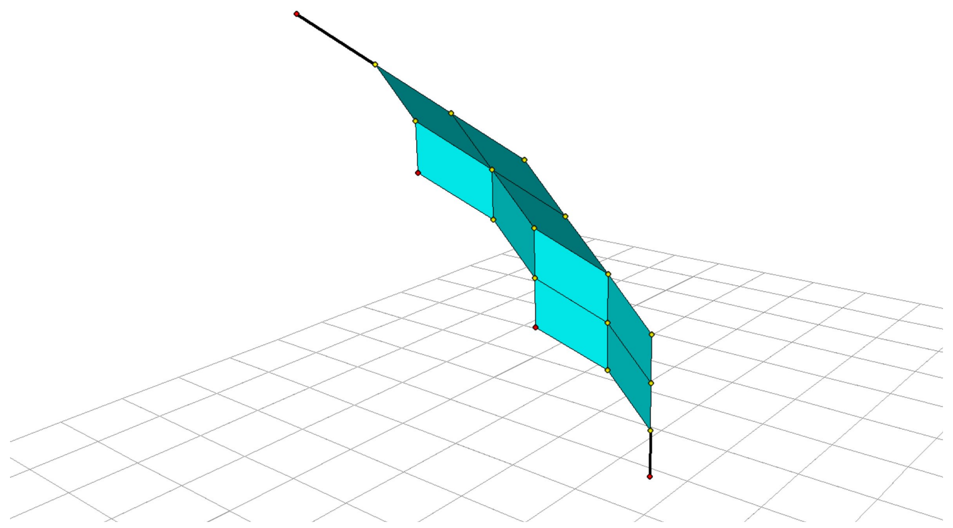
Tropical Attention: Neural Algorithmic Reasoning for Combinatorial Algorithms

Baran Hashemi, Kurt Pasque, **Chris Teska**, Ruriko Yoshida

ORIGINS Cluster, Technical University of Munich, Germany

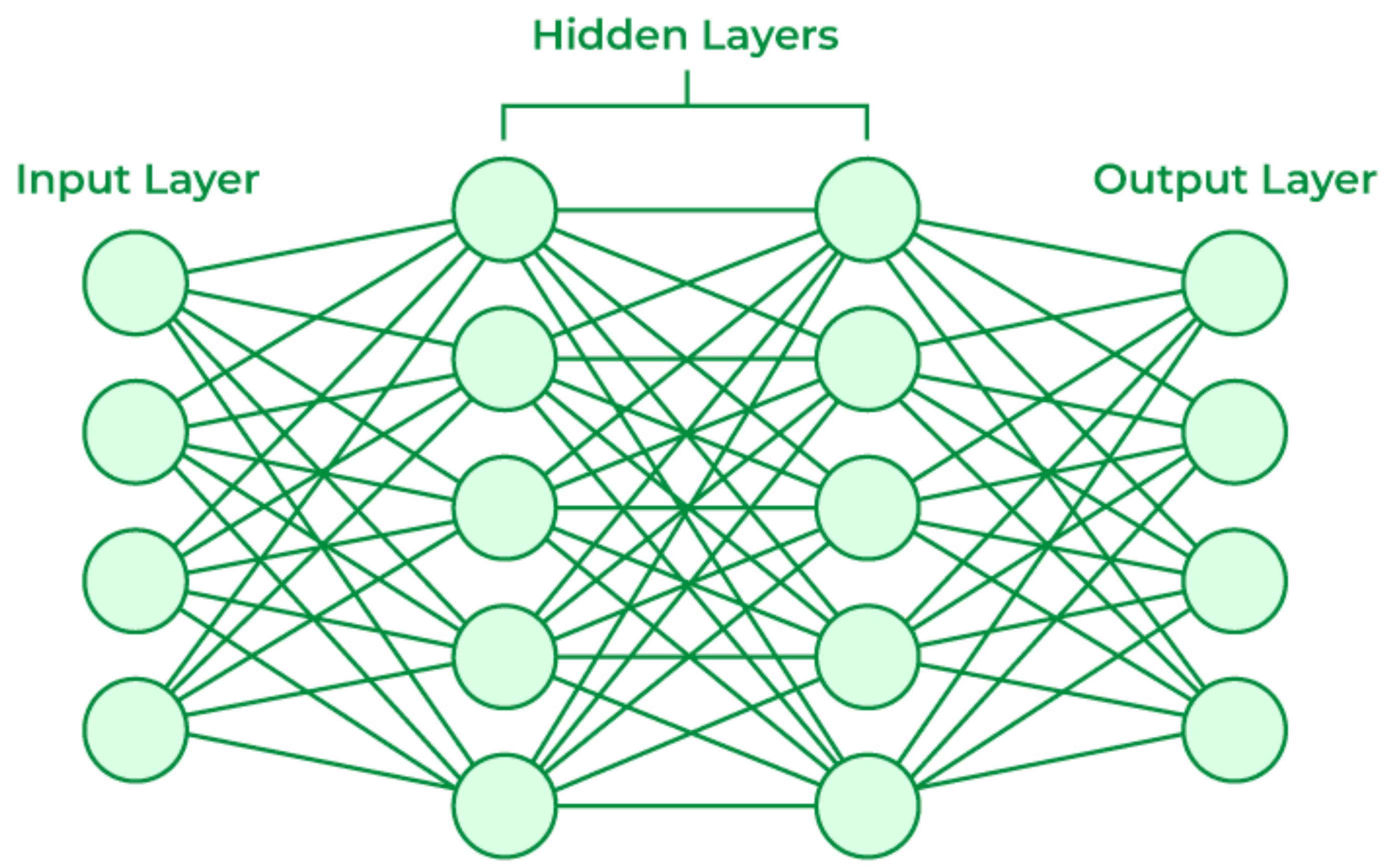
Naval Postgraduate School Monterey, California

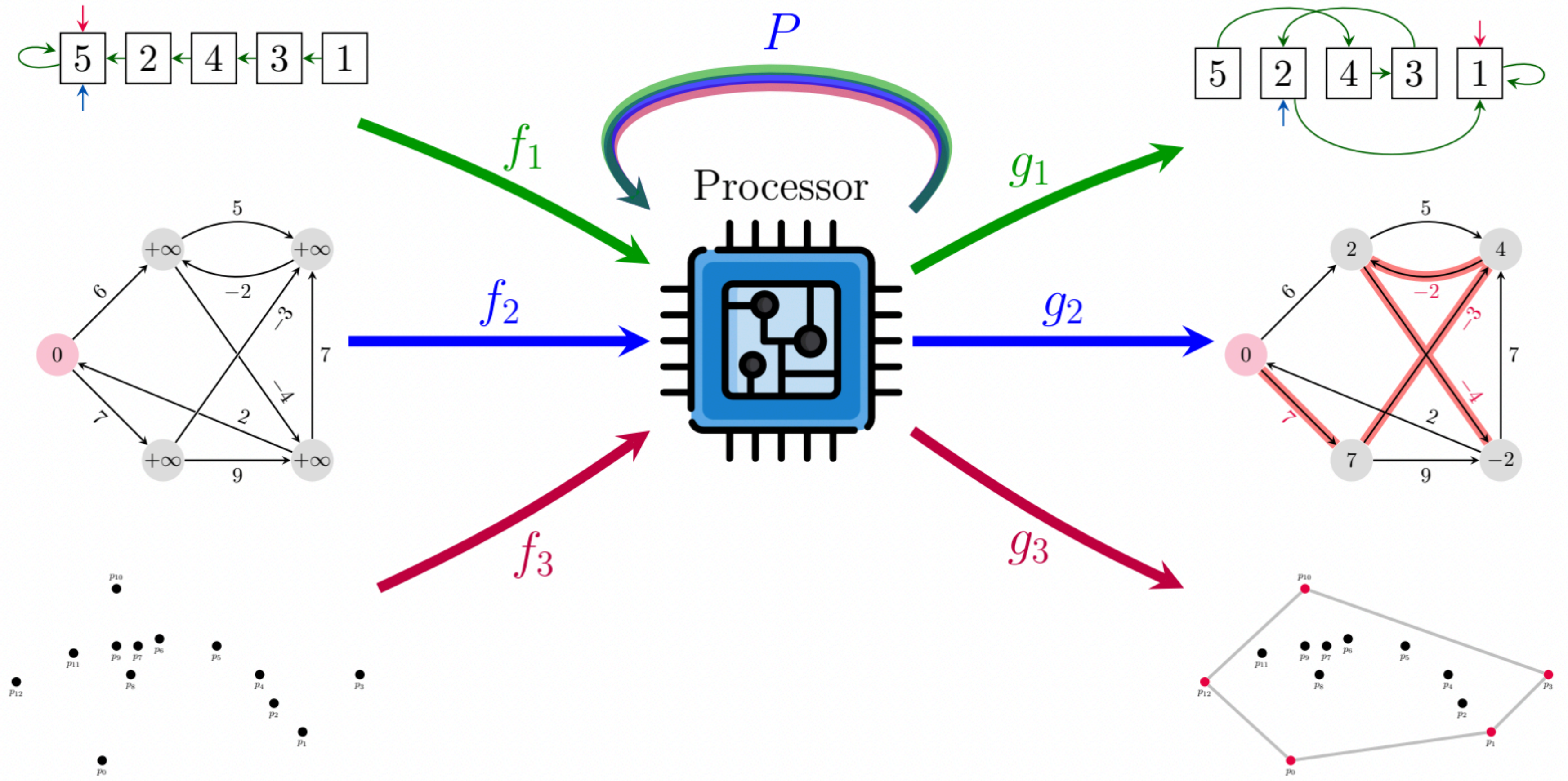


arXiv:2505.17190

Combinatorial Reasoning

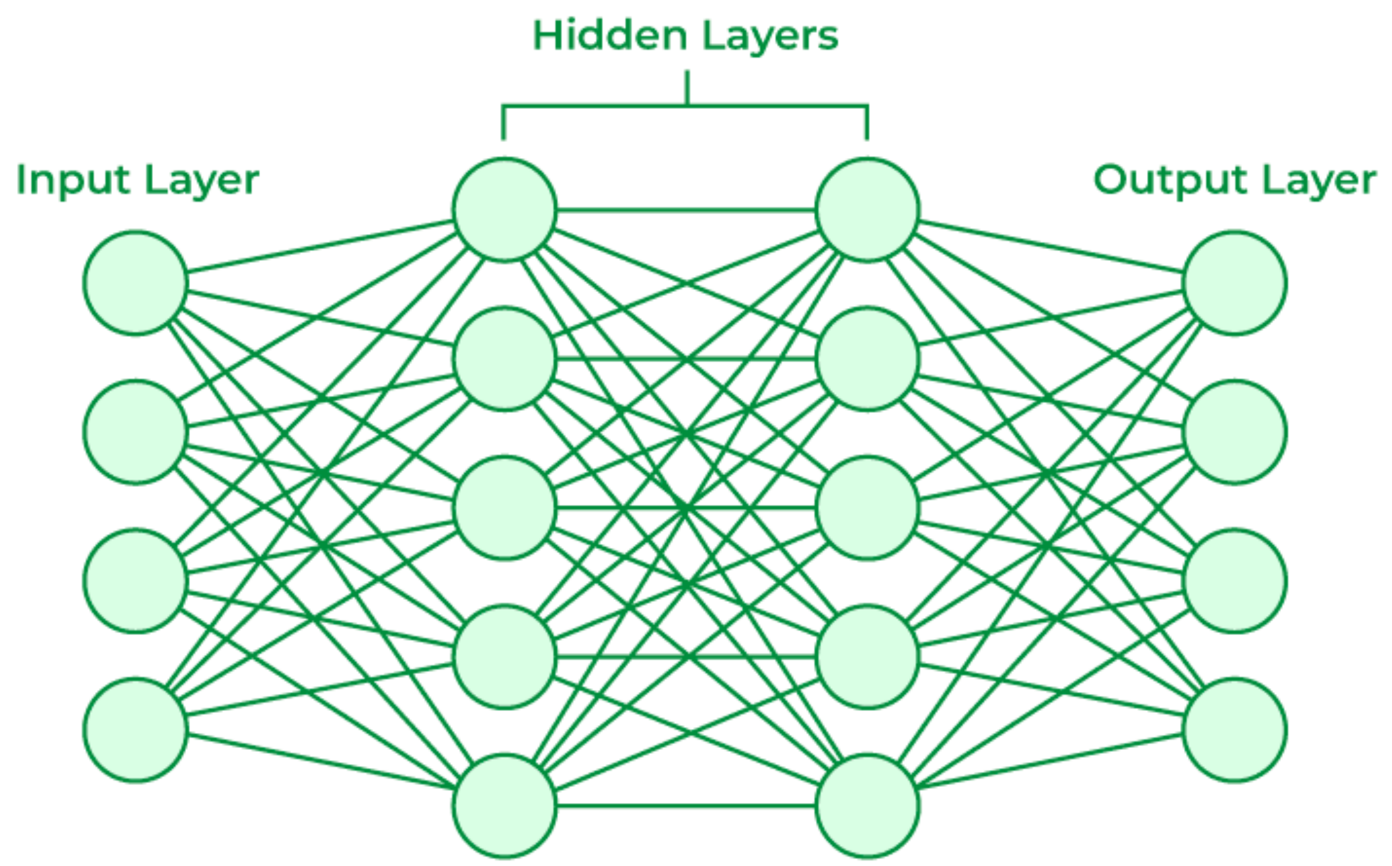
- Neural networks are exceptionally capable at pattern recognition

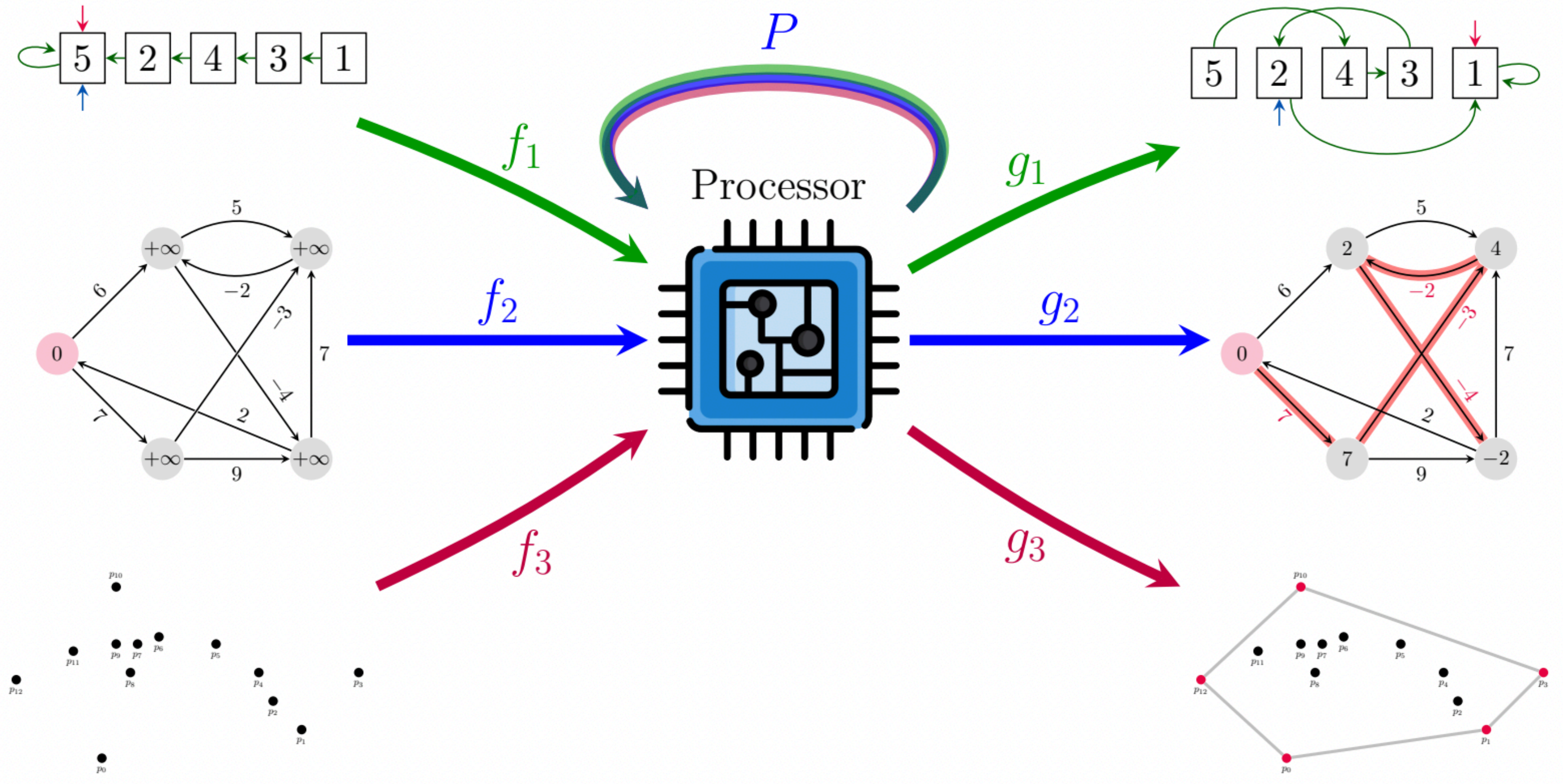




Combinatorial Reasoning

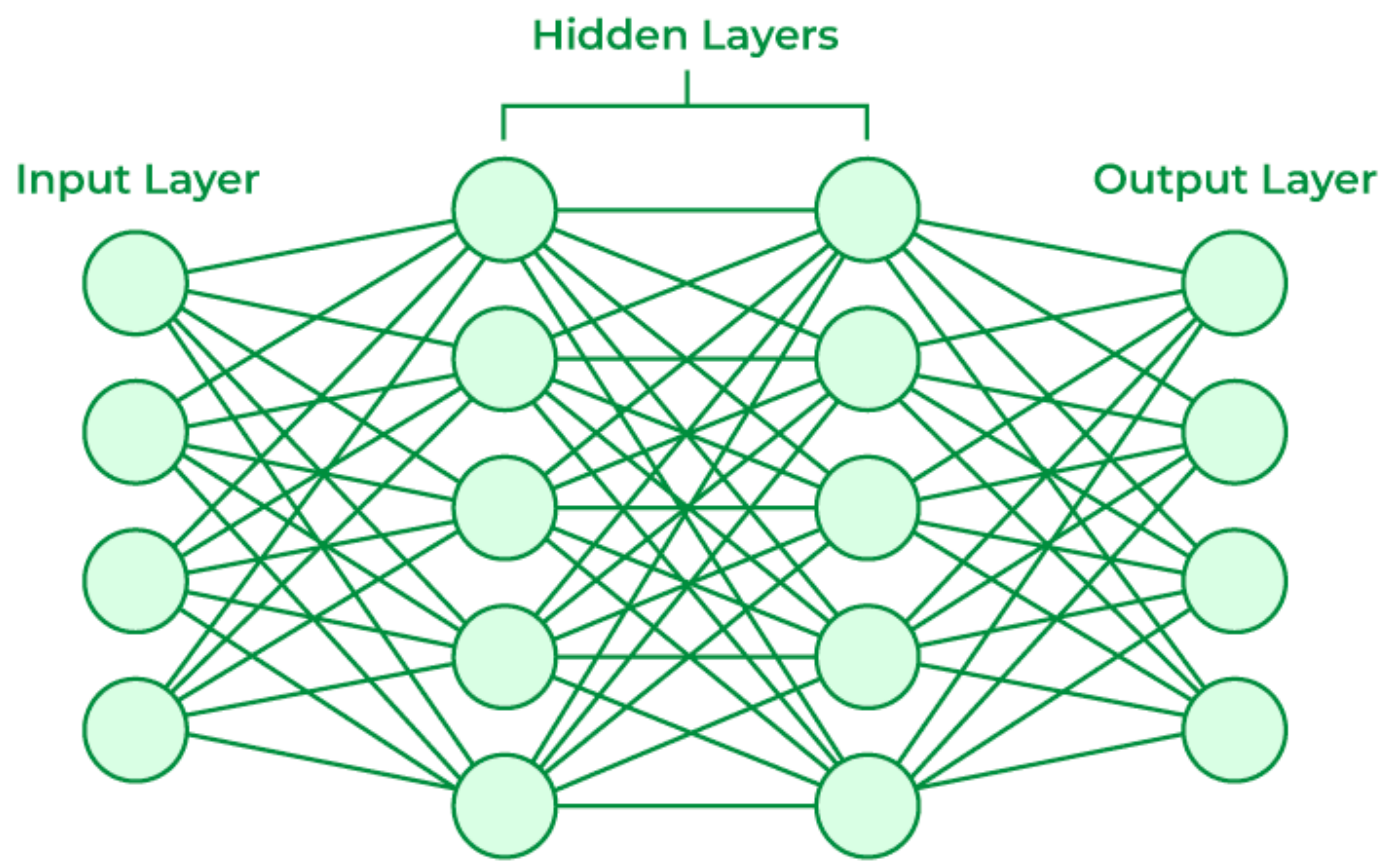
- Neural networks are exceptionally capable at pattern recognition
- In *Neural Algorithmic Reasoning* networks should *execute algorithms* vice mimic memorized input-output pairs

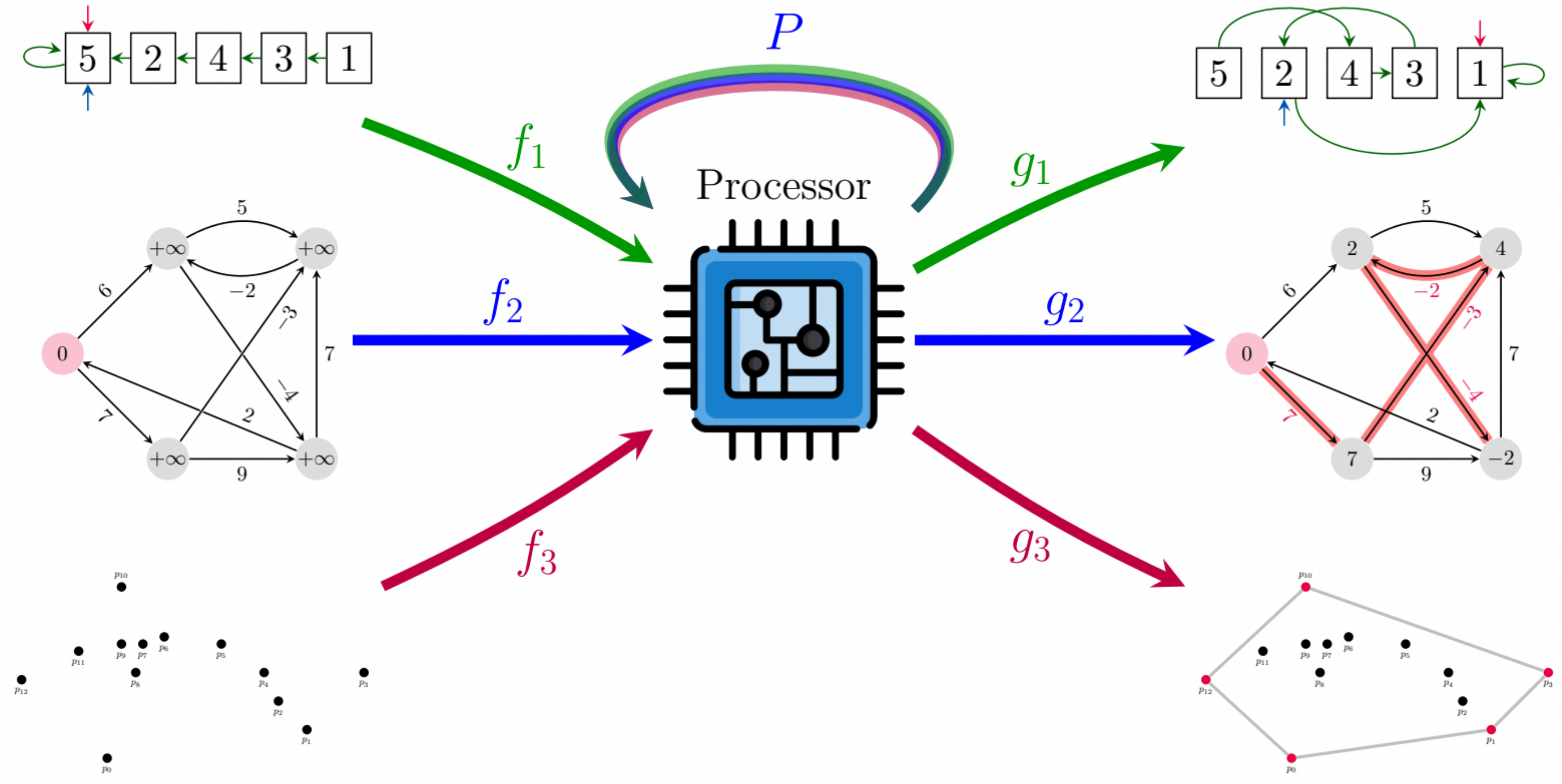




Combinatorial Reasoning

- Neural networks are exceptionally capable at pattern recognition,
- In *Neural Algorithmic Reasoning* networks should *execute algorithms* vice mimic memorized input-output pairs
- **Problem:** Standard softmax-dot product attention mechanisms live in smooth Euclidean space while combinatorial algorithms live in *piecewise-linear, polyhedral spaces*





Why Does Softmax Fail?

$$\alpha_{ij} = \mathrm{softmax}_\tau\left(\langle \mathbf{q}i, \mathbf{k}_j \rangle\right) := \frac{\exp\left(\langle \mathbf{q}_i, \mathbf{k}_j \rangle / \tau\right)}{\sum_{t=1}^{N} \exp\left(\langle \mathbf{q}_i, \mathbf{k}_t \rangle / \tau\right)}$$

Logit ***blurriness*** or ***Dispersion*** starts to happen ==> **Attention Fading**
Leads to soft/blurry rather than decisive selections,
hampering tasks that require deterministic decisions

Why Does Softmax Fail?

$$\alpha_{ij} = \mathrm{softmax}_\tau\big(\langle \mathbf{q}i, \mathbf{k}_j \rangle\big) := \frac{\exp\big(\langle \mathbf{q}_i, \mathbf{k}_j \rangle / \tau\big)}{\sum_{t=1}^{N} \exp\big(\langle \mathbf{q}_i, \mathbf{k}_t \rangle / \tau\big)}$$

Logit ***blurriness*** or ***Dispersion*** starts to happen ==> **Attention Fading**
Leads to soft/blurry rather than decisive selections,
hampering tasks that require deterministic decisions

What do we need?

Algorithmic alignment → ***Smart Inductive Bias*** *from combinatorial algorithms*

Algorithmic Alignement

What do we need?

1. Combinatorial Algorithms require hard argmax/argmin operations
2. Captures the piecewise-linear geometry (decision boundary) of combinatorial problems
3. Robust wrt noise and perturbations (e.g adversarial attacks)

Algorithmic Alignement

What do we need?

1. Combinatorial Algorithms require hard argmax/argmin operations
2. Captures the piecewise-linear geometry (decision boundary) of combinatorial problems
3. Robust wrt noise and perturbations (e.g adversarial attacks)

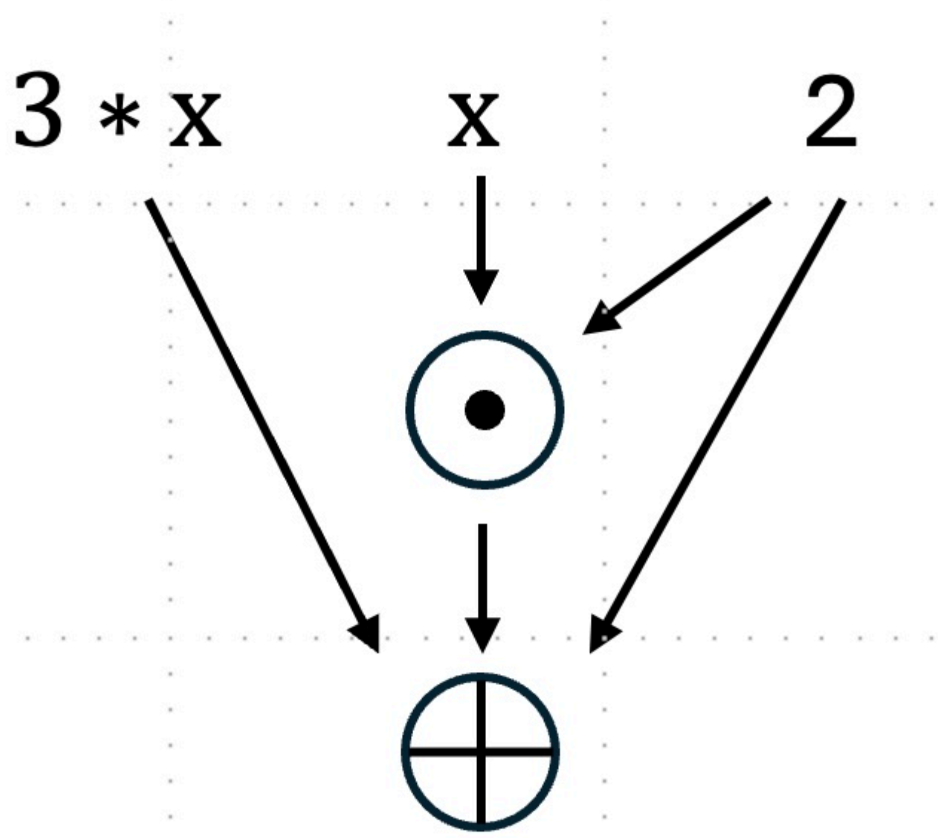
These three pillars are exactly the native properties of Tropical Algebraic Geometry

- (max, +) *are the basic operations*
- *Polyhedral and piecewise-linear boundaries are the geometry*
- *Robustness to perturbations is built in*

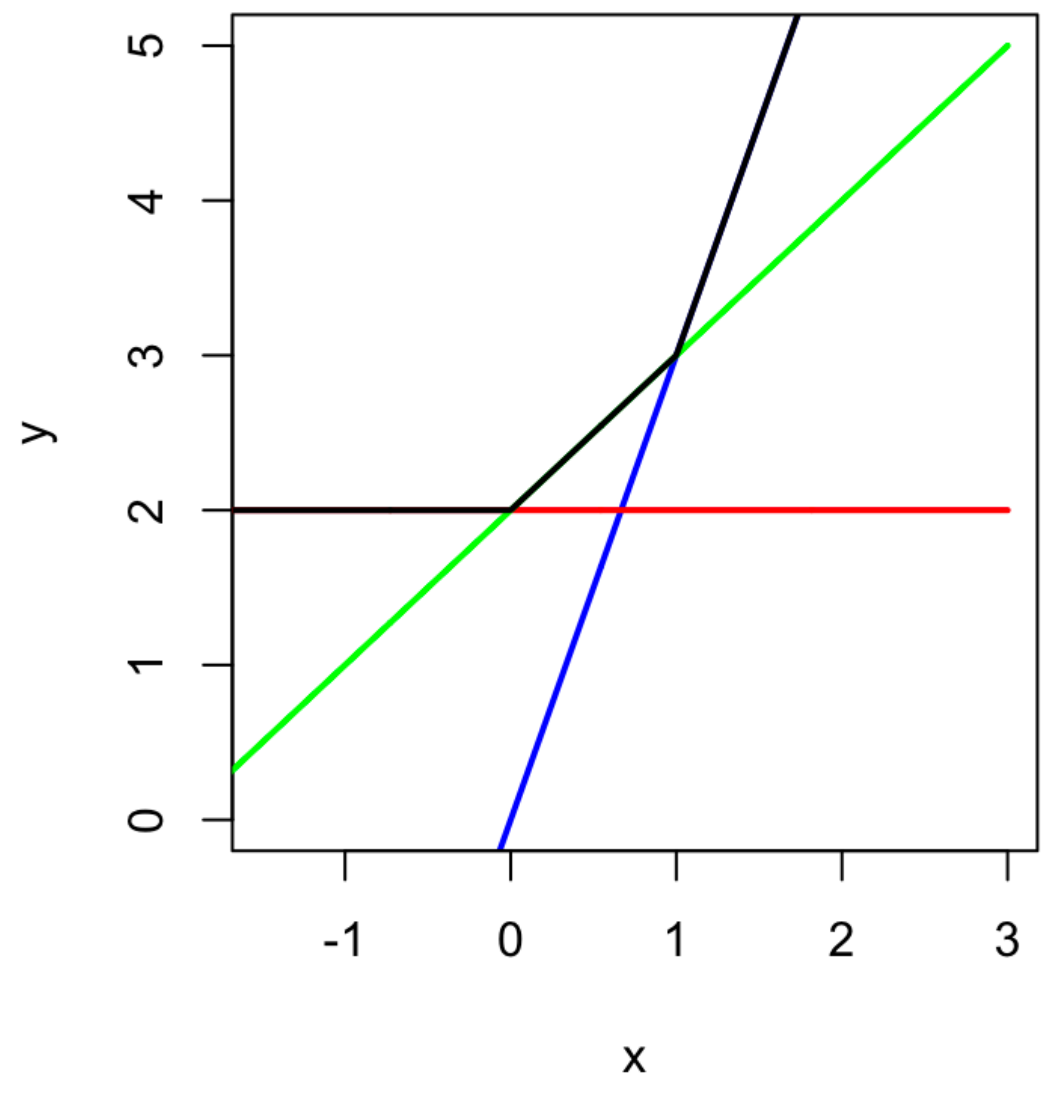
Algorithmic Alignement

What do we need?

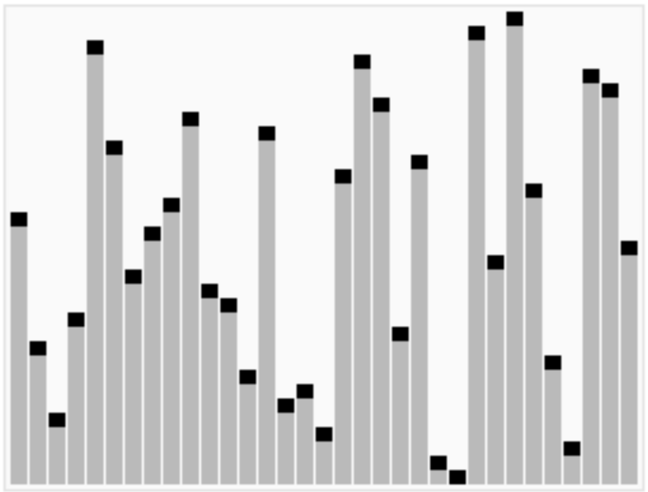
1. Combinatorial Algorithms require hard argmax/argmin operations
2. Captures the piecewise-linear geometry (decision boundary) of combinatorial problems
3. Robust wrt noise and perturbations (e.g adversarial attacks)



$f(x) = x^3 \oplus 2 \odot x \oplus 2 = max\{3x,\ 2 + x,\ 2\}$



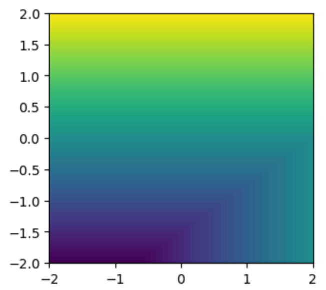
Tropical Attention



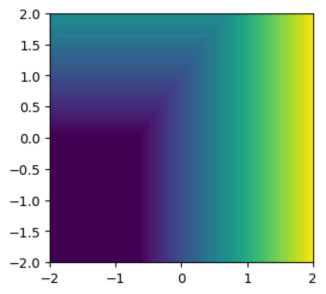
$\mathbf{X} \in \mathbb{R}^{N \times d}$

Tropical Simplex Valuation Map

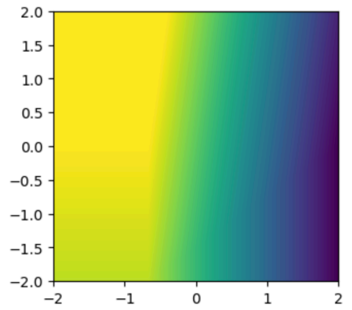
$\Phi_\lambda(\mathbf{X}) = \log(\max(0, \mathbf{X})) - \lambda$



$\mathbf{K} = \bigoplus(\Phi_\lambda(\mathbf{X}) \odot \mathbf{W_k})$



$\mathbf{Q} = \bigoplus(\Phi_\lambda(\mathbf{X}) \odot \mathbf{W_q})$



$\mathbf{V} = \bigoplus(\Phi_\lambda(\mathbf{X}) \odot \mathbf{W_v})$

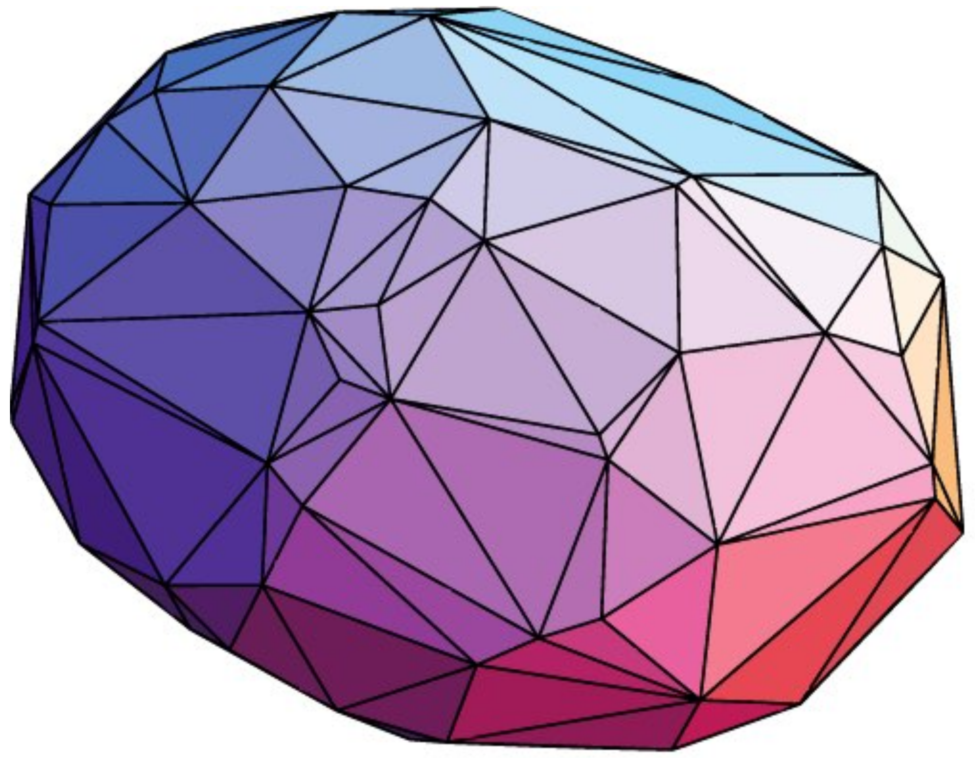
Tropical Attention

$$d_{\mathbb{H}}(\mathbf{K}, \mathbf{Q}) = \max(\mathbf{K} - \mathbf{Q}) - \min(\mathbf{K} - \mathbf{Q})$$

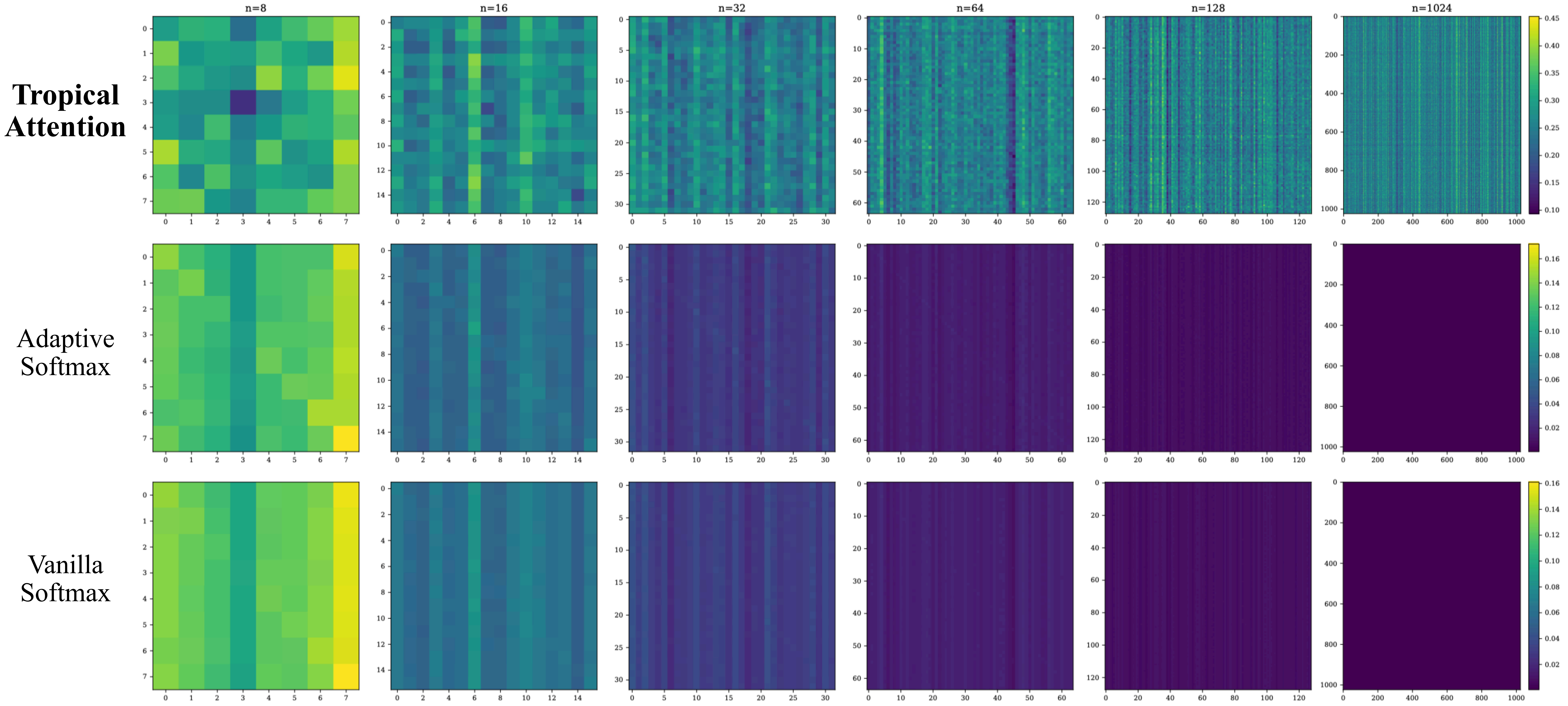
$$\mathbf{C} = \bigoplus(-d_{\mathbb{H}} \odot \mathbf{V}) = \max(-d_{\mathbb{H}} + \mathbf{V})$$

attention becomes:

“pick the value that best aligns projectively with the query.”

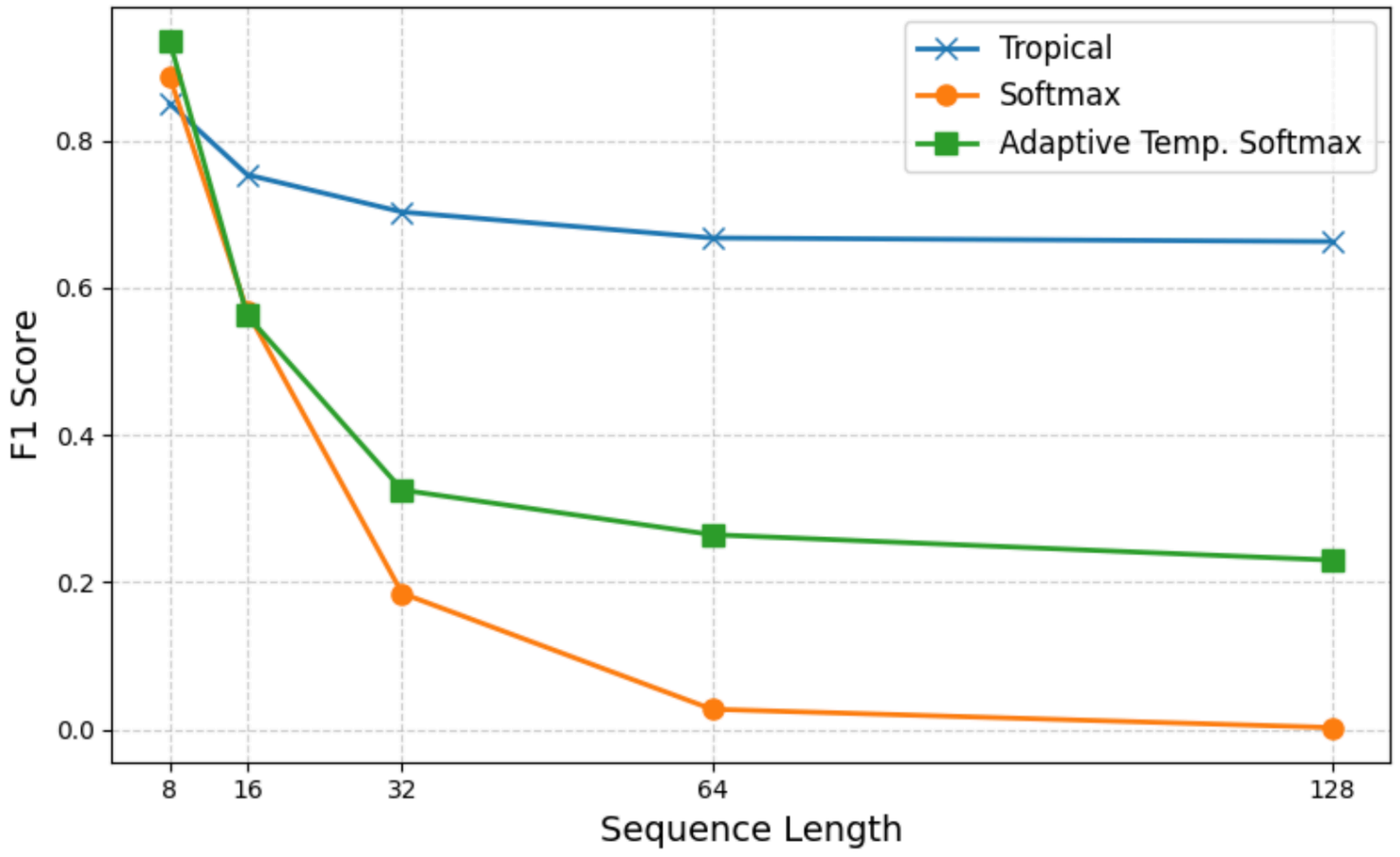


Result: Quickselect

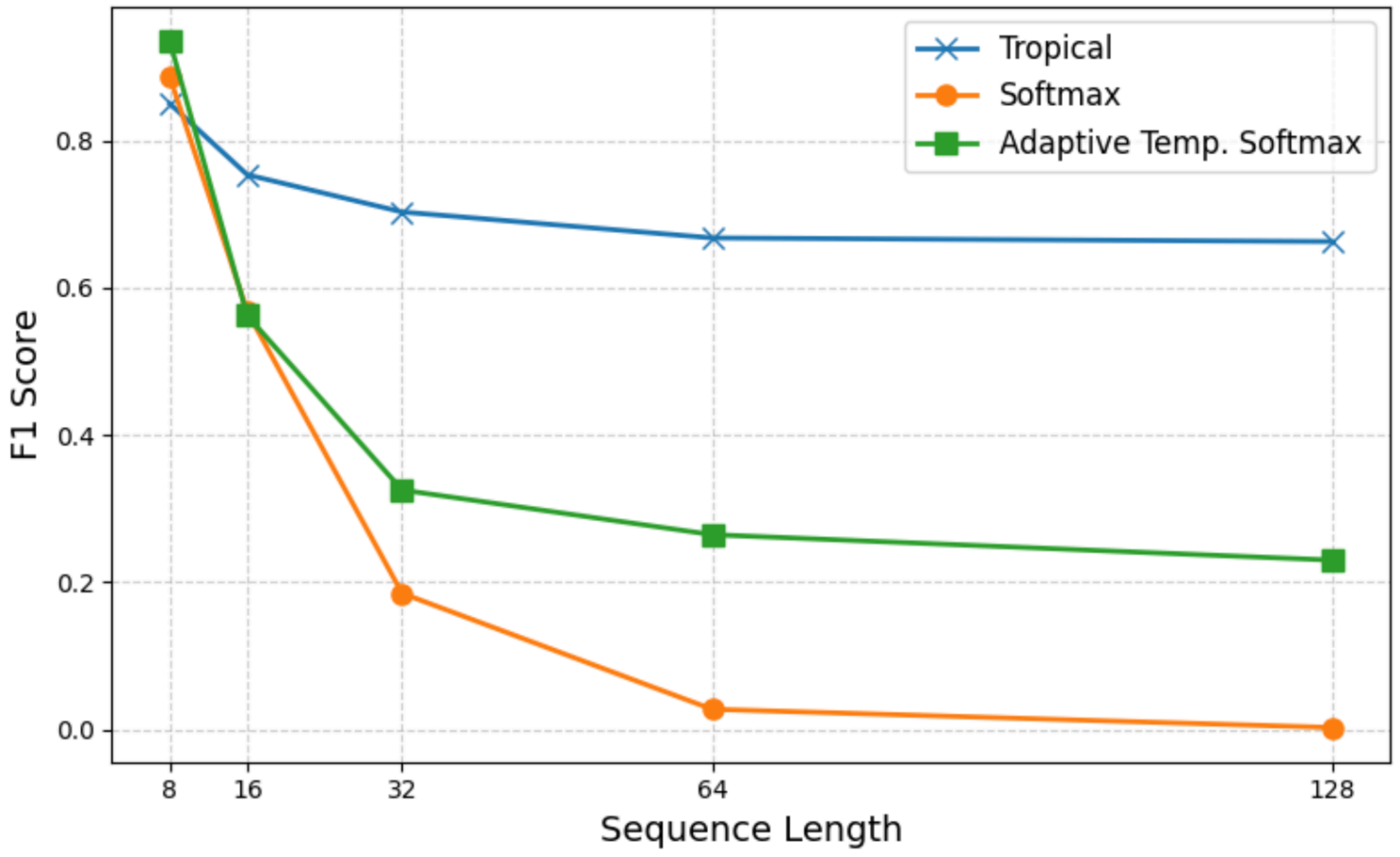


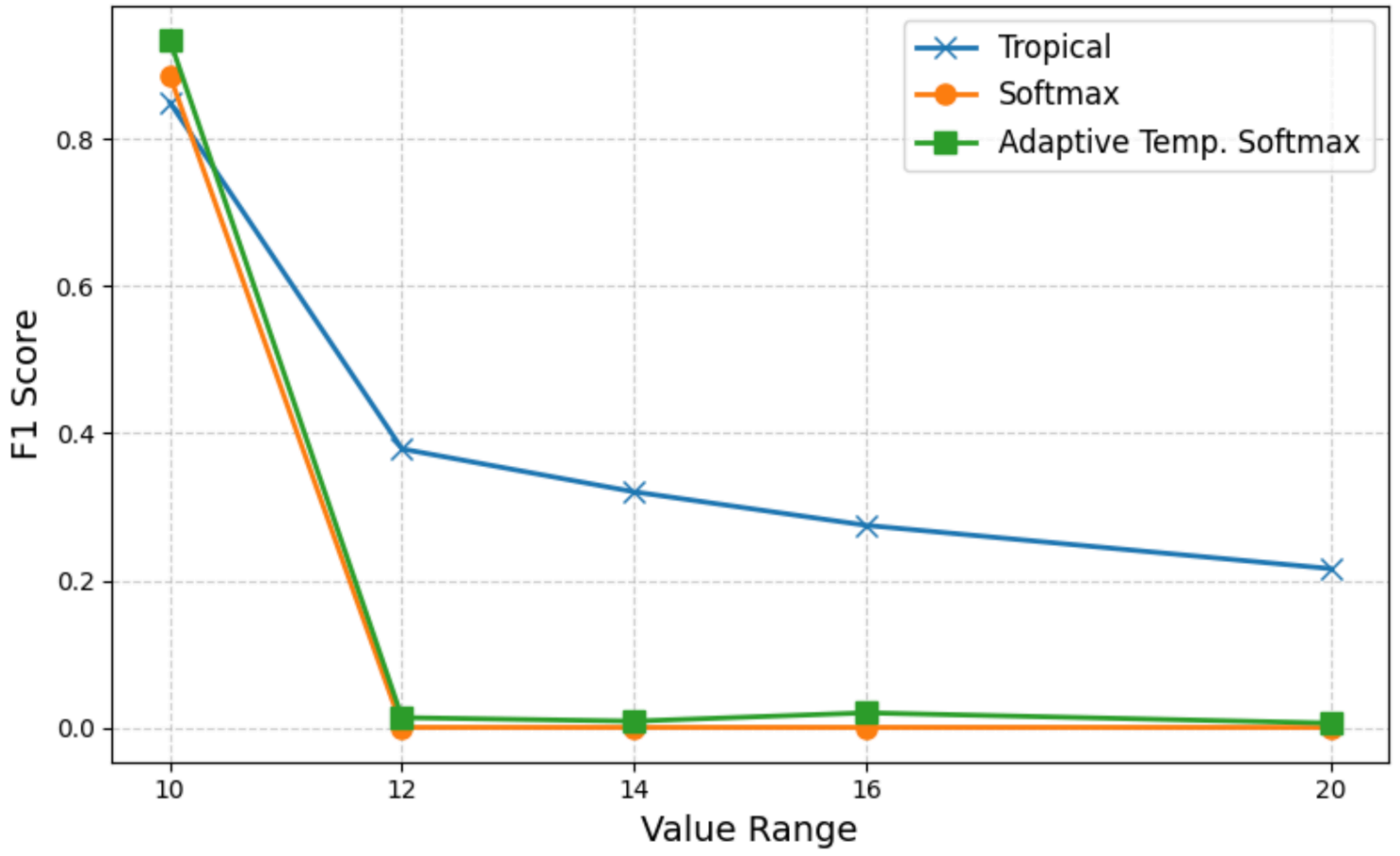
2410.01104 (Softmax is not Enough (for Sharp Size Generalisation))

Result: Quickselect

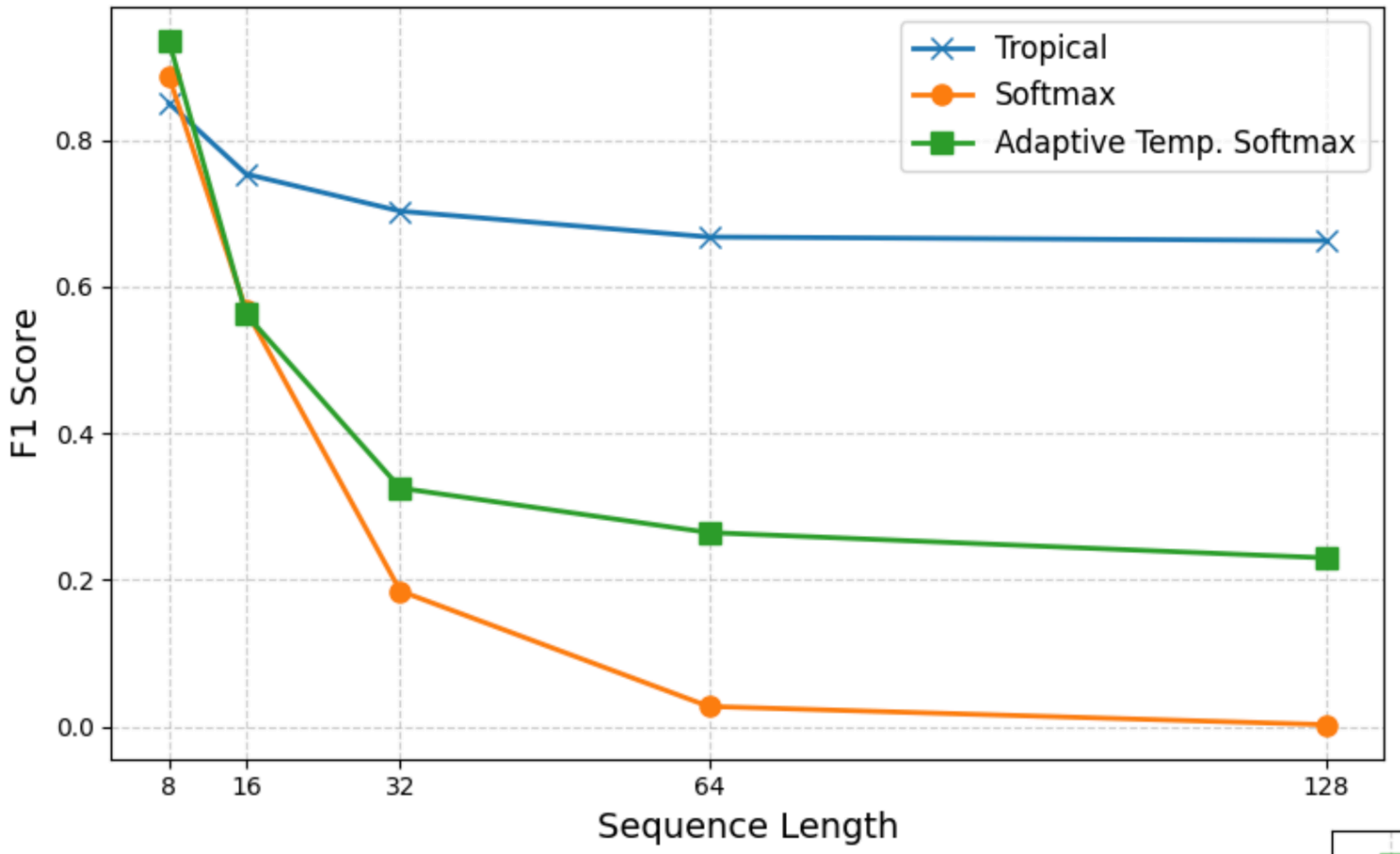


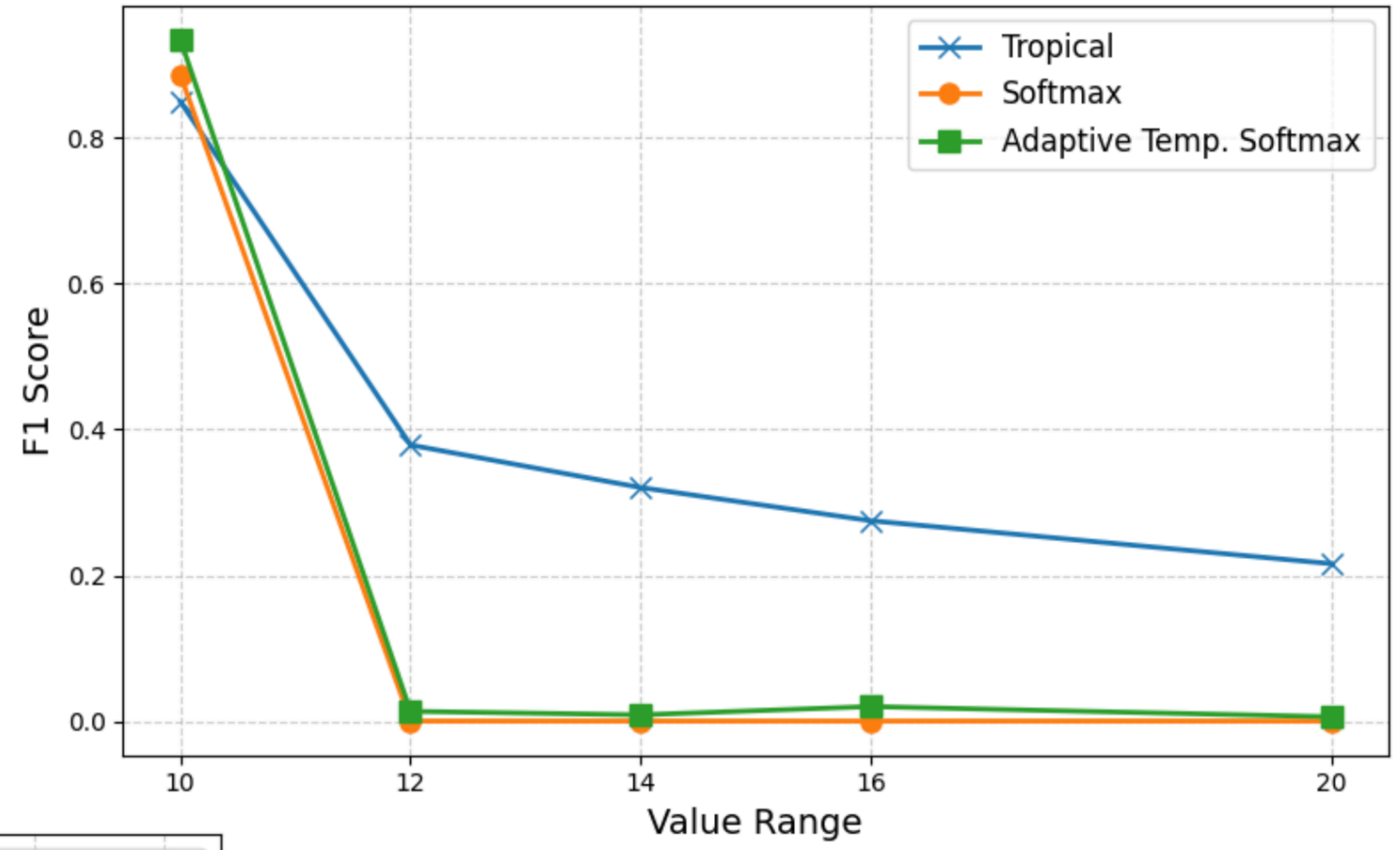
Result: Quickselect

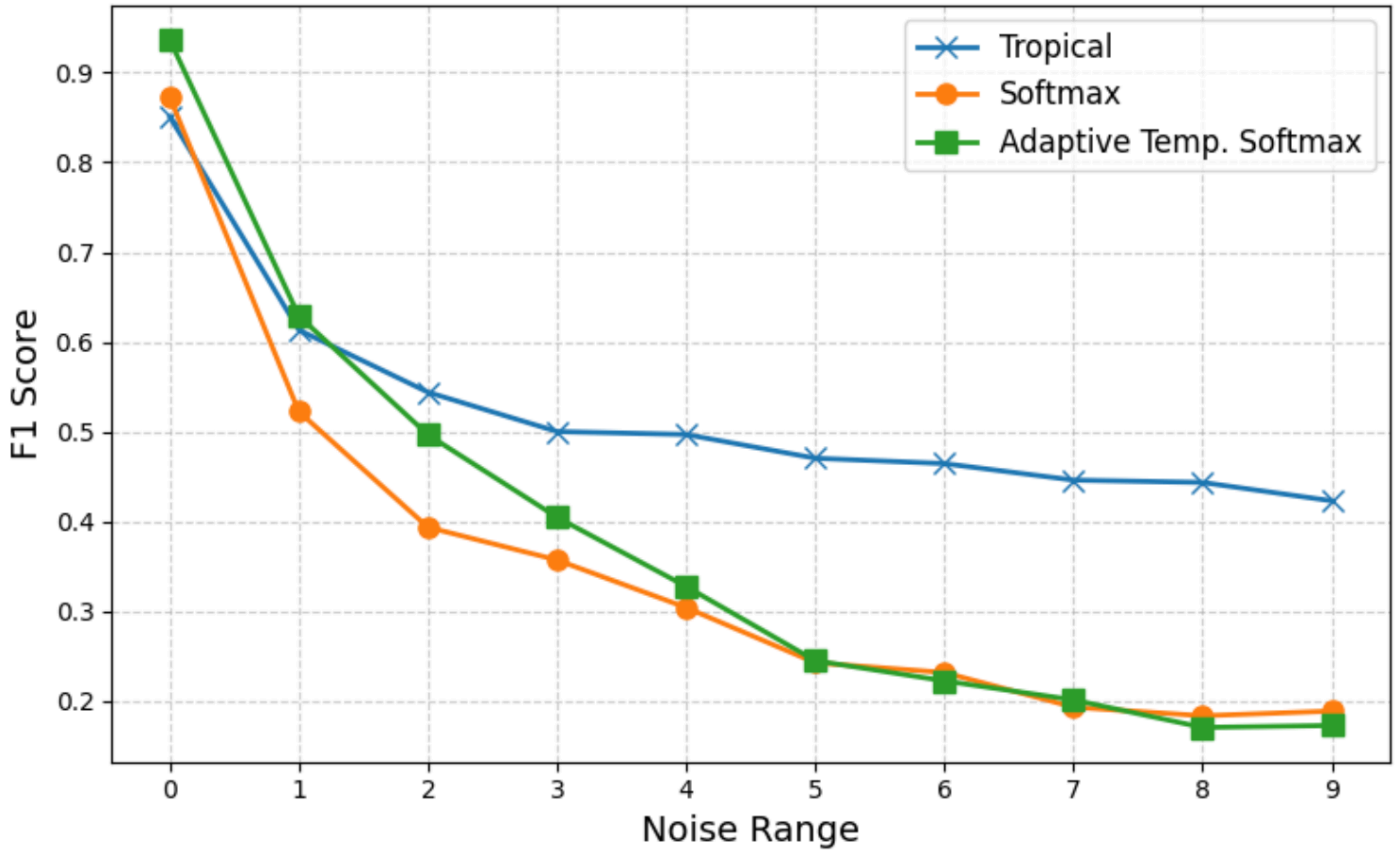




Result: Quickselect







11 CO tasks

1. **Quickselect:** Find the k-th smallest elements in a set. (Classification)
2. **3SUM (Decision):** Decide if there exist a, b, c with $a + b + c = T$. (Classification)
3. **Balanced Partition:** Split numbers into two subsets with equal sum. (NP-complete Classification)
4. **Convex Hull:** Given 2D points, identify the hull. (Classification)
5. **Subset Sum (Decision):** Decide if some subset sums to a target T. (NP-complete Classification)
6. **0/1 Knapsack:** Maximize value under capacity with binary item choices. (NP-hard Classification)
7. **Fractional Knapsack:** Items can be taken fractionally; predict optimal value. (Regression)
8. **Floyd–Warshall:** All-pairs shortest paths on a weighted undirected graph. (Both Regression and Classification)
9. **Strongly Connected Components (SCC):** Decompose a directed graph with community structure; predict pairwise same-component. (Classification)
10. **Bin Packing:** Pack items into the fewest bins of fixed capacity. (NP-hard Classification)
11. **Min Coin Change:** Minimum number of coins to reach a target T with each coin used at most once. (Classification)

Results

Algorithmic Tasks	Length OOD			Value OOD			Perturbative Noise		
	`Vanilla`	`Adaptive`	`Tropical`	`Vanilla`	`Adaptive`	`Tropical`	`Vanilla`	`Adaptive`	`Tropical`
ConvexHull	42.752.06	48.250.96	**97.001.15**	22.753.59	23.773.10	**34.251.71**	90.752.22	91.002.16	**96.002.16**
Knapsack	41.061.76	39.182.59	**60.002.09**	38.873.43	26.921.33	**49.672.01**	67.853.19	68.363.51	**74.673.13**
QuickSelect	4.665.98	22.892.49	**77.063.78**	74.222.30	**74.301.99**	71.103.11	33.877.11	34.824.79	**57.225.01**
BinPacking	60.752.49	64.251.09	**66.011.55**	67.263.70	74.231.51	**78.541.89**	55.385.10	60.643.92	**61.194.33**
SCC	51.303.91	56.502.22	**89.253.49**	78.513.08	**81.382.62**	74.865.01	70.005.98	**71.331.96**	69.864.17
SubsetSum	21.132.45	22.755.25	**87.506.45**	34.756.60	28.5010.12	**79.255.38**	3.751.50	3.001.63	**72.7510.01**
BalancedPartition	80.552.91	91.905.52	**96.733.50**	**63.404.29**	56.571.18	55.765.63	51.062.66	57.061.08	**57.291.33**
3SUM	80.000.82	79.750.50	**82.751.59**	26.003.16	**26.253.50**	22.002.16	47.509.47	49.259.22	**65.253.59**
MinCoinChange	9.251.86	17.982.29	**42.521.47**	23.644.07	2.201.13	**33.185.64**	22.122.75	18.444.49	**33.754.89**
Floyd–Warshall	12.814.03	1.310.36	**0.810.08**	87.685.65	56.303.04	**55.304.36**	7.543.63	5.292.56	**4.391.62**
FractionalKnapsack	0.880.06	0.860.08	**0.660.10**	0.240.12	0.170.03	**0.080.03**	0.050.02	0.030.01	**0.020.01**

Experimental Saga: LRA benchmark

Long Range Arena (LRA) benchmark,
A standard for testing transformers on long-sequence tasks across text, image, and math domains.

ListOps: Calculating the result of nested mathematical operations in parentheses (e.g min[max[1,2,3],4,5] = 3.)
Text Classification: The IMDB sentiment analysis task
Document Retrieval: Finding a similarity score between two documents
Image Classification: The CIFAR-10 image classification task
Pathfinder: To identify whether two white points in a black background are connected by a path of dashed white lines.

Models	ListOps	Text	Retrieval	Image	Pathfinder	Avg.	Complexity
Transformer	36.37	64.27	57.46	42.44	71.40	54.39	$\mathcal{O}(n^2)$
Longformer	35.63	62.85	56.89	42.22	69.71	53.46	$\mathcal{O}(n)$
Linformer	35.70	53.94	52.27	38.56	76.34	51.36	$\mathcal{O}(n)$
Reformer	37.27	56.10	53.40	38.07	68.50	50.67	$\mathcal{O}(n)$
Performer	18.01	65.40	53.82	42.77	77.50	51.41	$\mathcal{O}(n)$
AdaptiveSoft.	47.15	75.52	79.56	51.58	80.94	66.95	$\mathcal{O}(n^2)$
Elliptical	37.8	65.6	80.3	40.2	73.2	61.24	$\mathcal{O}(n^2)$
Fourierformer	40.73	75.02	85.35	53.17	83.43	67.54	$\mathcal{O}(n \log n)$
MEGA	63.14	**90.43**	**91.25**	**90.44**	96.01	**86.25**	$\mathcal{O}(n \log n)$
Tropical	**68.65**	70.13	64.82	60.04	**97.33**	72.79	$\mathcal{O}(n^2)$

Discussion

- Tropical Attention brings:
  - **OOD length and value generalization** performance while being **robust** to heavy noise
  - **Expressivity:** *Multi-Head Tropical Attention is a Tropical circuit that universally approximates* (max , +) *Dynamic Programs. (Theorem 3.2)*
    - Preserves the piecewise-linear geometry of combinatorial problems
    - A shallow network provides a one-shot solution (sufficient) via the tropical transitive closure
  - **Sharp attention maps** that upholds algorithmic reasoning beyond the training data ==> *Size Invariance*

- *Open Horizons:*
  - Tropical Attention directly on Graphs (Tropical GNN)?
  - Applications in tasks with combinatorial flavour?

https://github.com/Baran-phys/Tropical-Attention