Algorithmic Task Capture, Computational Complexity, and Inductive Bias of Infinite Transformers

Orit Davidovich
IBM Research, Israel
orit.davidovich@ibm.com

Zohar Ringel
Racah Institute of Physics, HUJI
zohar.ringel@mail.huji.ac.il

ISDSA 2026

At a Glance

Questions.

- Ambitious: What are transformers capable of "understanding"?
- Specific: Can transformers learn scalable heuristics for combinatorial tasks?

Introduce.

- Algorithmic task capture.
- Efficient poly-time heuristic schemes (EPTHS).

Upshot.

- Transformers can exhibit genuine algorithmic capture on some combinatorial tasks, but their inductive bias appears to favor procedures with low inference-time EPTHS complexity.

Infinite-width Transformer

- We study trained transformers as inference-time algorithms.
- We consider transformers at the infinite-width limit:
  - Embedding dimension and number of heads $\to \infty$.
  - It removes finite-width capacity as the bottleneck.
  - This uses kernel predictors [1].
- Learning regimes of infinite-width transformers:
  - Neural network Gaussian process (NNGP) [4].
  - Neural tangent kernel (NTK) / "lazy" learning [2, 3].
  - "Rich" learning / feature learning.

Notation.

- $\mathbf{X} \in \mathbb{R}^{T \times d}$ transformer input / task instance
- T input context length / task instance size
- $X \sim \mu_{X,T}$ probability measure per T
- $g : \mathcal{X} \to \mathcal{Y}$ combinatorial task
- $\boldsymbol{\Delta}_g > 0$ accuracy margin of combinatorial task g

Theoretical Results

Theorem. The computational complexity of evaluating the Note Carlo (MC) estimator of the NNGP / NTK kernel predictor is $O(PN_{\text{MC}}T^3)$, with MC error scaling as $O(\sqrt{P/N_{\text{MC}}})$.

Convergence Assumption. The discrepancy between the infinite-width predictor and the finite-width transformer in the corresponding regime scales as P^γ/N, where N is the architecture width scale and $\gamma > 0$ is architecture-dependent.

Theorem. Consider a finite-width transformer of width N, trained with full-batch gradient descent (GD), weight decay, and additive white Gaussian noise. Let $f(X) = \mathbf{z}_T^{\text{out}}(X)$ for $X \in \mathbb{R}^{T \times d}$, and let $f_K(X)$ be its NNGP kernel predictor with $K = \Sigma_{TT}^{\text{out}}$. Then, in standard scaling,

$$\left|\mathbb{E}_{f \sim P[f]}[f(X_*)] - f_K(X_*)\right| = O\left(\frac{P^\gamma}{N}\right)$$

where $P[f]$ [5, Eq. (3)] is the posterior on the trained f.

EPTHS Complexity

Definition. An algorithm A is an *efficient polynomial-time heuristic scheme* (EPTHS) of complexity $O(T^k)$ for a combinatorial task $g : \mathcal{X} \to \mathcal{Y}$ if for every $\delta \in (0, 1]$ there exists a δ-implementation A_δ of A that returns an output $A_\delta(X) \in \mathcal{Y}$ in time $O(\eta(1/\delta)T^k)$ such that

$$\Pr_{X \sim \mu_{X,T}}\left[\text{dist}_{\mathcal{Y}}(A_\delta(X), g(X)) < \Delta_g/2\right] > 1 - \delta$$

for sufficiently large T.

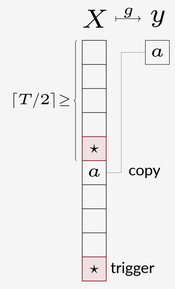
EPTHS from Average-case. If an algorithm A is always correct in time $t(X)$ and $\mathbb{E}_{X \sim \mu_{X,T}}[t(X)] = O(T^k)$, then A defines an EPTHS of complexity $O(T^k)$.

Combinatorial Tasks

A combinatorial task maps structured problem instances to discrete outcomes. We used 6 tasks to probe whether transformers learn scalable procedures.

Induction

$X \xmapsto{g} y$



Vocab. $= 1024$

Sorting

$X \xmapsto{g} y$

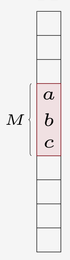
42	7
7	7
91	13
42	24
13	42
75	42
7	58
58	75
24	88
88	91

Vocab. $\in \{100, 200, 300\}$

String Matching

$X \xmapsto{g} y$

$\mathbf{1}_{M \subseteq X}$

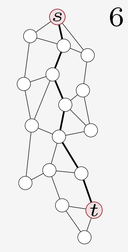


Vocab. $= 26$, $|M| = 3$

Shortest Path Problem (SPP)

$X \xmapsto{g} y$

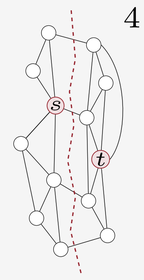
6



MinCut

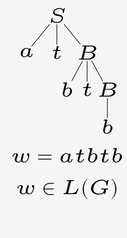
$X \xmapsto{g} y$

4



Context-free Grammar (CFG)

$w \xmapsto{g} \mathbf{1}_{\{w \in L(G)\}}$



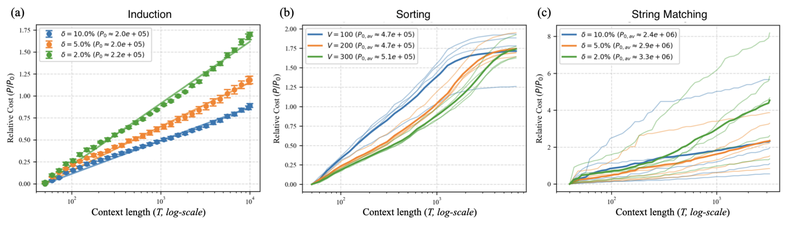
$w = atbtb$

$w \in L(G)$

$G = (V, \Sigma, R, S)$
$V = \{S, B\}$
$\Sigma = \{a, b, t\}$
R :
$S \to a$
$S \to atB$
$B \to b$
$B \to btB$
$B \to at$

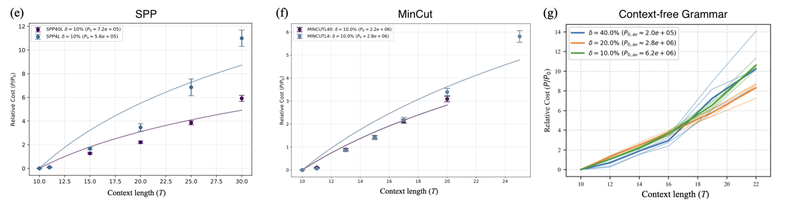
Empirical Data Adaptation

We fine-tune models trained at an initial context length T_0 on larger task sizes T, and measure the relative additional number of samples (P/P_0) required to re-achieve the target error δ. Logarithmic growth is evidence of algorithmic task capture.



Evidence of capture at log scale; Panels (b-c) plot-lines per seed, average in bold.

(a) Induction: initial $P_0 \simeq (0.20\text{–}0.22)$M; re-achieve $\delta = 10\%, 5\%, 2\%$.
(b) Sorting: initial $P_0 \simeq (0.47\text{–}0.51)$M; re-achieve $\delta = 5\%$ on $|V| = 100, 200, 300$.
(c) String Matching: initial $P_0 \simeq (2.4\text{–}3.3)$M; re-achieve $\delta = 10\%, 5\%, 2\%$.



Evidence of non-capture; Panels (e-f) accelerating curves; log fits shown as visual reference; Panel (g) plot-lines per seed, average in bold.

(e) SPP: initial $P_0 \simeq (0.56\text{–}0.72)$M; fine-tune to re-achieve $\delta = 10\%$.
(f) MinCut: initial $P_0 \simeq (2.2\text{–}2.8)$M; fine-tune to re-achieve $\delta = 10\%$.
(g) CFG: initial $P_0 \simeq (0.2\text{–}6.2)$M; re-achieve $\delta = 40\%, 20\%, 10\%$.

Algorithmic Task Capture

Definition. A transformer f is said to *algorithmically capture* a combinatorial task $g : \mathcal{X} \to \mathcal{Y}$ if for every sufficiently small $\delta > 0$ there exist $T_0 = T_0(\delta)$, $C_0 = C_0(\delta)$, and $P_0 = P_0(\delta)$ such that, for every $T \geq T_0$,

$$\Pr_{X \sim \mu_{X,T}}\left[\text{dist}_{\mathcal{Y}}(f_T(X), g(X)) < \Delta_g/3\right] > 1 - \delta,$$

where f_T is the predictor obtained by the two-stage training protocol:

(i) initial training on P_0 samples from $\mu_{X,1}, \ldots, \mu_{X,T_0}$ shared across all $T \geq T_0$;
(ii) secondary fine-tuning on $C_0 \log(T/T_0)$ additional samples from $\mu_{X,1}, \ldots, \mu_{X,T}$.

The underlying transformer architecture, training procedure, hyperparameter choices, and initialization/training randomness convention are fixed.

Combinatorial Task	$\mathbf{L}$	$\mathbf{H}$	$\mathbf{d}_{\text{mod}}$	$\mathbf{T}$	Architecture
Induction	2	4	256	$[50, 10000]$	Causal, RoPE
Sorting	2	2	128	$[50, 7000]$	Causal, RoPE
String Matching	3	1	64	$[50, 3000]$	Causal, RoPE
SPP	$4, 40$	4	256	$[10, 30]$	Bidir. enc.
MinCut	$4, 40$	4	128	$[10, 25]$	Decoder only
CFG	12	4	128	$[10, 22]$	MP, Bidir. enc.

- Depth $\mathbf{L}$; heads $\mathbf{H}$; embedding dimension $\mathbf{d}_{\text{mod}}$.
- Rotary positional encoding (RoPE); causal masking; bi-directional/non-causal encoder (Bidir. enc.); mean pooling (MP).
- Graph instances sampled from random geometric graphs (RGGs) near critical connectivity.

Complexity Upper Bounds

Regime	Evaluation	Inference Complexity	Evaluation Error	EPTHS
NNGP/NTK	Kernel	$\mathbf{O(PN_{MC}T^3)}$	$\mathbf{\sqrt{P/N_{MC}}}$	$\mathbf{O(T^{3+\epsilon})}$
	Forward-pass	$O(NT^2)$	$\mathbf{P^\gamma/N}$	$\mathbf{O(T^{2+\epsilon})}$
Rich learning	Forward-pass	$O(NT^2)$	$\mathbf{P^\gamma/N}$	$\mathbf{O(T^{2+\epsilon})}$

- P datapoints; N finite width; N_{MC} Monte Carlo samples; $\gamma \geq 1$; $\epsilon \geq 0$.
- Kernel evaluation estimation using Monte Carlo integration.
- **Theorem** gives EPTHS ceiling for infinite-width transformers.
- Forward-pass improves complexity upper bound under **Convergence Assumption**.

Theory vs. Experiments

Combinatorial Task	EPTHS	Theory Prediction	Experimental Evidence
Induction	$O(T)$	Allowed	Capture
Sorting	$O(T)$	Allowed	Capture
String Matching	$O(T)$	Allowed	Avg. capture
SPP	$O(T^{1+\epsilon})$	Allowed	No capture
MinCut	$O(T^{2+\epsilon})$	Allowed	No capture
CFG	$O(T^3)$	Ruled out under $O(T^{2+\epsilon})$	No capture

- Theory gives a necessary condition for capture, not a sufficient one.
- Future work: refine for SPP and MinCut.

References

[1] Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 342–350, 2009.
[2] Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: NNGP and NTK for deep attention networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 4376–4386. PMLR, 2020.
[3] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 8571–8580, 2018.
[4] Jaehoon Lee, Yasaman Bahri, Roman Novak, Sam Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018.
[5] Gadi Naveh, Oded Ben David, Haim Sompolinsky, and Zohar Ringel. Predicting the outputs of finite deep neural networks trained with noisy gradients. *Phys. Rev. E*, 104:064301, Dec 2021.