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

Dynamic programming (DP) algorithms for combinatorial optimization problems work with taking maximization, minimization, and classical addition in their recursion algorithms. The associated value functions correspond to convex polyhedra in the max plus semiring. Existing Neural Algorithmic Reasoning models, however, rely on softmax-normalized dot-product attention where the smooth exponential weighting blurs these sharp polyhedral structures and collapses when evaluated on out-of-distribution (OOD) settings. We introduce Tropical attention, a novel attention function that operates natively in the max-plus semiring of tropical geometry. We prove that Tropical attention can approximate tropical circuits of DP-type combinatorial algorithms. We then propose that using Tropical transformers enhances empirical OOD performance in both length generalization and value generalization, on algorithmic reasoning tasks, surpassing softmax baselines while remaining stable under adversarial attacks. We also present adversarial-attack generalization as a third axis for Neural Algorithmic Reasoning benchmarking. Our results demonstrate that *Tropical attention restores the sharp, scale-invariant reasoning absent from softmax*.

1 Introduction

The *tropical semiring* $\mathbb{T} := (\mathbb{R} \cup \{-\infty\}, \max, +)$ (or its "max-plus" variant) replaces ordinary addition by maximum and multiplication by addition [28]. Polynomials over this semiring evaluate to *piecewise-linear*, polyhedral functions. These are the main objects of study in *tropical geometry*, with wide applications across the intersection of matroid theory, combinatorial optimization, Auction theory and algebraic geometry [2, 3, 11, 20, 28], and recently Machine Learning [38, 53, 54, 30, 6, 55]. Because it analyses the entire polyhedral structure of solutions rather than a single Euclidean point, tropical geometry is a 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 functions. They are just recursively constructed

Preprint. Under review.

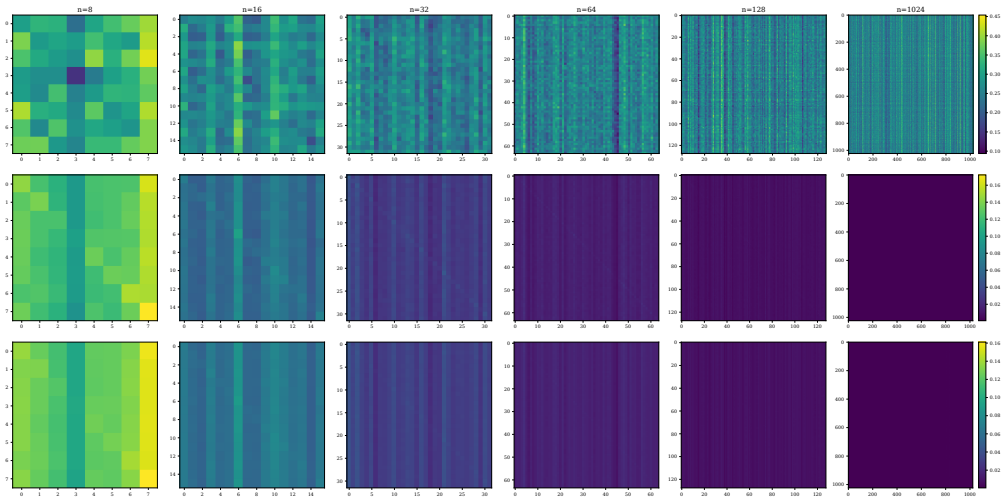

Figure 1: (top) **Tropical attention** with sharp attention maps on learning the **Quickselect** algorithm, showcasing a size-invariance and OOD lengths generalization behavior far beyond training ($8 \rightarrow 1024$). In contrast, both (middle) adaptive-softmax and (bottom) vanilla-softmax heads dilute and disperse as sequence length grows, failing to generalize.

circuits over the corresponding semirings. Each such circuit computes, in a natural way, some polynomial over the underlying semiring. Most known dynamic programming algorithms correspond to circuits over the $(max, +)$ (or $(min, +)$ semirings, or tropical circuits, making the DP update step effectively linear within this algebraic framework [22]. Therefore, DP computations inherently operate tropically, propagating these polyhedral forms [12]. Algorithms like Floyd-Warshall explicitly manifest this, essentially performing tropical matrix operations to explore the facets of a polyhedral solution space [21]. This inherent reliance on sharp, tropical operations poses a challenge for vanilla Attention operation where it has to execute a combinatorial algorithm[48]. Some notable applications are constructing the combinatorial structures of a particular type [19], enumeration [16], and dynamic programming [17, 10].

Vanilla Transformers[47] use softmax-normalized dot-product attention, which in Euclidean space yields smooth, quadratic decision boundaries. This smoothness blurs the hard $\arg\max$ / $\arg\min$ structure that DPs rely on, that is approximating a true maximum. This phenomena where as the length input vector grows, the resulting probability distribution becomes increasingly flat has several names such as dispersion [49] or attention fading [33]. 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 classical algorithmic benchmarks. For non-algorithmic reasoning tasks, even though injecting positional information [45, 42, 9] can alleviate the length extrapolation issue, we believe that the core of the issue lies within the this softmax function.

As a result, we propose *Tropical Attention*, a novel attention function that, maps Euclidean input information to the tropical semiring, perform information routing there by tropical geometric operations, then maps the result back to Euclidean space so that subsequent Transformer blocks remain unchanged. Since, the max-plus aggregation is 1-Lipschitz and piecewise-linear, Tropical Attention preserves the polyhedral structure of the underlying DP while inheriting the projective Hilbert metric that captures shortest-path dynamics [20]. As a result, we prove that multi-head Tropical attention captures the target class of max–plus piecewise-linear maps that arise in tropical circuits, combinatorial optimization and dynamic programming. Our expressivity theorems build directly on this correspondence, showing that multi-head Tropical Attention simulates such circuits without super-polynomial blow-ups.

Our contributions in this paper goes as follows:

1. **A novel attention mechanism** Introducing Tropical Transformer with the Tropical attention mechanism operating in the max-plus arithmetic $(\max, +)$, with a theoretical guarantee.

2. **Theoretical study on expressivity** Demonstrating that that max–plus dynamic programming lies within the expressive envelope of Tropical attention, and proving that multi-head Tropical attention can simulate any DP-like algorithm, laying theoretical groundwork for future bridges between discrete optimization and reasoning models.

3. **Extensive empirical evaluation.** On eleven canonical combinatorial tasks—FLOYD–WARSHALL, QUICKSELECT, 3SUMDECISION, BALANCED-PARTITION, CONVEXHULL, SUBSETSUMDECISION, $0\backslash1$ and fractional KNAPSACK,STRONGLYCONNECTEDCOMPONENTS, BINPACKING, and MINCOINCHANGE—Tropical transformer achieve state-of-the-art (SOTA) out-of-distribution (OOD) generalization in length and value scale and exhibit superior adversarial robustness.

The remainder of the paper develops these ideas. Section 2 formally define OOD generalization, tropical algebra, Softmax attention and prior work; Section 3 formalizes our mechanism and theoretical results; Sections 4– 5 detail the experimental study; Section 5 discusses the results and implications; and Section 6 concludes and suggests future directions.

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 [7] notations formally, let $\mathcal{X}$ denote the feature space and $\mathcal{Y}$ the label set. A model $h : \mathcal{X} \to \mathcal{Y}$ 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, $\mathcal{R}_{D_{\mathrm{te}}}(h) := \mathbb{E}_{(x,y)\sim D_{\mathrm{te}}}[\ell(h(x), y)]$, where ℓ is a loss function. Its empirical estimate on a finite sample S drawn from D_{te} (i.e., $S \sim D_{\mathrm{te}}^{|S|}$) 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 $\mathcal{R}_{D_{\mathrm{te}}}(h)$ remains comparable to its in-distribution risk $\mathcal{R}_{D_{\mathrm{tr}}}(h)$, 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:

Length Generalization. Both distributions draw their numerical entries from the same range but the test sequences are strictly longer, $D_{\mathrm{te}}(\mathcal{X}) \subsetneq \left(\mathbb{R}^{>0}\right)^{n_{\max}}$ with $n_{\max} > n_{\mathrm{tr}}$. 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.

Value generalization. The two distributions share the same support with respect to sequence length but $\mathrm{supp}\big(D_{\mathrm{te}}(\mathcal{X})\big)$ contains magnitudes never encountered during training, i.e. $\mathrm{supp}\big(D_{\mathrm{te}}(\mathcal{X})\big) \setminus \mathrm{supp}\big(D_{\mathrm{tr}}(\mathcal{X})\big) \neq \varnothing$. 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.

Adversarial-attack generalization. Adversarial attacks aim to cause a model to make mistakes with perturbations (adversarial samples) or predefined patterns (backdoor triggers). Here D_{te} is obtained from D_{tr} by an ℓ_p-bounded, perturbation map $\mathcal{A} : \mathcal{X} \to \mathcal{X}$: $x_{\mathrm{adv}} = \mathcal{A}(x)$ with $\|x_{\mathrm{adv}} - x\|_p \leq \varepsilon$. 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 adversarial reliability of Neural Algorithmic Reasoning models are very important for many real-world systems, especially for cryptographic schemes [41].

Length, value, and adversarial generalization stress complementary facets of algorithmic competence[34, 29]. Thus, a model as a true reasoning circuit[27, 10] 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 $\mathbf{X} = \big[\mathbf{x}_1, \dots, \mathbf{x}_N\big]^\top \in \mathbb{R}^{N \times d_x}$, 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,

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

where the $\operatorname{softmax}$ 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 $\tau \to 0$ 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 it's success in many tasks [47, 8], its geometric and numerical properties are ill-suited to algorithmic reasoning[7, 44]. 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 [49, 33]. 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 [31, 37]. The attention kernel's inability to sharpen with scale is a primary culprit.
2. **Sensitivity to small perturbations.** Because $\operatorname{softmax}(z) \propto e^z$, 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 $\mathbf{Q}\mathbf{K}^\top/\tau$ by $\mathcal{O}(\log N)$ may invert the ranking of two tokens, propagating an $\mathcal{O}(1)$ error to downstream activations [52, 25]. This ℓ_∞-fragility persists even after common stabilisers such as temperature scaling or normalization layers [52].
3. **Mismatch with polyhedral decision boundaries.** In a combinatorial optimizations the value function is a tropical polynomial—piecewise linear with faces aligned to coordinate hyperplanes [20, 23]. The quadratic forms generated by Euclidean dot products carve the domain into spherical caps [35] 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 $\tau \to 0$ the Jacobian of the softmax grows like τ^{-1}, causing gradient explosion/vanishing [24]. Careful schedule tuning or gradient clipping becomes mandatory [52], adding hyper-parameter overhead.

2.3 Tropical Geometry

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

Definition 2.1. *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 2.2. *(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* $\mathrm{val}\colon \mathbb{R} \to \mathbb{R} \cup -\infty$ *satisfying the following three axioms:*

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

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 coordinatewise logarithm of the absolute value of any point on the variety [12]. The logarithm turns ordinary multiplication into tropical addition:

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

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

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

its *tropicalization*. All subsequent reasoning, including attention weight computations, will take place in this max-plus space. When X is a smooth manifold, $\mathrm{Trop}(X)$ 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 2.3. *(The tropical projective space [28].) We regard $\mathbb{T}^d$ as a semimodule over the tropical semiring by coordinate-wise operations. Introduce*

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

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*

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

is the tropical projective space. *See [28] 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 2.4. *(The tropical Hilbert projective metric.) For $x := (x_1, \ldots, x_{d+1}), y := (y_1, \ldots, y_{d+1}) \in \mathbb{T}^{d+1}$ put*

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

where $x \oslash y$ denotes the coordinate-wise tropical quotient $(x_1 - y_1, \ldots, x_{d+1} - y_{d+1})$ 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 [46].** Every tropical linear map $A : \mathbb{T}^{d+1} \to \mathbb{T}^{m+1}$ is 1-Lipschitz: $d_{\mathbb{H}}\big(Ax, Ay\big) \leq d_{\mathbb{H}}(x, y)$.

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 [1, 36].

2.4 Neural Algorithmic Reasoning

Bridging symbolic algorithms and differentiable models has become has been established officially under the name of *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 some early work [50] that aimed to demonstrate the applicability of Graph Neural Networks (GNNs) to approximate classical algorithm, the community has then developed and expanded further in different directions [26, 43, 13, 14, 4, 51, 44, 5, 18]. A fundamental objective of NAR is to achieve robust out-of-distribution (OOD) generalization. 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 a universality guarantees for an attention layer within the NAR agenda.

For combinatorial tasks in particular, dynamic-programming recurrences over the $(\max, +)$ semiring can be seen as shallow tropical circuits; conversely, every tropical circuit induces a DP on a weighted Directed acyclic graph (DAG). This equivalence was established for mean-payoff games [1, 36], while it was established [23] that depth and size lower bounds for tropical circuits and their DP relations. Our expressivity theorems build directly on this correspondence, showing that multi-head Tropical Attention captures such circuits without super-polynomial blow-ups.

Recent work only tried to quantify NAR failures when test sequences are longer [34, 29], numerically larger [7], but not adversarially perturbed scenarios. Adversarial perturbations itself is also important 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 *adversarial-attack generalization* as a third pillar for NAR benchmarking and show that our Tropical attention demonstrates systematic gains across all three axes.

Based on the past related works, one can establish a demand for an attention mechanism that (i) respects the underlying algebraic and geometric structure of combinatorial algorithms, (ii) mitigates softmax dispersion that hampers OOD generalization, and (iii) delivers OOD and robustness benefits. Tropical attention positions itself at this intersection, drawing on a decade of tropical-geometric insights to advance neural algorithmic reasoning.

3 Tropical Attention

Tropical Attention mechanism arises from a parallel with classical combinatorial algorithms, that is information exchange is governed less by raw magnitudes than by order statistics—maxima, minima, and interval widths, all of which live naturally in the max-plus semiring. We therefore posit that the the dot-product kernel design choices hinder scale generalization in methods that incorporate vanilla attention mechanism. In this section, our goal is to replace the *dot–product*, Softmax-based kernel of vanilla self-attention with an operation that (i) preserves the piecewise-linear geometry of combinatorial problems, and (ii) inherits the *1–Lipschitz* robustness of tropical linear maps. We therefore project queries, keys, and values to the max–plus semiring, compute attention weights with the tropical Hilbert metric, aggregate by a tropical matrix–vector product, and finally map the result back to Euclidean space so that the rest of the Euclidean 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 OOD generalizations tasks.

Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be the embedding of the input via a learnable affine feed-forward neural network. We define the *tropicalization map* by going to the amoeba representation of the input followed by a learnable map,

$$\Phi_\lambda(\mathbf{X}) \;=\; \log\big(\mathrm{ReLU}(\mathbf{X})\big) \;-\; \lambda, \tag{2}$$

where the learnable vector $\lambda \in \mathbb{R}^N$. The constant shift enforces $\max_i \phi_\lambda(\mathbf{x})_i = 0$, so the output of ϕ_λ always lies in the tropical simplex, $\Delta^{\,d-1} := \big\{\, z \in \mathbb{R}^d \big|\, \max_i z_i = 0\big\}$, where every vector is projectively equivalent to exactly one point in the tropical simplex.

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

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

where $\boldsymbol{\phi_\lambda}$ is a (projective) valuation map. Hence the shifted map $\widetilde{v}(x) = v_\lambda(x) + \lambda = \log(\mathrm{ReLU}(x))$ is a non-Archimedean valuation in the classical sense, and $\Phi_\lambda : \mathbb{R}^{N\times d} \to (\mathbb{R} \cup \{-\infty\})^{N\times d}$ is a matrix-valued valuation modulo tropical scalars; its image lies in the tropical simplex.

After mapping each input token to tropical projective space, $\mathbf{Z} = \phi_\lambda(\mathbf{X}) \in \mathbb{TP}^{N\times d}$, 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 $\mathbf{Q}^{(h)} = \mathbf{Z} \;\odot\; \mathbf{W}_Q^{(h)\top}, \mathbf{K}^{(h)} = \mathbf{Z} \;\odot\; \mathbf{W}_K^{(h)\top}, \mathbf{V}^{(h)} = \mathbf{Z} \;\odot\; \mathbf{W}_V^{(h)\top}$ 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 2.4, we will have the* tropical attention score

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

Thereafter, the head outputs are aggregated via tropical matrix–vector product,

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

In the end, 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, $\mathbf{H} = \big[\psi(\mathbf{C}^{(1)}) \parallel \ldots \parallel \psi(\mathbf{C}^{(H)})\big] \in \mathbb{R}^{N\times d}$.

Why Tropical Attention? Every operation inside MHTA, is piecewise linear, hence the entire network computes a tropical polygonal map whose cells are polyhedral cut out by hyperplanes. This is aligned with classical DP recurrences, whose value function is itself a tropical polynomial. Training a neural network with Tropical attention therefore starts from a hypothesis space that already mirrors the solution structure of a combinatorial algorithm. By contrast, Euclidean soft-max attention inserts an exponential map, blurring the sharp decisions on their input data. Moreover, since every intermediate representation of MHTA lies in the projective simplex Δ^{d-1}, going through $\arg\max$ is therefore well-defined (no equal maxima except on a set of measure 0), is stable by global scaling meaning that shifting the entire vector by $\lambda \in \mathbb{R}$ does not alter which index attains the maximum. Whereas classical transformers modulate the entropy–versus–sharpness trade-off through a temperature parameter in the softmax; MHTA sharpness is built in and temperature-free.

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 nothing more than recursively-defined tropical circuits of this kind; consequently, lower bounds for tropical circuits translate directly into limits for such DP schemes. Tropical circuits compute by alternating $\max$-gates with $+$-gates (tropical multiplication). An MHTA head can also be interpreted as a single tropical gate. A single head implements the composite transformation $(u, v) \longmapsto \max_j\{\, S_{ij} + v_j \,\}$, 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. Stacking layers composes tropical gates as a DP table composes its local recurrences; training 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 Multi-Head Tropical Attention, we prove that it is a universal approximator of max-plus dynamic programming for combinatorial optimization (Theorem A.3, Corollary A.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*

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

For every finite horizon $T \in \mathbb{N}$ there exists a MHTA network of *depth* T, using N *heads per layer* and no additional width blow-up, whose token values at layer t equal the DP state vector $(d_v(t))_{v\in S}$ for all $0 \le t \le T$. In particular, the network outputs the optimal max–plus value function $d_\bullet(T)$ after T layers. Hence, without any architectural restriction MHTA captures the target class of max–plus piecewise-linear maps that arise in tropical circuits, combinatorial optimization and dynamic programming. The result is a tight capacity argument that no super-polynomial blow-up in width or depth is required to embed combinatorial algorithmic reasoning into attention.

4 Experiments

We evaluate Tropical transformers on eleven canonical combinatorial tasks. For every task we measure both the in-distribution performance and three complementary forms of out-of-distribution (OOD) generalization: Length OOD (longer inputs), Value OOD (unseen magnitudes), and Adversarial Attack (perturbed inputs). A procedure to compare between vanilla attention and Tropical attentions is described in Appendix B. All datasets, generation scripts, and OOD protocols are described in Appendix C and D. 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 [49]. **Tropical**: Our proposed transformer in which every attention block is replaced by Multi-Head Tropical Attention (MHTA). To ensure a fair comparison, all three 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: (1) **Length OOD** – inputs drawn from the same value range but with longer input sequence lengths. (2) **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 $[-10, 10]$). (3) **Adversarial 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.

Combinatorial Tasks Our evaluation suite covers eleven canonical problems:

1. **Floyd–Warshall**: predict all–pairs shortest-path distances in a weighted digraph (regression);
2. **QuickSelect**: return a one-hot mask of the k-th smallest element in an unsorted list (token-wise classification);
3. **3SUM-Decision**: decide if any triple of integers sums to a target value (binary decision);
4. **Subset-Sum-Decision**: decide if any subset sums to given target sum (binary decision);
5. **Balanced Partition**: output minimum absolute difference between two subset sums (regression);
6. **0–1 Knapsack**: compute the maximum value achievable under a weight budget with indivisible items (regression);
7. **Fractional Knapsack**: same objective with items divisible (regression);
8. **Convex Hull**: label planar points that lie on the convex envelope (pointer classification);
9. **Strongly Connected Components (SCC)**: label each node pair as connected/not-connected in an undirected graph (pairwise classification);
10. **Bin Packing**: predict number of bins used by first-fit-decreasing for given capacity (regression);
11. **Min Coin Change**: return the fewest coins required to make a target amount with a given currency system (regression).

Table 1: Out-of-distribution regression performance (MSE) for Vanilla, Adaptive, and Tropical models across three OOD scenarios.

Dataset	Length OOD			Value OOD			Adversarial Attack		
	Vanilla	Adaptive	Tropical	Vanilla	Adaptive	Tropical	Vanilla	Adaptive	Tropical
BalancedPartition	13.48	4.19	**0.25**	34.45	**5.35**	5.80	13.28	2.83	**0.28**
BinPacking	136.50	129.82	**95.36**	0.56	24.81	**0.20**	0.24	10.20	**0.12**
FloydWarshall	37.57	38.09	**20.08**	81.44	**58.17**	59.41	48.87	39.86	**29.44**
FractionalKnapsack	125	118	**117**	394	309	**168**	9.12	8.56	**5.58**
Knapsack	41	37	**24**	32	45	**23**	92	55	**27**
MinCoinChange	0.84	0.42	**0.19**	1.06	0.67	**0.21**	1.36	0.98	**0.67**

Table 2: Out-of-distribution test Micro-F_1 for Vanilla, Adaptive, and Tropical models across three OOD scenarios.

Algorithm	Length OOD			Value OOD			Adversarial Attack		
	Vanilla	Adaptive	Tropical	Vanilla	Adaptive	Tropical	Vanilla	Adaptive	Tropical
ConvexHull	45%	49%	**95%**	21%	21%	**34%**	92%	93%	**98%**
Quickselect	3%	17%	**68%**	0%	0%	**44%**	18%	22%	**50%**
SCC	40%	52%	**71%**	82%	65%	**85%**	22%	**31%**	25%
SubsetSum	18%	30%	**80%**	30%	38%	**72%**	2%	3%	**85%**
3SUM	81%	80%	**82%**	28%	**30%**	25%	60%	62%	**66%**

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 2. 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 structure of the combinatorial algorithm.

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. Figure 2 shows a modified attention head for Quickselect, a problem that requires a sharp $argmax/argmin$ classification. This visualization 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. We see that the attention on both softmax models quickly dilute/disperse as sequence length grows OOD while the tropical attention maintains focus.

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 the structure of the problem vice the specific training data.

Limitations Although Tropical attention is out performing in almost all algorithmic tasks, this study was conducted on synthetic combinatorial algorithms and we have not yet demonstrated how Tropical Transformers can scale to perform on other reasoning domains such as natural language or vision. In particular, the computational and memory overhead introduced by max-plus operations and the tropical Hilbert metric could incur nontrivial runtime costs or scaling challenges.

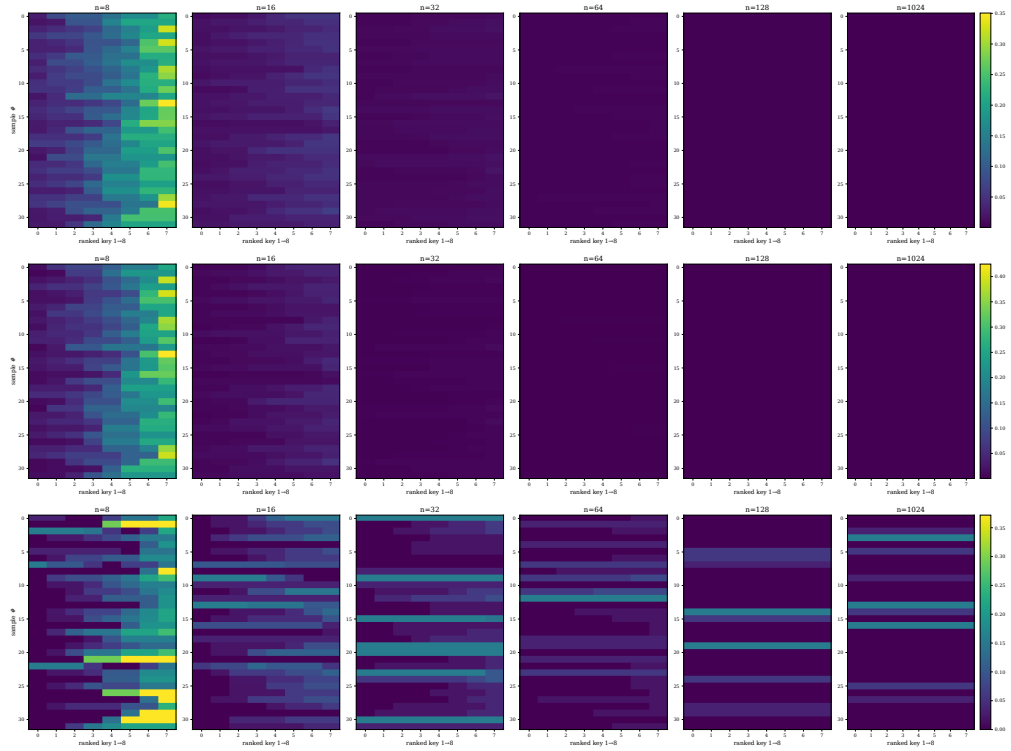

Figure 2: Stacked attention head representations for Quickselect under (a) Vanilla, (b) Adaptive, and (c) **Tropical** models. Each model was trained on length 8 sequences and was evaluated from Left to Right on length 16 to 1024 sequences. Each image was generated by a batch of 32 inputs. The columns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

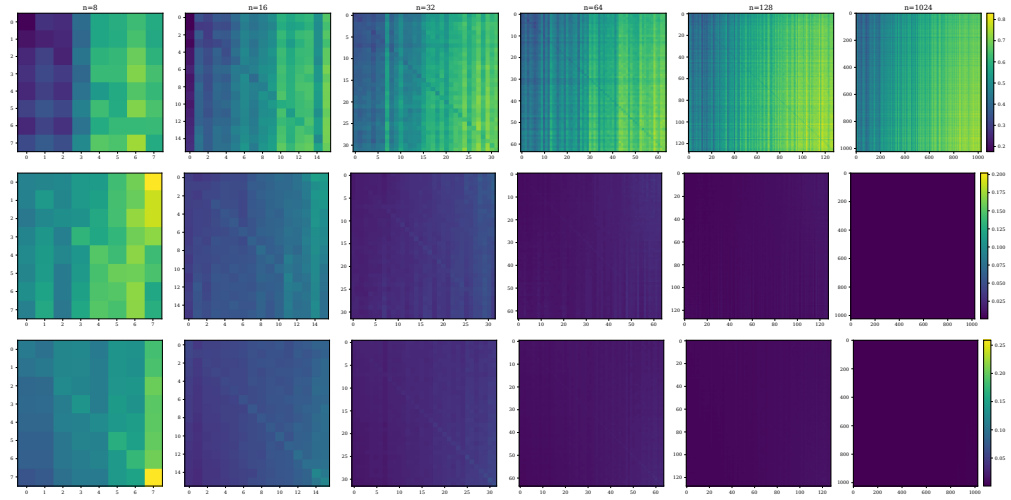

Figure 3: (top) **Tropical attention** with sharp attention maps on learning the *Knapsack algorithm*, showcasing a size-invariance and OOD lengths generalization behavior far beyond training ($8 \rightarrow 1024$). In contrast, both (middle) adaptive-softmax and (bottom) vanilla-softmax heads dilute and disperse as sequence length grows, failing to generalize.

6 Conclusion

We introduced *Tropical Attention*, a novel transformer module that replaces the softmax-normalized dot-product with an idempotent $\max/+$ (tropical) normalization. From a theoretical standpoint, we proved that Tropical Attention enjoys universal approximation properties on tropical polynomials and simulates max–plus tropical circuits that naturally align with dynamic programming algorithms in combinatorial optimization (Theorem A.3, Corollary A.3.1, and Theorem 3.2). Such an expressiveness result is critical for reasoning engines, where they have to generalize beyond the distribution they are trained on. Empirically, across eleven classic optimization problems, our Tropical transformers achieved SOTA out-of-distribution generalization on both sequence length and input-value scaling, and delivered markedly stronger adversarial robustness than Euclidean counterparts.

These findings carry an important message for both neural algorithmic reasoning and Large Language Model communities: **we demonstrate that tropical geometric extension beyond softmax not only enrich the algorithmic power of attention mechanisms but also yield 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 resoning 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] 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.

[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] 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.

[5] 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.

[6] Marie-Charlotte Brandenburg, Georg Loho, and Guido Montúfar. The real tropical geometry of neural networks. *arXiv preprint arXiv:2403.11871*, 2024.

[7] Artur Back de Luca, George Giapitzakis, Shenghao Yang, Petar Veličković, and Kimon Fountoulakis. Positional attention: Expressivity and learnability of algorithmic computation, 2025.

[8] 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.

[9] Shaoxiong Duan, Yining Shi, and Wei Xu. From interpolation to extrapolation: Complete length generalization for arithmetic transformers, 2024.

[10] 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.

[11] Alex Fink and Felipe Rincón. Stiefel tropical linear spaces, 2015.

[12] 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.

[13] Dobrik Georgiev Georgiev, Danilo Numeroso, Davide Bacciu, and Pietro Lio. Neural algorithmic reasoning for combinatorial optimisation. In *The Second Learning on Graphs Conference*, 2023.

[14] 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.

[15] 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.

[16] Baran Hashemi, Roderic Guigo Corominas, and Alessandro Giacchetto. Can transformers do enumerative geometry? In *The Thirteenth International Conference on Learning Representations*, 2025.

[17] 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.

[18] 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.

[19] Yunhui Jang, Dongwoo Kim, and Sungsoo Ahn. Graph generation with k^2-trees. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Michael Joswig. *Essentials of Tropical Convexity*. American Mathematical Society, 2021.

[21] Michael Joswig and Benjamin Schröter. Parametric shortest-path algorithms via tropical geometry. *Mathematics of Operations Research*, 47(3):2065–2081, August 2022.

[22] Stasys Jukna. *Lower Bounds for Tropical Circuits and Dynamic Programs*, volume 57. Springer Science and Business Media LLC, October 2014.

[23] Stasys Jukna. Tropical circuit complexity. *Limits of Pure Dynamic Programming/by Stasys Jukna.-*, 2023.

[24] 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.

[25] Gihyun Kim, Juyeop Kim, and Jong-Seok Lee. Exploring adversarial robustness of vision transformers in the spectral perspective, 2023.

[26] 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.

[27] Xihan Li, Xing Li, Lei Chen, Xing Zhang, Mingxuan Yuan, and Jun Wang. Circuit transformer: A transformer that preserves logical equivalence, 2025.

[28] Diane Maclagan and Bern Sturfels. *Introduction to Tropical Geometry*. American Mathematical Society, 2015.

[29] 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.

[30] Petros Maragos, Vasileios Charisopoulos, and Emmanouil Theodosis. Tropical geometry and machine learning. *Proceedings of the IEEE*, 109(5):728–755, 2021.

[31] 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.

[32] 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.

[33] Ken M. Nakanishi. Scalable-softmax is superior for attention, 2025.

[34] Robert R. Nerem, Samantha Chen, Sanjoy Dasgupta, and Yusu Wang. Graph neural networks extrapolate out-of-distribution for shortest paths, 2025.

[35] Stefan K. Nielsen, Laziz U. Abdullaev, Rachel S. Y. Teo, and Tan M. Nguyen. Elliptical attention, 2024.

[36] 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.

[37] Euan Ong and Petar Veličković. Learnable commutative monoids for graph neural networks, 2022.

[38] Kurt Pasque, Christopher Teska, Ruriko Yoshida, Keiji Miura, and Jefferson Huang. Tropical decision boundaries for neural networks are robust against adversarial attacks, 2024.

[39] 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.

[40] 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.

[41] 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.

[42] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.

[43] Gleb Rodionov and Liudmila Prokhorenkova. Neural algorithmic reasoning without intermediate supervision. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[44] Gleb Rodionov and Liudmila Prokhorenkova. Discrete neural algorithmic reasoning, 2025.

[45] Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer, 2022.

[46] Roan Talbut, Daniele Tramontano, Yueqi Cao, Mathias Drton, and Anthea Monod. Probability metrics for tropical spaces of different dimensions, 2024.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[48] Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7):100273, July 2021.

[49] Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. softmax is not enough (for sharp out-of-distribution), 2024.

[50] Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. Neural execution of graph algorithms. In *International Conference on Learning Representations*, 2020.

[51] Kaijia Xu and Petar Veličković. Recurrent aggregators in neural algorithmic reasoning. In *The Third Learning on Graphs Conference*, 2024.

[52] Hao Xuan, Bokai Yang, and Xingyu Li. Exploring the impact of temperature scaling in softmax for classification and adversarial robustness, 2025.

[53] 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.

[54] Ruriko Yoshida, Leon Zhang, and Xu Zhang. Tropical principal component analysis and its application to phylogenetics, 2017.

[55] 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.

A Proofs

A.1 Proof of lemma 3.1

Proof. 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 $\log(ab) = \log a + \log b$:

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

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 $\widetilde{v}$. Collecting the d coordinate-wise maps gives the vector-valued projection $\phi_\lambda : \mathbb{R}^d \to \Delta^{d-1}$, which is therefore a valuation map up to the projective (constant-shift) equivalence native to tropical geometry. □

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 A.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*

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

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 $c_{i^*}^{(h^*)} = \bigoplus_{j\in J}\big(0 \odot (x_j \odot w_j)\big) = \max_{j\in J}(x_j + w_j)$.

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

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

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

$$\mathbf{k}_r^{(h^*)} = \big(0, \ldots, 0,\, -\Gamma_r\big), \qquad \Gamma_r \gg 0,$$

so that the last coordinate differs from that of the query by Γ_r; consequently $d_{\mathbb{H}}\big(\mathbf{q}_{i^*}^{(h^*)}, \mathbf{k}_r^{(h^*)}\big) = \Gamma_r$ 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. □

Lemma A.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 $m \in [M]$,*

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

where i_m is the query token of head m.

Proof. For $m \in [M]$ and head $h = m$, we can apply Lemma A.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. □

Definition A.1 (Tropical circuit [22])**.** *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 $f : \mathbb{T}^n \to \mathbb{T}^m$ 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* $\mathrm{depth}(\mathcal{C}) = L$ *and* $\mathrm{size}(\mathcal{C}) = |\mathcal{C}|$ *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

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

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 A.2 therefore applies directly.

Theorem A.3 (Single–layer universality for tropical polynomials)**.** *Let $P : \mathbb{T}^n \to \mathbb{T}^m$ be a vector-valued tropical polynomial map whose m coordinates are $P_\ell(z) = \bigoplus_{k\le K_\ell} (A_{\ell k} \odot z) \oplus b_{\ell k}$. There exists a* single MHTA *layer with $H = \sum_{\ell=1}^{m} K_\ell$ heads and $d_k \ge 2$ 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 A.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. □

Corollary A.3.1 (Depth–L universality)**.** *Let $F : \mathbb{T}^n \to \mathbb{T}^m$ 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*

$$\mathbf{C}^{(L)}(x) \;=\; F\big(\mathrm{val}(x)\big).$$

Proof. We can apply Theorem A.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. □

Theorem A.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*

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

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 $\big(d_v(t)\big)_{v\in S}$ for all $t \le T$.*

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 A.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. □

B 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

function ATTENTION($\mathbf{X} : n \times d$)
$\mathbf{Q}, \mathbf{K}, \mathbf{V} \leftarrow \mathsf{linear}(\mathbf{X}).\mathsf{chunk}(3)$
$\widetilde{\mathbf{A}} \leftarrow \mathsf{einsum}(id,\ jd\ \rightarrow\ ij, \mathbf{Q}, \mathbf{K})$
$\mathbf{A} \leftarrow \mathsf{softmax}(\widetilde{\mathbf{A}}/\sqrt{d},\ -1)$
$\mathbf{O} \leftarrow \mathsf{einsum}(ij,\ jd\ \rightarrow\ id, \mathbf{A}, \mathbf{V})$
return $\mathsf{linear}(\mathbf{O})$
end function

function TROP_ATTENTION($\mathbf{X} : n \times d$)
$\mathbf{Q}', \mathbf{K}', \mathbf{V}' \leftarrow \log(\mathrm{ReLU}(\mathsf{linear}(\mathbf{X})))\_\mathrm{chunk}(3)$
$\lambda \leftarrow \mathrm{Parameter}(\mathbf{N})$
$\mathbf{Q} \leftarrow \mathbf{Q}' - \lambda,\ \mathbf{K} \leftarrow \mathbf{K}' - \lambda,\ \mathbf{V} \leftarrow \mathbf{V}' - \lambda$
$\mathbf{Q}_{btd} = \max_j(\mathbf{Q}_{btj} + W^{(Q)}_{dj})$
$\mathbf{K}_{btd} = \max_j(\mathbf{K}_{btj} + W^{(K)}_{dj})$
$\mathbf{V}_{btd} = \max_j(\mathbf{V}_{btj} + W^{(V)}_{dj})$
$\forall i, j:\ \mathbf{D}_{bij} \leftarrow \max_d(\mathbf{Q}_{bid} - \mathbf{K}_{bjd}) - \min_d(\mathbf{Q}_{bid} - \mathbf{K}_{bjd})$
$\mathbf{S} \leftarrow -\mathbf{D}$
$\forall i, d:\ \mathbf{C}_{bid} \leftarrow \max_j(\mathbf{S}_{bij} + \mathbf{V}_{bjd})$
$\mathbf{O} \leftarrow \exp(\mathbf{C})$
return $\mathsf{linear}(\mathbf{O})$
end function

C Dataset Details

Floyd–Warshall Dataset The Floyd–Warshall dataset presents the all-pairs shortest-path problem as a combinatorial regression task. During training, each example is a graph of size n with nonnegative integer edge weights. We compute the full distance matrix via the Floyd–Warshall algorithm, replace unreachable pairs with a large finite value, and flatten both weights and distances into input–output pairs. Out-of-distribution evaluation uses larger graphs, broader weight intervals, and randomly perturbed input.

QuickSelect Dataset The QuickSelect dataset frames the search for the k-th smallest element as a combinatorial classification challenge. During training, each example presents an unsorted list of n integers alongside the fixed order statistic k. The model's task is to predict a binary mask marking all positions in the original list that hold the k-th smallest value. To mimic real-world uncertainty. For out-of-distribution evaluation, list lengths are increased, value ranges broadened, and randomly perturbed inputs.

Three–Sum Decision Dataset The Three–Sum Decision dataset was constructed as a classification task. During training, each example begins with a list of n integers. We then sort the list and pair each element with the target sum T, presenting the model with a sequence of input tokens $[x_i, T]$. The label is a binary decision: 1 if any three distinct elements sum to T, and 0 otherwise. Out-of-distribution evaluation tests models on larger list lengths (larger n), wider value ranges each element can be, and randomly perturbed input data.

Balanced-Partition Dataset Each example begins with a sorted list of n integers drawn from a fixed interval; we compute the minimum absolute difference between the sums of two complementary subsets on this clean data. Out-of-distribution evaluation increases list lengths, broadens the integer range, and introduces sparse input perturbations.

Convex-Hull Dataset Each example starts with a set of n two-dimensional points sampled uniformly; we compute the convex hull and label each point as 1 if it lies on the hull and 0 otherwise. Out-of-distribution evaluation uses larger point clouds, wider coordinate ranges, and sparse input perturbations.

Subset-Sum-Decision Dataset Each sample is a sorted list of n integers together with a target T, and the label indicates whether any subset sums exactly to T on the clean data. Out-of-distribution evaluation increases list lengths, broadens both value and target ranges, and adds sparse perturbations.

$0/1$ **Knapsack Dataset** Each example consists of a sorted sequence of (v_i, w_i) item pairs and a capacity C, from which we compute the maximum achievable value under the capacity constraint on pristine inputs. Out-of-distribution evaluation uses longer item lists, larger value and weight ranges, and sparse perturbations.

Fractional-Knapsack Dataset Each case comprises a list of (v_i, w_i) items and a capacity C, and we compute the optimal fractional-knapsack value on the unperturbed data via a greedy ratio-based algorithm. Out-of-distribution evaluation increases item counts, broadens value and weight intervals, and introduces sparse perturbations.

Min-Coin-Change Dataset Each training instance presents a sorted list of coin denominations and a target amount T, and we compute the minimum number of coins (or zero if impossible) on this clean currency system. Out-of-distribution evaluation increases the number of denominations, broadens value and target ranges, and applies sparse perturbations.

Bin Packing Dataset Each example begins with a sorted list of item sizes and a global bin capacity; we label it with the number of bins estimated by a First-Fit Decreasing heuristic on the pristine sizes. Out-of-distribution evaluation increases item counts, expands size and capacity ranges, and introduces sparse perturbations.

Strongly Connected Components (SCC) Dataset Each sample is a random directed graph on n nodes, symmetrized to undirected, and labeled by computing connected components on the clean adjacency to produce a binary mask for node-pair connectivity. Out-of-distribution evaluation increases graph size, broadens edge-inclusion probabilities, and adds sparse perturbations.

D 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 repository. The primary packages utilized in constructing our experiment is Pytorch [39], Pandas and Scipy [32], SciKitLearn [40], and Numpy [15]. 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 3
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 100k 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,
   - token-wise BCE,
   - mean-squared error – regression tasks.

   $N_{\text{epoch}} = 100$ except for BIN PACKING and BALANCEDPARTITION, where $N_{\text{epoch}} = 1000$
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 3 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 Adversarial OOD, each input was perturbed with probability 0.5 with a random integer sampled from the task's adversarial range.

Table 3: Training hyperparameters and data ranges for each combinatorial task. Each task was trained with 100k samples, learning rate of 0.0001, input sequence length of 8, and no adversarial 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	Adversarial 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	1000	N/A	N/A	(1,10)	(11,100)	(10,30)
BinPacking	1000	(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 [1]	100	N/A	N/A	0.001	0.1	N/A

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