tilde{\mathbf{A}} \leftarrow \text{einsum}(id, jd \rightarrow ij, \mathbf{Q}, \mathbf{K})$ $\mathbf{A} \leftarrow \text{softmax}(\tilde{\mathbf{A}} / \sqrt{d}, -1)$ $\mathbf{O} \leftarrow \text{einsum}(ij, jd \rightarrow id, \mathbf{A}, \mathbf{V})$ return linear(\mathbf{O}) end function </pre>	<pre> function TROP_ATTENTION($\mathbf{X} : n \times d$) $\mathbf{Q}', \mathbf{K}', \mathbf{V}' \leftarrow \log(\text{ReLU}(\text{linear}(\mathbf{X})))_chunk(3)$ $\lambda \leftarrow \text{Parameter}(\mathbf{N})$ $\mathbf{Q} \leftarrow \mathbf{Q}' - \lambda, \mathbf{K} \leftarrow \mathbf{K}' - \lambda, \mathbf{V} \leftarrow \mathbf{V}' - \lambda$ $\mathbf{Q}_{btd} = \max_j (\mathbf{Q}_{btj} + W_{dj}^{(Q)})$ $\mathbf{K}_{btd} = \max_j (\mathbf{K}_{btj} + W_{dj}^{(K)})$ $\mathbf{V}_{btd} = \max_j (\mathbf{V}_{btj} + W_{dj}^{(V)})$ $\forall i, j : \mathbf{D}_{bij} \leftarrow \max_d (\mathbf{Q}_{bid} - \mathbf{K}_{bjd}) - \min_d (\mathbf{Q}_{bid} - \mathbf{K}_{bjd})$ $\mathbf{S} \leftarrow -\mathbf{D}$ $\forall i, d : \mathbf{C}_{bid} \leftarrow \max_j (\mathbf{S}_{bij} + \mathbf{V}_{bjd})$ $\mathbf{O} \leftarrow \exp(\mathbf{C})$ return linear(\mathbf{O}) end function </pre>
--	---

E Dataset Details

Our evaluation suite covers eleven canonical problems²:

Floyd–Warshall Dataset. Each example is a weighted directed graph with nonnegative edge weights; we compute the all-pairs shortest-path distances with the Floyd–Warshall algorithm and use the resulting distance matrix as the regression target. Inputs are the flattened (zero-filled for missing edges) weight matrix with positional indices.

²Code used to generate data can be found in our public repository: <https://github.com/Baran-phys/Tropical-Attention/blob/main/dataloaders.py>

QuickSelect Dataset. Each example is an unsorted list of integers together with an order statistic k ; the label is a token-wise binary mask marking all positions that contain the k -th smallest value. Inputs provide per-element features (value and k or a normalized rank proxy) in the original list order.

3SUM–Decision Dataset. Each example consists of a list of integers and a target T ; the label is 1 if any three distinct elements sum to T , and 0 otherwise. Inputs present per-token pairs $[x_i, T]$ so the model can condition each element on the common target.

Subset–Sum–Decision Dataset. Each example is a list of integers and a target T ; the label indicates whether some subset sums exactly to T . Inputs again use per-token pairs $[x_i, T]$, and labels are computed by an exact dynamic-programming subset-sum solver.

Balanced Partition Dataset. Given a multiset of integers, the goal is to split it into two subsets with minimal absolute difference of sums; the label is a token-wise 0/1 membership vector for one optimal subset (chosen deterministically). Ground truth is produced via a standard DP over achievable partial sums with a traceback to a canonical solution.

0–1 Knapsack Dataset. Each instance provides item pairs (v_i, w_i) and a capacity C ; the label is a token-wise 0/1 mask for an optimal item set that maximizes total value under the weight budget. We compute ground truth with an exact dynamic-programming solver and repeat C on every token as an input feature.

Fractional Knapsack Dataset. Each instance also provides (v_i, w_i) and a capacity C ; the label is a per-item fraction in $[0, 1]$ from the optimal fractional solution. Ground truth is computed by the canonical greedy algorithm that takes items in decreasing value-to-weight ratio.

Convex Hull Dataset. Each example is a set of 2-D points; the label is 1 for points on the convex hull and 0 otherwise. Hull membership is determined with a standard monotone chain construction.

Strongly Connected Components (SCC) Dataset. Each example is a directed graph; the label is a flattened $n \times n$ matrix where entry (i, j) is 1 iff nodes i and j belong to the same strongly connected component. The graphs are generated Erdős–Rényi. However, to make the dataset more challenging, we also added curvature to the graph by adding communities to the graph. We obtain SCC memberships using a graph library routine and expose adjacency plus positional indices as inputs.

Bin Packing Dataset. Each example supplies item sizes and a single bin capacity; the label is a token-wise 0/1 indicator of whether an item starts a new bin under First-Fit Decreasing applied to the clean sizes. Inputs include each item’s size, the shared capacity, and a normalized position index after sorting.

Minimum Coin Change (0/1) Dataset. Each instance contains a multiset of coin values and a target amount T ; the label is a token-wise 0/1 mask for one optimal solution using the fewest coins (or the all-zero mask if no solution exists). Ground truth is computed with an exact dynamic-programming solver with traceback.

F Training & Evaluation Protocol

This appendix complements the experimental setup outlined in Sec. 4. We focus on the conceptual pipeline. The low-level engineering choices (e.g. logging cadence, file formats) are documented in the public code repository³. The primary packages utilized in constructing our experiment is Pytorch [74], Pandas and Scipy [75], SciKitLearn [76], and Numpy [77]. The basic workflow is described below:

1. **Dataset generation.** For the selected combinatorial task we generate input and output pairs using the hyperparameters in Table 5
2. **Model instantiation.** A shallow Transformer encoder—configured with 1 layer, 2 attention heads and hidden width 64—is equipped with one of three attention mechanisms: *Vanilla*, *Tropical*, or *Adaptive*.
3. **Optimization.** We train for N_{epoch} epochs using AdamW (10^{-3} , constant, no warm-up). We use one NVIDIA Tesla V100 GPU to train each model. Models trained with a sufficiently large batch size (500) training over 10M samples, took approximately 2.5 minutes to train. For more memory intensive graph models, our training time was approximately 45 minutes given small batch sizes of 16. The objective is chosen per-task:
 - BCE with logits – pooled binary tasks,

³<https://github.com/Baran-phys/Tropical-Attention/>

- token-wise BCE,
- mean-squared error – regression tasks.

$N_{\text{epoch}} = 100$

4. **Evaluation.** After training we reload the final checkpoint, generate a new test set, and compute (i) mean loss for regression tasks and (ii) F_1 for classification tasks on the generated test set. We evaluate our models on in-distribution data (data generated using the same hyperparameters as during training) and on out-of-distribution (OOD) data using the hyperparameters described in Table 5 using the OOD protocol described in Section 4. For Length OOD, all models were trained on sequence length of 8 and we evaluated them at sequence length of 64, with the exception of the graph problems (FloydWarshall and SCC), which were evaluated on sequence length of 16. For Perturbative Noise OOD, each input was perturbed with probability 0.5 with a random integer sampled from the task’s perturbative noise range.

Table 5: Training hyperparameters and data ranges for each combinatorial task. Each task was trained with 10M samples, learning rate of 0.0001, input sequence length of 8, and no perturbations. The ranges in the table are used to draw random integer values for the given parameter within the data generation portion of the training.

Dataset	Epochs	Target Range	Weight Range	Value Range	OOD Value Range	Perturbative Noise Range
SubsetSumDecision	100	(1,10)	N/A	(-5,5)	(-20,20)	(10,30)
Knapsack	100	(10,20)	(1,10)	(1,10)	(11,21)	(10,30)
FractionalKnapsack	100	(10,20)	(1,10)	(1,10)	(11,21)	(1,5)
MinCoinChange	100	(10,20)	N/A	(1,10)	(11,21)	(1,5)
Quickselect	100	N/A	N/A	(1,10)	(11,21)	(1,5)
BalancedPartition	100	N/A	N/A	(1,10)	(11,100)	(10,30)
BinPacking	100	(10,30)	N/A	(1,10)	(11,100)	(10,30)
ConvexHull	100	N/A	N/A	(0,10)	(11,21)	(1,5)
ThreeSumDecision	100	(-75,75)	N/A	(-20,20)	(-375,375)	(40,60)
FloydWarshall	100	N/A	N/A	(1,15)	(16,30)	(1,10)
SCC ⁴	100	N/A	N/A	0.001	0.1	N/A

⁴SCC uses a connectivity probability rather than an integer input value, hence the small decimal for Value Ranges and N/A for perturbative noise range. For perturbative noise range the connectivity switches with given perturbation probability.

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 Introduction (Section 1) state exactly the three contributions that are later substantiated: the Tropical Attention mechanism, its theoretical guarantees (Section 3 and Appendix C), and the eleven-task OOD evaluation (Sections 4–5). No claims extending beyond those results are made.

Guidelines:

- The answer NA 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 NA 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: Section 5 acknowledges that our empirical validation is restricted to synthetic combinatorial tasks and that the extra max-plus operations and tropical-Hilbert metric introduce non-trivial memory and runtime overhead, leaving broader domain generalization and scalable efficiency as open issues.

Guidelines:

- The answer NA 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: All assumptions are stated with each theorem in 3. Complete, formal proofs are given in Appendix C, and every lemma/theorem is numbered and cross-referenced from the main text.

Guidelines:

- The answer NA 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: The model, problem sets evaluated, and processes to produce problem set data are outlined in detail in [4](#), [E](#), and [F](#). Additionally, code will be provided in the supplementary materials and is maintained in a github repository which will be made publicly available after the review period.

Guidelines:

- The answer NA 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: All data for experimentation was generated as outlined in [E](#). Code for data production, training, and testing will be provided in the supplementary materials. This code is maintained in a github repository which will be made public following the review period.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 4 provides a high level overview of training and testing. Appendix F provides details on optimization and data. Table 5 provides full training hyperparameter. Further details are provided in the code repository.

Guidelines:

- The answer NA 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: [Yes]

Guidelines:

- The answer NA 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: Appendix F describes the GPU used to train the models and the training time required.

Guidelines:

- The answer NA 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: All experiments use synthetic; no personal, biometric, or sensitive information is processed, and results are released under a permissive license with usage guidelines.

Guidelines:

- The answer NA 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: [No]

Justification: There is no direct or implicit societal impact of our new attention mechanism.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Released assets operate solely on small synthetic integer sequences and graphs; they have no foreseeable dual-use risk beyond existing open-source combinatorial libraries.

Guidelines:

- The answer NA 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: We rely only on well-known open-source Python libraries (e.g. PyTorch, NumPy). Each is cited in Appendix F and its BSD/MIT license is acknowledged in the supplemental material; all other code and models were written by the authors

Guidelines:

- The answer NA 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: [Yes]

Justification: The synthetic datasets and trained checkpoints are accompanied by a README describing generation parameters, file format, and intended use (see supplemental material).

Guidelines:

- The answer NA 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: [NA]

Justification: The study involves no human subjects or crowdsourcing.

Guidelines:

- The answer NA 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: [NA]

Justification: No human subjects research was conducted.

Guidelines:

- The answer NA 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: [NA]

Justification: Large language models are not part of the proposed method; any LLM assistance was restricted to writing support and has no bearing on scientific results.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.