InfoFlow: A Framework for Multi-Layer Transformer Analysis

Penghao Yu
Department of Mathematics
National University of Singapore
penghaoyu@u.nus.edu

Haotian Jiang
Institute for Functional Intelligent Materials
National University of Singapore
haotian@nus.edu.sg

Zeyu Bao
Department of Mathematics
National University of Singapore
zeyu@u.nus.edu

Qianxiao Li
Department of Mathematics
Institute for Functional Intelligent Materials
National University of Singapore
qianxiao@nus.edu.sg

Abstract

While the approximation properties of single-layer Transformer architectures have been studied in recent works, a rigorous theoretical understanding of the multi-layer setting remains limited. In this work, we first establish a depth separation that multi-layer Transformers possess fundamentally different approximation capabilities from single-layer ones on certain retrieval tasks: any single-layer Transformer requires least $\Omega(\varepsilon^{-k})$ parameters to achieve precision ε, where k grows linearly with sequence length T, whereas a two-layer Transformer with a single head per layer achieves the same approximation precision with at most $O(\varepsilon^{-1})$ parameters. We further identify two structural mechanisms underlying multi-layer approximation. Specifically, softmax attention can only efficiently retrieve the token attaining the maximum attention score, incurring exponential-in-length parameter cost for k-th largest retrieval with $k \geq 2$. Moreover, the parameter cost of decoding coupled information scales with the size of the retrieved token set. Motivated by these findings, we propose **InfoFlow**, a framework for multi-layer Transformers approximating retrieval-type targets. The framework tracks an information set of accessible input positions at each token and layer, assigning an explicit approximation rate to each mode of information propagation. This abstraction recovers known approximation bounds, remains consistent with experimental observations on trained networks, and yields concrete predictions in settings where direct theoretical analysis is currently intractable. Our results provide a principled framework for reasoning about the approximation efficiency of multi-layer Transformers on retrieval-type targets.

1 Introduction

The Transformer architecture [Vaswani et al., 2017] has become the dominant framework for sequence modeling, achieving remarkable empirical success across natural language processing [Devlin et al., 2019, Brown et al., 2020], computer vision [Dosovitskiy et al., 2021]. Despite this empirical success, the theoretical understanding of its working principles remains incomplete. Progress has so far focused primarily on the single-layer setting: universality results have been established by Yun et al. [2020a,b], explicit approximation rate estimates for single-layer Transformers were derived in Jiang and Li [2024], and the role of attention head count was characterized in Yu et al. [2026], showing that insufficient head count incurs parameter cost scaling at least as $O(\varepsilon^{-cT})$. However, the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

approximation theory of multi-layer Transformers remains largely open. The analytical techniques developed for the single-layer case do not extend directly to multi-layer settings, where the interplay between attention layers and feed-forward networks across depth introduces substantial additional complexity. To make this gap precise, we establish a depth separation result: there exists a target such that any single-layer Transformer requires parameter cost growing with sequence length T, while a two-layer Transformer with one head per layer achieves the same approximation precision ε with parameter count at most $O(\varepsilon^{-1})$.

Motivated by the difficulty of direct theoretical analysis in the multi-layer setting, we develop a framework, which we call **InfoFlow**, in the spirit of effective theories in the physical sciences. Just as thermodynamics predicts the phase transition of water without quantum mechanical analysis of individual molecules, **InfoFlow** provides a tractable abstraction of multi-layer Transformer computation that retains sufficient structure to yield principled predictions on retrieval-type target approximation. Rather than tracking hidden representations, **InfoFlow** associates to each token position t and layer l an information set $I(t, l)$, defined as the collection of input positions whose information remains accessible, together with a parameter cost law governing each mode of information propagation. The design of **InfoFlow** is grounded in two structural results established in this work: softmax attention can only efficiently retrieve the token attaining the maximum attention score, with parameter cost growing exponentially with T for any other token, and the cost of decoding aggregated information from a token state scales exponentially with the number of input tokens contributing to it.

We validate **InfoFlow** both theoretically and experimentally. On the theoretical side, it recovers known single-layer approximation bounds and is consistent with the depth separation result. On the experimental side, it correctly predicts the intrinsic dimension phenomenon in two-layer Transformers and explains the approximation difficulty of higher-order retrieval tasks such as the triangle-center problem. We further evaluate these predictions on bAbI question-answering tasks, obtaining additional qualitative support for the framework. Our main contributions are as follows. **I**: We establish a depth separation result between single and two-layer Transformers (Theorem 1), a precise characterization of the retrieval limitation of softmax attention (Theorem 2), and an estimate of the parameter cost of decoding aggregated information (Theorem 3). **II**: We propose **InfoFlow**, a framework for Transformer approximation of retrieval-type targets comprising an information set update rule and an associated parameter cost law, grounded in the above theoretical results and consistent with experimental observations. **III**: We validate **InfoFlow** empirically, demonstrating that it correctly predicts the intrinsic dimension phenomenon and explains the approximation hardness of higher-order retrieval tasks such as the triangle-center problem, and further validating qualitatively for its predictions on bAbI question-answering tasks.

2 Related Work

The theoretical study of Transformer approximation properties has developed along two main directions: universality results and approximation rate results. Yun et al. [2020a] first established that Transformers are universal approximators of continuous sequence-to-sequence functions on compact domains, subsequently extended to sparse attention matrices [Yun et al., 2020b] and to single self-attention layers with low-rank weight matrices [Kajitsuka and Sato, 2024]. Beyond universality, explicit Jackson-type approximation rate estimates for single-layer Transformers were derived in Jiang and Li [2024], where a complexity measure governing the low-rank pairwise coupling structure of the target was identified as the key quantity. This was extended in Yu et al. [2026], which established both upper and lower approximation bounds, showing that insufficient head count incurs parameter cost scaling with sequence length T. Hahn [2020] showed that the computational power of self-attention is fundamentally limited, and that modeling certain sequential dependencies requires the number of layers or heads to grow with input length. More recently, Sanford et al. [2024a,b] established that one-layer Transformers cannot solve the induction head task without exponentially increasing model size, while two layers suffice with only logarithmic growth. A parallel line of work focuses on understanding multi-layer Transformer by tracking how information propagates through the network, rather than through formal approximation theory. Elhage et al. [2021] introduced the Transformer Circuits framework, which decomposes the residual stream into communication channels and identifies specific circuits responsible for particular computational behaviors. **InfoFlow**, by contrast, is a theoretical approximation tool that operates prior to training and yields quantitative predictions about approximation rates and task feasibility. Several works

have proposed simplified mathematical descriptions of Transformers aimed at tractable analysis. Another line of work interprets multi-layer Transformers as discretizations of continuous differential equations or mean-field interacting particle systems [Geshkovski et al., 2023, Lu et al., 2020], where the token sequence evolves according to a continuous-time flow governed by the attention mechanism. These approaches abstract away the discrete layer structure by taking continuous limits of the hidden representations, and focus on analyzing the dynamical properties of the resulting flow. **InfoFlow** is developed in the same spirit, but abstracts multi-layer Transformer computation through discrete information sets rather than continuous limits, and focuses on predicting approximation rates and task feasibility rather than analyzing dynamical properties.

3 Preliminaries and Mathematical Formulations

We introduce the multi-layer Transformer architecture and establish the notation used throughout the paper. We consider the input space $\mathcal{X}^T = \{x(s) \in [0,1]^d : s \in [T]\}$, where $[T] = \{1, \dots, T\}$ and T denotes the sequence length. The output is a single vector $y \in \mathbb{R}^o$. Each token is mapped into $\mathbb{R}^{E_1}$ by a trainable encoder. When positional encoding is used, the encoder takes the form $P_\phi : [0,1]^d \times [T] \to \mathbb{R}^{E_1}$, $(x, s) \mapsto P_\phi(x, s)$, depending on both token content and position, with standard choices including additive encoding $P_\phi(x(s), s) = \mathrm{Emb}(x(s)) + p(s)$ and augmented encoding $P_\phi(x(s), s) = (\mathrm{Emb}(x(s)), p(s))$. Without positional encoding, we use $P_\phi : [0,1]^d \to \mathbb{R}^{E_1}$, $x \mapsto P_\phi(x)$. In both cases, we append a trainable classification token $c_1 \in \mathbb{R}^{E_1}$ and set $X_1[T] = \{P_\phi(X_T)_{t=1}^T, c_1\} \in \mathbb{R}^{E_1 \times (T+1)}$.

For an L-layer Transformer, we denote by h_l the number of heads in the l-th layer, n_l the per-head embedding dimension, and $E_l = h_l n_l$ the total embedding dimension of the l-th layer. For $l = 1, \dots, L-1$, the l-th attention layer takes $X_l[T] = \{x_l(1), \dots, x_l(T), c_l\} \in \mathbb{R}^{E_l \times (T+1)}$ as input and produces $X_l'[T] = \{x_l'(1), \dots, x_l'(T), c_l'\} \in \mathbb{R}^{E_l \times (T+1)}$, where

$$\begin{aligned} x_l'(t) &= x_l(t) + W_{O,l} \,\mathrm{Concat}_{i=1}^{h_l}\Big(\sum_{s=1}^{T} \sigma\big[(W_{Q,i,l} x_l(t))^\top W_{K,i,l} x_l(s)\big] W_{V,i,l} x_l(s)\Big), \\ c_l' &= c_l + W_{O,l} \,\mathrm{Concat}_{i=1}^{h_l}\Big(\sum_{s=1}^{T} \sigma\big[(W_{Q,i,l} c_l)^\top W_{K,i,l} x_l(s)\big] W_{V,i,l} x_l(s)\Big), \end{aligned} \tag{1}$$

with query/key projections $W_{Q,i,l}, W_{K,i,l} \in \mathbb{R}^{n_l \times E_l}$, value projection $W_{V,i,l} \in \mathbb{R}^{n_l \times E_l}$, output projection $W_{O,l} \in \mathbb{R}^{E_l \times E_l}$, and residual connections. The softmax with scaling factor $\beta > 0$ is defined by $\sigma[\rho](s) = \frac{\exp(\beta\, \rho(s))}{\sum_{t'=1}^{T} \exp(\beta\, \rho(t'))}$. Between layers, a two-layer feed-forward network $F_l : \mathbb{R}^{E_l} \to \mathbb{R}^{E_{l+1}}$ with hidden-layer width w_l maps the outputs of the l-th attention layer to the inputs of the $(l+1)$-th layer via $x_{l+1}(t) = F_l(x_l'(t))$ and $c_{l+1} = F_l(c_l')$, with sigmoidal activation functions that are 1-Lipschitz. The final layer takes $X_L[T] = \{x_L(1), \dots, x_L(T), c_L\} \in \mathbb{R}^{E_L \times (T+1)}$ as input and produces the output

$$y = \hat{F}\Big(c_L + W_{O,L} \,\mathrm{Concat}_{i=1}^{h_L}\Big(\sum_{s=1}^{T} \sigma\big[(W_{Q,i,L} c_L)^\top W_{K,i,L} x_L(s)\big] W_{V,i,L} x_L(s)\Big)\Big), \tag{2}$$

where $\hat{F} : \mathbb{R}^{E_L} \to \mathbb{R}^o$ is a feed-forward block.

We then formalize the retrieval-type targets considered throughout this work. Intuitively, a retrieval-type target selects a small subset of relevant tokens and computes its output from the retrieved information.

Definition 1 (Retrieval-type target)**.** *Let $F : \bigcup_{T \geq 1} \mathcal{X}^T \to \mathbb{R}$ be a target defined on sequences of arbitrary length, and let $F_T := F|_{\mathcal{X}^T}$. Assume that each F_T is differentiable almost everywhere. At a differentiable point $X_T = (x(1), \dots, x(T)) \in \mathcal{X}^T$, define the active index set of F_T by*

$$I_{F_T}(X_T) := \big\{t \in [T] : \nabla_{x(t)} F_T(X_T) \neq 0\big\}. \tag{3}$$

We call F a retrieval-type target if its active-index-set size is uniformly bounded across T, namely,

$$D_0(F) := \sup_{T \geq 1} \operatorname*{ess\,sup}_{X_T \in \mathcal{X}^T} |I_{F_T}(X_T)| < \infty \tag{4}$$

4 Structural Mechanisms of Multi-Layer Transformer Approximation

We now turn to the approximation properties of the multi-layer Transformer defined above. The central questions are how the parameter cost of ε-approximation scales with the architectural parameters T, L, E_l, h_l, and which complexity measures of the target determine this scaling. We show that these questions cannot be addressed by a direct extension of single-layer results, and then identify the structural mechanisms underlying multi-layer approximation.

4.1 Depth Separation Between Single-Layer and Multi-Layer Transformers

A natural starting point for the analysis of multi-layer Transformers is to ask whether their approximation properties can be obtained by extending single-layer results. We show that this is not the case. In fact, single-layer and two-layer Transformers can exhibit fundamentally different approximation behaviors on certain retrieval targets, as the following theorem demonstrates.

Theorem 1 (Informal)**.** *There exists a target $F : \mathcal{X}^T \to \mathbb{R}$ satisfying the following.*

1. *A single-layer Transformer with H heads and per-head dimension n must incur parameter count at least $\Omega(\varepsilon^{-k(T)})$ with $k(T) = \frac{T-2H}{H(n+1)+1} - 1$ to ε-approximate F.*

2. *For every $\varepsilon > 0$, there exists a two-layer Transformer with one head per layer and per-head dimension 6 with parameter count at most $O(\varepsilon^{-1})$ that ε-approximates F.*

We defer the formal statement and proof to Appendix A.1. The depth separation in Theorem 1 reflects a fundamental architectural difference between single-layer and multi-layer Transformers. A single-layer Transformer performs only one round of attention, and can therefore compare each token against the sequence at most once. A two-layer Transformer, by contrast, can perform comparisons in two stages: the first layer allows each token to find its most relevant neighbor, and the second layer aggregates these local results into a global answer. For the target $F(X_T) = \min_{1\le s,t\le T} 2(1 + x(s)^\top x(t))$, this two-stage structure allows the model to decompose the global minimization over T^2 pairs into two successive minimizations each of order T, achieving $O(\varepsilon^{-1})$ approximation regardless of T. A single-layer Transformer has no such decomposition available, and must instead rely on very wide feed-forward layers to resolve the remaining comparisons, requiring parameter count at least $\Omega(\varepsilon^{-k(T)})$ where $k(T)$ grows linearly with T.

4.2 Information Propagation Mechanisms in Attention Layers

To understand the mechanism behind the depth separation established above, we examine how information propagates through the attention layers of a multi-layer Transformer. We identify three characteristic modes of information propagation, each supported by theoretical analysis or experimental evidence. These modes are not intended as an exhaustive classification of all possible attention behaviors. Rather, they capture the dominant mechanisms that emerge in practice and are amenable to theoretical characterization, forming the basis of the **InfoFlow** framework introduced in Section 5.

Mechanism I: Max-position retrieval. A first observation from two-layer Transformers is that the token state at position t after one attention layer retains information from two sources. First, the residual connection ensures that the original token $x(t)$ itself remains accessible from the attention output $x^{(1)}(t)$ by construction. Second, experiments on trained two-layer Transformers show that the information of the token at the argmax position of each attention head, namely $x(s_i(t))$ where $s_i(t) = \arg\max_s A_i(x(t), x(s))$, can be recovered from $x^{(1)}(t)$. Specifically, we measure the normalized recovery error $\mathbb{E}_{X_T}\mathbb{E}_t|g_i(x'_1(t)) - x(s_i(t))|^2/|x(s_i(t))|^2$ for the best map g_i, and find that training reduces this error by an order of magnitude compared to initialization across all configurations tested, with ratios of trained to initialized error as low as 0.08 (see Appendix C.1 for full experimental details). Furthermore, the following theorem establishes that softmax attention is fundamentally limited to efficient retrieval at the argmax position, and cannot efficiently retrieve any other token. This is further confirmed in the experiment in Appendix C.2.

Theorem 2 (Informal)**.** *For a given k with $2 \leq k \ll T$, if a softmax attention head with embedding dimension n retrieves the information of the token with k-th largest attention score with precision $\varepsilon > 0$, then it requires at least $\Omega\left(\varepsilon^{1-\frac{T-k-1}{n+1}}\right)$ parameters in the feed-forward blocks.*

We defer the formal statement and proof to Appendix A.2. The experimental observation and Theorem 2 together establish what we term the *max-position retrieval mechanism*. The residual connection and argmax concentration of softmax attention ensure that each token state retains information from its own position and from the position attaining the maximum attention score, while retrieving information from any other position incurs a parameter cost growing exponentially in T. This mechanism is a structural property of softmax attention that persists regardless of the projection matrices chosen.

Mechanism II: Global information aggregation. A single attention head may also aggregate information from the entire input sequence at one position, enabling the model to handle relationships that depend on all tokens simultaneously. As established in [Yu et al., 2026], this global aggregation mode incurs a parameter cost of $O(\varepsilon^{-T/E_l})$ in the feed-forward block following the attention layer when the embedding dimension satisfies $E_l \leq T$. This cost grows rapidly with sequence length T, reflecting the fundamental difficulty of compressing global sequence information into a fixed-dimensional token state, and suggesting that global aggregation is only effective when the sequence length is not large relative to the embedding dimension.

Mechanism III: Specific position aggregation. When positional encoding is available, a Transformer can aggregate information from a fixed set of positions determined by positional information alone, independent of the sequence content. To illustrate this mechanism, we train a one-layer one-head Transformer with positional encoding on the target $F(X_T) = x(1) + x(2) + x(3)$ for sequences of length $T = 10$. As shown in Figure 4, the CLS token's attention concentrates with weight $1/3$ on each of positions 1, 2, 3 and weight 0 on all remaining positions, a pattern that is stable across samples regardless of token values. This confirms that the model selects positions by index alone rather than by content, a mode of information propagation that is unavailable without positional encoding, as attention weights in that setting depend only on token content and thus cannot implement position-specific selection. Full experimental details are given in Appendix C.3.

We demonstrate the three mechanisms in Figure 3. They together characterize the dominant ways in which attention layers move information across token positions. This evidence motivates a framework that tracks which input positions contribute to each token state across positions and layers. We formalize this framework in the next section.

5 InfoFlow: A framework for analyzing Multi-Layer Transformers

Motivated by the three information propagation mechanisms identified in Section 4, we propose **InfoFlow**, a framework that replaces Transformer's hidden vector representations with discrete information sets, together with a parameter cost law governing each mode of propagation.

5.1 The InfoFlow Model

Instead of tracking the actual hidden vectors, we associate to each position $t \in \{1, \ldots, T+1\}$ at layer l an information set $I(t, l) \subset \{1, \ldots, T+1\}$, which represents the collection of input positions whose information remains accessible from the token state $x'_l(t)$ at position t after l layers. Here the $(T+1)$-th position corresponds to the classification token c'_l. At the input layer, each token contains information from its own position only, so we set

$$I(t, 0) = \{t\}, \quad 1 \leq t \leq T, \qquad I(T+1, 0) = \varnothing. \tag{5}$$

We now define the update rule from layer l to layer $l+1$. In **InfoFlow**, we model information propagation from (t, l) to $(t, l+1)$ through the following three update rules, which are motivated by the three mechanisms identified in Section 4.

(1) Max-position retrieval. Corresponding to the max-position retrieval mechanism, we define for each head $i \in \{1, \ldots, h_{l+1}\}$ the position at layer l whose information contributes most to token t at

layer $l+1$ under head i as $s_{i,l}(t) = \arg\max_{s\in\{1,\dots,T\}} A_{l+1,i}\big(X[I(t,l)],\, X[I(s,l)]\big)$, where $A_{l+1,i}$ denotes the attention score function of the i-th head in layer $l+1$, and $X[I]$ denotes the subsequence of X_T at indices in I. The information set update rule is then $I(t,l+1) \subset \bigcup_{i=1}^{h_{l+1}} I(s_{i,l}(t),l) \cup I(t,l)$. The new token state uses only information from the argmax positions selected by each head and from its own position.

(2) Global information aggregation. Corresponding to the global aggregation mechanism, one attention head aggregates information from the entire input sequence at position t, so the information set after the update contains all input positions, $I(t,l+1) = \{1,\dots,T\}$. The new token state thus depends on the entire input sequence.

(3) Specific position aggregation. Corresponding to the specific position aggregation mechanism, which requires positional encoding, the attention head selects a fixed set of positions $I'(t,l+1)$ determined by the position t and layer $l+1$ alone, independent of the sequence content. The information set update rule is then $I(t,l+1) \subset \bigcup_{j\in I'(t,l+1)} I(j,l)$, where $I'(t,l+1)$ is determined only by t and $l+1$, not by the values of the input sequence.

Having defined the three modes of information propagation, we say an **InfoFlow** model learns a retrieval target F in Definition 1 for a given T if $I_{F_T}(X_T) \subseteq I(T+1,L)$ for $X_T \in \mathcal{X}^T$ *a.e.* in Lebesgue measure. Here $I(T+1,L)$ is the information set of the classification token in the final layer. An illustrative example is shown in Appendix E.

5.2 Parameter Cost

In practical Transformers, information accumulated in the information set $I(t,l)$ is not represented explicitly but is embedded in the token state $x_l(t) \in \mathbb{R}^{E_l}$, and recovering it requires nontrivial decoding by the feed-forward layers. We use the following theorem to quantify the parameter cost of this encoding-decoding process.

Theorem 3 (Informal)**.** *For index set I of size T_0, if we use two-layer FFN f_1 to encode $X[I]$ in $[0,1]^n$ and f_2 to decode $\hat{X}[I]$ from $[0,1]^n$. If $\hat{X}[I] = f_2 \circ f_1(X[I])$ is ε-close to $X[I]$ for all X_T, then f_1 and f_2 have parameter count of $O(\varepsilon^{1-\frac{T_0 d}{n}})$.*

We defer the formal statement and proof to Appendix A.3. Motivated by the encoding and decoding bounds in Theorem 3, we assign a parameter cost $O(\varepsilon^{1-\frac{|I(t,l+1)|d}{E_{l+1}}})$ to each **InfoFlow** update according to the information-set size $|I(t,l+1)|$ and embedding dimension E_{l+1}. This assignment models the cost of processing the information accessible at each token. The three propagation modes have the same form of parameter cost, but differ in their information-set update rules, and hence in the resulting value of $|I(t,l+1)|$. Regarding the size of $I(t,l+1)$, we then associate to each (t,l) a parameter cost for ε-approximation as follows.

1. In max position aggregation, the parameter cost is $O(\varepsilon^{1-|I(t,l+1)|d/E_{l+1}})$.
2. In global information aggregation, the parameter cost is $O(\varepsilon^{1-Td/E_{l+1}})$.
3. In specific position aggregation, the parameter cost is $O(\varepsilon^{1-|I'(t,l+1)|\max_s |I(s,l)|d/E_{l+1}})$.

Here E_l denotes the total embedding dimension at layer l. The total parameter cost of the **InfoFlow** model is the sum of parameter costs over all (t,l).

Remark. Feed-forward networks also incur a parameter cost of $O(\varepsilon^{-\gamma})$ for approximating nonlinear relations in target, where γ corresponds to the smoothness of the function. We omit this term as it is not the dominant cost in regimes where Transformer approximation is fundamentally limited.

5.3 How InfoFlow Works

Given a Transformer structure and a target F, **InfoFlow** estimates the approximation rate of F via the following two-step procedure. The key quantity linking the complexity of the task to the capacity of the **InfoFlow** model is the number of comparison, which we define as follows.

Number of Comparison of a target. To quantify the computational complexity of a target function, we introduce the notion of the *Number of Comparison* for retrieval-type targets in Definition 1. We first define the Tree of Comparison to aid the derivation of Number of Comparison.

Definition 2. *A Tree of Comparison $\mathcal{T}$ is a full binary tree, with each leaf V_l having an ordered index set I_l allowing repetition of elements and depending only on T, and each internal node V_{in} associated to a continuous function f_{in}. For each input X_T, the Tree of Comparison updates in the following way: For each internal node V_{in} with children V_{left} (with index set I_{left})and V_{right} (with I_{right}), V_{in} is associated with index set $I = \arg\max\{f_{in}(X[I_{left}]), f_{in}(X[I_{right}])\}$.*

By the structure of full binary tree, the Tree of Comparison is well-defined. We call $I_{final}(X_T, \mathcal{T})$ the index set associated to the root of $\mathcal{T}$ with input X_T, the number of leaves $N_{leaf}(\mathcal{T}) = |\{V_l : V_l \in \mathcal{T}\}|$ the size of the Tree, and $\max_{V_l} |I_l|$ the dimension of the Tree. We then define the Dimension, Order and Number of Comparison of the target F:

Definition 3. *For F with $D_0(F) < \infty$, define the Dimension of Comparison β_1 as the smallest $\beta \in \mathbb{Z}^+$ for which there exist a constant $C_0 > 0$ and an exponent $\beta' \in \mathbb{Z}^+$ with $\beta' \le \beta$ such that, for every T, there exist $D_1 \le D_0(F)$ Trees of Comparison $\mathcal{T}_1, \dots, \mathcal{T}_{D_1}$. Each tree has dimension at most β and size $N_{\text{leaf}}(\mathcal{T}_i) \le C_0 T^{\beta'}$, and they satisfy $I_{F_T}(X_T) \subseteq \bigcup_{j=1}^{D_1} I_{\text{final}}(X_T, \mathcal{T}_j)$ for $X_T \in \mathcal{X}^T$, a.e. Fixing $\beta = \beta_1$, the Order of Comparison is defined as the smallest exponent β' for which the above construction is possible. The minimum total number of pairwise comparison $N' = \sum_{j=1}^{D_1}(N_{leaf}(\mathcal{T}_j) - 1)$ over all choices of Trees is called the Number of Comparison of F.*

We focus on F with finite Dimension and Order of Comparison. Intuitively, the Number of Comparison of a target is how many tuples of tokens one needs to consider to compute the target output, and Dimension of Comparison is the maximum size of such tuples. For example, the generalized D-retrieval target $F(X_T) = F_0(\oplus_{i=1}^{D} \max_{1\le t\le T} f_i(x(t)))$ studied in Yu et al. [2026] has $\beta' = \beta_1 = 1$, with Number of Comparison upper bounded by $D(T-1)$ and lower bounded by $D(T-D)$.
However, in general, targets can involve comparison between larger tuples. A simple example is the target $F(X_T) = \min_{1\le t_1,t_2,t_3\le T} \|x(t_1) + x(t_2) + x(t_3)\|_2^2$, we can show that it has $\beta' = \beta_1 = 3$, with Number of Comparison upper bounded by $T^3 - 1$ and lower bounded by $\Omega(T^3)$. This example will be used later to show that Transformers are inherently limited in capturing such relationships in Section 6. We defer explanations of the two examples to Appendix D.

Number of Comparison of InfoFlow. To model the comparison capacity of a Transformer, **InfoFlow** accounts for comparisons through the following two mechanisms when approximating a retrieval-type target with Dimension of Comparison β_1. First, softmax attention at each position performs at most $T-1$ comparisons per head. Second, within the subsequence $X[I(t,l)]$, at most $|I(t,l)|^{\beta_1} - 1$ comparisons can be performed. Summing up across all positions in all layers, the total Number of Comparison of the **InfoFlow** model is therefore

$$\sum_{t=1}^{T}\sum_{l=1}^{L-1}\Big(|I(t,l)|^{\beta_1} - 1 + \sum_{i=1}^{h_l}(T-1)\Big) + \sum_{l=1}^{L}\Big(|I(T+1,l)|^{\beta_1} - 1 + \sum_{i=1}^{h_l}(T-1)\Big). \tag{6}$$

Two-step estimation procedure. Given a Transformer structure $(L, h_1, \dots, h_L, E_1, \dots, E_L)$ and a retrieval-type target F, the approximation rate is estimated as follows. First, the Number of Comparison of the **InfoFlow** model must be no smaller than that of the target. This yields a lower bound on $M = \max_{t,l} |I(t,l)|$, which in turn gives a parameter cost lower bound of $O(\varepsilon^{1-Md/E_l})$ for ε-approximation of F. Second, one selects an update rule for each (t,l) such that the resulting **InfoFlow** model learns the target in the sense of active index set, yielding a corresponding upper bound on the parameter cost. The two steps provide parameter-cost estimates, which we use as predictions for Transformer approximation prior to actual training. Their applicability still depends on whether the actual Transformer computation is captured by the information-set updates and cost assignments of **InfoFlow**.

6 Predictions and Experimental Validation

Before presenting new predictions, we first verify that **InfoFlow** is consistent with existing theoretical results from Yu et al. [2026] and Yun et al. [2020a].

For the generalized D-retrieval target, a single-layer Transformer with s heads using only max position aggregation has at most $s(T-1)$ comparisons, while the target requires at least $D(T-D)$ comparisons. **InfoFlow** therefore predicts that at least $s \geq D$ heads are needed for effective approximation, which matches Theorem 2.2 of Yu et al. [2026].

Alternatively, a single-layer one-head Transformer can use global information aggregation to ε-approximate the generalized D-retrieval target with $O(\varepsilon^{-Td/E})$ parameters. This matches Theorem 2.3 of Yu et al. [2026] and the universality results of Yun et al. [2020a].

Having verified consistency with known results, we first turn to two cases where direct theoretical analysis is currently intractable but **InfoFlow** could provide correct predictions, and then test these predictions on the question-answering task.

6.1 Intrinsic Dimension Phenomenon in Two-Layer Transformers

We consider the permutation-invariant target

$$F(X_T) = \sum_{i=1}^{D} \max_{1 \leq s,t \leq T} x(s)^\top A_i x(t), \tag{7}$$

where $A_i \in \mathbb{R}^{d \times d}$ are distinct matrices. We next apply **InfoFlow** to analyze how effective transformers can approximate this target.

Following the estimation in **InfoFlow**, we have the following proposition.

Proposition 1 (Informal)**.** The Number of Comparison of F is upper bounded by $DT^2 + o(T^2)$; The Dimension and Order of Comparison of F are $\beta_1 = \beta' = 2$.
The **InfoFlow** model of a two-layer Transformer with (h_1, h_2) heads using only max-position aggregation has Number of Comparison at most $O(h_1 T^2)$, to learn F we need $h_1 \geq D$. Also, there exists a two-layer Transformer with (D, D) heads using only max position aggregation that learns the target.

We defer the detailed explanation to Appendix B.1. Together, the proposition indicates that D is the intrinsic dimension of the target, in the sense that $h_1 = h_2 = D$ is the minimal head count for effective approximation, which we examine experimentally below.

Experimental verification. We train two-layer Transformers with sequence length $T = 64$ on the target (7) for $D = 2, 3, 4, 5, 6$, comparing three head configurations for each D: (D, D), $(D-1, D)$, and $(D, D-1)$. As shown in Figure 1a, the results exhibit a sharp phase transition at $h_1 = h_2 = D$. The (D, D) configuration achieves best validation NMSE on the order of 10^{-5} across all values of D, while the $(D-1, D)$ and $(D, D-1)$ configurations achieve NMSE on the order of 10^{-3} to 10^{-2}, a gap of two to three orders of magnitude. This sharp separation across all tested values of D provides strong empirical confirmation of the **InfoFlow** prediction.
Beyond validating the framework, this result also reveals a structural insight about multi-layer Transformer approximation: even in the multi-layer setting, the number of heads remains a critical architectural parameter, and must match the intrinsic dimension of the target for effective approximation. This suggests an extension of the single-layer finding of Yu et al. [2026], where insufficient head count leads to exponentially growing parameter cost, to the multi-layer setting: a two-layer Transformer with fewer than D heads in the first layer cannot approximate a target of intrinsic dimension D efficiently, regardless of the second layer's configuration. Further training details and ablation results are given in Appendix C.4.

6.2 Approximation Hardness of Higher-Order Retrieval Tasks

The intrinsic dimension experiment demonstrates that when a retrieval-type target requires pairwise comparisons, a two-layer Transformer with sufficiently many heads can approximate it efficiently. However, it is critical to observe that not all targets can be handled efficiently by a fixed Transformer architecture as T grows. The key insight from **InfoFlow** is that when a target requires three or more simultaneous comparisons rather than pairwise ones, the Number of Comparison of the target grows as a higher power of T, and no fixed Transformer architecture can match this complexity without requiring number of parameters to grow exponentially with T. This is consistent with the

representational limitations of Transformers studied in Sanford et al. [2023].We then propose the the following proposition to demonstrate this phenomena.

Proposition 2 (Informal)**.** Under the **InfoFlow** parameter cost assignments, if a target F have Order of Comparison $\beta' > 2$, then for an L-layer H-heads-per-layer Transformer with embedding dimension E, there exists a constant $C > 0$ such that it requires a parameter cost of $\Omega(\varepsilon^{1-\frac{CT^{(\beta'-1)/\beta_1}}{LE}})$ for ε-approximation of F_T.

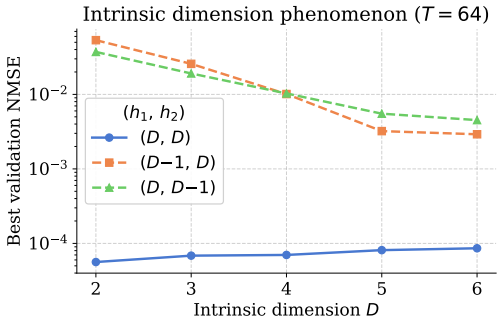
We defer the detailed explanation to Appendix B.2. This proposition predicts that any target F with Order of Comparison $\beta' > 2$ cannot be approximated effectively by any transformer of fixed size as T grows. To illustrate this phenomenon, we consider the triangle-center target,

$$F(X_T) = \min_{1 \le t_1, t_2, t_3 \le T} \|x(t_1) + x(t_2) + x(t_3)\|_2^2, \tag{8}$$

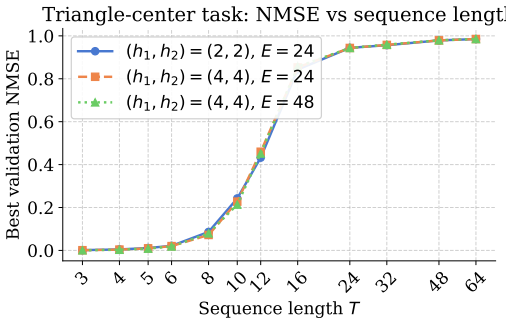
which have Order of Comparison $\beta' = 3$ according to Appendix D.1. Unlike the target in (7), which requires only pairwise comparisons between tokens and has number of comparison $O(T^2)$, the triangle-center target requires three tokens to be compared simultaneously, resulting in Number of Comparison $\Omega(T^3)$. This cubic growth in comparison complexity is the source of the fundamental difficulty: **InfoFlow** predicts that this target cannot be approximated effectively by any Transformer of fixed size as T grows, regardless of the depth, number of heads, or embedding dimension.

InfoFlow prediction. Following Proposition 2, for the target F defined in (8), it requires an L-layer H-heads-per-layer Transformer with embedding dimension E a parameter cost of $O(\varepsilon^{1-\frac{T^{2/3}}{6LE}})$ for ε-approximation of F. Since the exponent $\frac{T^{2/3}}{LE}$ grows with T, for any fixed L and E, the parameter cost of approximating F must be super-polynomial in $1/\varepsilon$ for large T, regardless of the number of heads or embedding dimension. This demonstrates a fundamental limitation that cannot be overcome by increasing model size.

Experimental verification. We train two-layer Transformers of three different sizes on the target (8) for sequence lengths $T \in \{3, 4, 5, 6, 8, 10, 12, 16, 24, 32, 48, 64\}$. We consider the following head configurations: $(h_1, h_2) = (2, 2)$ with embedding dimension 24, $(h_1, h_2) = (4, 4)$ with embedding dimension 24, and $(h_1, h_2) = (4, 4)$ with embedding dimension 48. As shown in Figure 1b, all three configurations exhibit rapidly growing NMSE as T increases, with NMSE exceeding 0.85 at $T = 16$ and approaching 1.0 for $T \ge 32$, regardless of model size. The three configurations perform almost identically across all values of T, indicating that increasing the number of heads or the embedding dimension does not improve approximation within the tested range. These results are consistent with the **InfoFlow** prediction that the parameter cost of approximating F grows with T for any fixed architecture. Further training details and ablation results are given in Appendix C.5.



(a) Best validation NMSE for two-layer Transformers trained on the target (7), showing a sharp phase transition at $h_1 = h_2 = D$ that confirms that D is the intrinsic dimension of the target.



(b) Best validation NMSE against sequence length T for three Transformer configurations trained on (8), consistent with the **InfoFlow** prediction that approximation difficulty grows with T regardless of model size.

Figure 1: Experimental validation of the intrinsic dimension phenomenon (left) and the approximation hardness of the triangle-center task (right).

6.3 Validation on bAbI Question Answering

To test whether the predictions extend beyond synthetic numerical targets, we consider Tasks 1–3 of the bAbI question-answering benchmark [Weston et al., 2015]. Each question is annotated with the supporting facts needed to answer it; the three tasks require one, two, and three supporting facts, respectively. We refer to this number as the *arity* and view it as the number of positions that must be retrieved. Each fact is represented as one input token, and the sequence length T is varied by adding answer-preserving distractor facts while keeping the entity pool fixed. We train Transformers from scratch with $h = 4$ heads, embedding width $E = 16$, and depth $L \in \{1, 2, 3, 4\}$. Table 1 reports the mean training accuracy across 10 seeds, and arity 1 is omitted from the table because its mean training accuracy is greater than 0.99 for all L and T. Further details are in Appendix C.6.

Table 1: Mean $\pm$ standard deviation of training accuracy across $N = 10$ seeds.

Arity 2

$L\backslash T$	20	40	80	128
1	.40±.08	.53±.08	.56±.09	.48±.06
2	.99±.00	.99±.01	.95±.12	.83±.26
3	1.00±.00	1.00±.00	1.00±.00	.86±.22
4	1.00±.00	1.00±.00	1.00±.00	.95±.14

Arity 3

$L\backslash T$	20	40	80	128
1	.45±.01	.39±.03	.53±.01	.48±.03
2	.53±.15	.84±.20	.69±.14	.53±.09
3	.80±.23	.92±.19	.91±.12	.56±.23
4	.80±.24	1.00±.00	.93±.11	.68±.25

The one-layer model fits the arity-1 task but not the arity-2 or arity-3 tasks, whereas two layers attain at least 0.99 accuracy on arity 2 for every T. This staged-retrieval advantage is consistent with the depth separation in Theorem 1. For arity 3, models of all depth L degrades at larger T. This dependence on sequence length, and its mitigation by greater depth, agrees qualitatively with the higher-order prediction of Proposition 2, instantiated by the triangle-center target in (8).

7 Discussion and Outlook

In this work, we proposed **InfoFlow**, a coarse-grained framework for analyzing the approximation properties of multi-layer Transformers on retrieval-type targets. Motivated by the depth separation result, the three structural mechanisms of information propagation and the Number of Comparison of retrieval-type targets, **InfoFlow** associates to each token position and layer an information set together with a parameter cost law, yielding a tractable abstraction that recovers known approximation bounds and yields concrete predictions where direct theoretical analysis is intractable. Experimental validation on the intrinsic dimension phenomenon, the triangle-center task and the question answering task further confirms the predictions of **InfoFlow**, demonstrating its utility as a tool for reasoning about approximation efficiency prior to training. Several directions remain open for future investigation. On the theoretical side, a more complete characterization of information propagation modes would strengthen the foundation of **InfoFlow**, and extending it to practical Transformer variants is an important open problem. On the practical side, **InfoFlow** can potentially guide architecture design by predicting which tasks are well-suited to a given Transformer structure and informing hyperparameter choices such as head count and embedding dimension.

Limitation The parameter cost law in **InfoFlow** does not account for the role of attention weight matrix rank in the attention score computation, a limitation also present in Yu et al. [2026] and studied by Amsel et al. [2025]. Incorporating rank into the cost law is a natural direction for future work. Additionally, the three propagation modes are motivated by synthetic tasks, and their relevance to real-world Transformer behavior requires further validation. Finally, the current framework applies to retrieval-type targets where the active index set does not grow with T, and extending it to more general targets would require new theoretical tools.

8 Acknowledgement

This project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-RP-2022-028).

References

Noah Amsel, Gilad Yehudai, and Joan Bruna. Quality over Quantity in Attention Layers: When Adding More Heads Hurts. In *The Thirteenth International Conference on Learning Representations*, April 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019.

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. In *International Conference on Learning Representations*, May 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and Tom Conerly. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on Transformers. *Bulletin of the American Mathematical Society*, 2023.

Michael Hahn. Theoretical Limitations of Self-Attention in Neural Sequence Models. *Transactions of the Association for Computational Linguistics*, 8:156–171, January 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00306.

Haotian Jiang and Qianxiao Li. Approximation Rate of the Transformer Architecture for Sequence Modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

Tokio Kajitsuka and Issei Sato. Are Transformers with One Layer Self-Attention Using Low-Rank Weight Matrices Universal Approximators? In *The Twelfth International Conference on Learning Representations*, May 2024.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-yan Liu. Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, February 2020.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Representational Strengths and Limitations of Transformers. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. One-layer transformers fail to solve the induction heads task, August 2024a.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 43276–43327, Vienna, Austria, July 2024b. JMLR.org.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Penghao Yu, Haotian Jiang, Zeyu Bao, Ruoxi Yu, and Qianxiao Li. The Effect of Attention Head Count on Transformer Approximation. In *The Fourteenth International Conference on Learning Representations*, March 2026. doi: 10.48550/arXiv.2510.06662.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are Transformers universal approximators of sequence-to-sequence functions? In *arXiv:1912.10077 [Cs, Stat]*, February 2020a.

Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. O(n) Connections are Expressive Enough: Universal Approximability of Sparse Transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 13783–13794. Curran Associates, Inc., 2020b.

A Formal statements and Proofs of Theorems

A.1 Formal statement and proof of Theorem 1

We choose $\mathcal{X} = [-1, 1]^3$ and consider the following task: the input $X_T \in \mathcal{X}^T$, the target $F : \mathcal{X}^T \to \mathbb{R}$ is defined as

$$F(X_T) = \min_{1 \le s,t \le T} 2(1 + x(s)^T x(t)) \tag{9}$$

For a model $M : \mathcal{X}^T \to \mathbb{R}$, define

$$\delta(M) = \sup_{X_T \in \mathcal{X}^T} |F(X_T) - M(X_T)| \tag{10}$$

For the theorem to hold, we also post the following constraints on the models.

- The embedding P_ϕ satisfies

  $$\|P_\phi(x(s), s)\|_2 \le 1, \qquad \forall\, s \in [T],\ X_T \in \mathcal{X}^T,$$

  ensuring embedded inputs remain uniformly bounded.
- All weights in all feed-forward blocks F_l and $\hat{F}$ and entries of the attention matrices $\{W_{Q,i,l}, W_{K,i,l}, W_{V,i,l}\}, W_{O,l}$ are bounded in magnitude by 1, ensuring stability of the model.

Under these assumptions, we then propose the formal theorem showing that this task F is easy for two-layer transformer and hard for single layer transformer.

Theorem 4 (Formal Version of Theorem 1)**.** *The following two cases holds.*

- *Let $M_1 : \mathcal{X}^T \to \mathbb{R}$ be a single-layer transformer with H heads and per-head embedding dimension n defined above. If M_1 satisfies that $\delta(M_1) \le \varepsilon$, then M_1 have parameter count of the order $\Omega(\varepsilon^{-k})$, where $k = \frac{T-H-1}{H(n+1)+1} - 1$.*
- *$\forall \varepsilon > 0$, there exists a two-layer transformer M_2 with 1 head in both layers and per-head dimension 6 defined above, with parameter count at most of order $O(\varepsilon^{-1})$ such that $\delta(M_2) \le \varepsilon$*

Proof Sketch The proof of the first part of Theorem1 follows the methods and notations of Theorem 2.2 in [Yu et al., 2026], by using the Pigeonhole Theorem to construct two subsequences which are indistinguishably close after softmax attention, on the order of $O(\varepsilon^{k+1})$, while their target values differs by 4ε. Then we extend them to full sequences to complete the proof.

The second part of Theorem 1 is proven with explicit construction.

Proof. **First Part of Theorem 1** We consider only the nontrivial case, where $T_0 = T - H - 1 > 0$. We use the following notations (let $E = nH$):

Notations For each head $i = 1, \ldots, s$, define the attention weight function

$$\lambda_i(x, t) \;=\; \exp\big(\beta\,(W_{K,i,1} P_\phi(x, t))^\top W_{Q,i,1} c_0\big), \quad i = 1, \ldots, H$$

and the value mapping

$$v_i(x, t) \;=\; W_{V,i,1} P_\phi(x, t) \;\in\; \mathbb{R}^n, \quad i = 1, \ldots, H$$

where $\gamma > 0$ is the softmax scaling factor, $W_{Q,i,1}, W_{K,i,1} \in \mathbb{R}^{n \times E}$ are the query and key projections, and $W_{V,i,1} \in \mathbb{R}^{n \times E}$ is the value projection for head i.

We denote by Q_0 the following set

$$Q_0 = \{x : x \in \mathbb{S}^2 \text{ with } x_1 \in [-1/2, 0], x_2 \in [-1/2, 0], x_3 < 0\} \tag{11}$$

We choose $H + 1$ points $b_1, \ldots, b_{H+1} \in Q_0$ satisfying the followings:
(1)$\|b_i - b_j\|_2 \ge 4L, \forall i \ne j$,

(2)$B(b_i, L) \cap \mathbb{S}^2 \subset Q_0$.
It is clear that there exists $L = \frac{C}{H}$ for positive absolute constant C for this to hold. We choose such L and such $b_1, \ldots, b_{H+1}$.

Let $K_i = -B(b_i, L) \cap Q_0$, $i = 1, \ldots, H+1$, and let $P_0 = \bigcup_{i=1}^{H+1} K_i$, we recursively define the following pairs (y_i, t_i) for $i = 1, \ldots, H$ as follows.

- For the first head,
$$(y_1, t_1) \;=\; \arg\max_{\substack{y \in P_0 \\ t \in [T]}} \lambda_1(y, t).$$
- For $j > 1$,
$$(y_j, t_j) \;=\; \arg \max_{\substack{y \in P_0 \\ t \in [T] \\ t \notin \{t_1, \ldots, t_{j-1}\}}} \lambda_j(y, t).$$
- If maximum can be obtained at multiple (y, t), then choose one of them.

Let $Y = \{y_1, \ldots, y_s\}$ (Consider Y as a "set" allowing repetition). Since the sets $K_1, \ldots, K_{H+1}$ are pairwise disjoint and $H < H + 1$, there exists at least one index $i \in \{1, \ldots, H+1\}$ such that
$$K_i \cap Y \;=\; \varnothing.$$
Without loss of generality, we assume that $i = 1$, and $t_j = T - j$ for $j = 1, \ldots, H$.

Sequences to be considered We then consider constructing the two sequences that are close after attention layer.

We denote by $M = T_0 \lfloor \frac{L^2}{4T_0\varepsilon} \rfloor$. And for $i = 1, \ldots, M$, we construct points p_i satisfying the following:
$$p_i \in \{x : x \in K_1, \|x - (-b_1)\|_2^2 = \frac{(i-1)L^2}{M}\} \tag{12}$$
As it is clear that $\{x : x \in K_1, \|x - (-b_1)\|_2^2 = \frac{(i-1)L^2}{M}\}$ must be non-empty, we can pick such p_i for $i = 1, \ldots, M$.

With $\{p_i\}$ constructed, we construct sets U_j for $j = 1, \ldots, T_0$:
$$U_j = \{p_i : \frac{(j-1)M}{T_0} < i \leq \frac{jM}{T_0}\} \tag{13}$$
We have that $|U_j| = \frac{M}{T_0} = \lfloor \frac{L^2}{4T_0\varepsilon} \rfloor = O(1/\varepsilon)$.

Existence of Two Distinct Sequences We then claim that there exists two distinct sequences $Z_{1,T} = z_1(1), \ldots, z_1(T)$ and $Z_{2,T} = z_2(1), \ldots, z_2(T)$ satisfying the following:

- $z_1(T) = z_2(T) = b_1$.
- $z_1(T-i) = z_2(T-i) = y_i$, for $i = 1, \ldots, H$.
- $z_1(t), z_2(t) \in U_t$, for $t = 1, \ldots, T_0$.
- $\left\| \frac{\sum_{t=1}^T \lambda_i(z_1(t),t) v_i(z_1(t),t)}{\sum_{t=1}^T \lambda_i(z_1(t),t)} - \frac{\sum_{t=1}^T \lambda_i(z_2(t),t) v_i(z_2(t),t)}{\sum_{t=1}^T \lambda_i(z_2(t),t)} \right\|_2 \leq \frac{\varepsilon^{k+1}}{6T}$, for $i = 1, \ldots, H$.
- $\frac{\sum_{t=1}^T \lambda_i(z_1(t),t)}{\sum_{t=1}^T \lambda_i(z_2(t),t)} \in \left[1/(1 + \frac{\varepsilon^{k+1}}{24T^2}),\ 1 + \frac{\varepsilon^{k+1}}{24T^2}\right]$.

We then justify the existence of such two distinct sequence by comparing the order of $1/\varepsilon$ on both side of the conditions.

- We first consider the order on the choice space of $Z_{1,T}, Z_{2,T}$. As each U_t have size $O(1/\varepsilon)$, $t = 1, \ldots, T_0$, Then the size of the choice space is $O(1/\varepsilon^{T_0})$.

- We then consider the order on the constraint space. Noticing that $\left\| \frac{\sum_{t=1}^{T} \lambda_i(z_j(t),t) v_i(z_j(t),t)}{\sum_{t=1}^{T} \lambda_i(z_j(t),t)} \right\|_2 \leq 1$ as it is a convex combination of vectors of norm no greater than 1, $j = 1, 2$. Thus discretizations to achieve accuracy $O(\varepsilon^{k+1})$ would lead to $O(1/\varepsilon^{(k+1)nH})$ possibilities across H heads. Let $M_i = \max\{\max_{1\leq j\leq H} \lambda_i(y_j, t_j), \lambda_i(b_1, T)\}$ for $i = 1, \ldots, H$, we have that $M_i \leq \sum_{t=1}^{T} \lambda_i(z_j(t), t) \leq TM_i$, $j = 1, 2$. Therefore the discretizations to achieve accuracy $O(\varepsilon^{k+1})$ of the term $\frac{\sum_{t=1}^{T} \lambda_i(z_1(t),t)}{\sum_{t=1}^{T} \lambda_i(z_2(t),t)}$ leads to $O(1/\varepsilon^{(k+1)H})$ possibilities across H heads. Thus it reaches to a total of $O(1/\varepsilon^{(k+1)(n+1)H})$ possibilities.

As $T_0 = (k+1)(H(n+1)+1) > (k+1)H(n+1)$, we have

$$O(\frac{1}{\varepsilon^{T_0}}) \gg O(\frac{1}{\varepsilon^{(k+1)(n+1)H}}) \tag{14}$$

Therefore, by the pigeonhole principle, there exists two distinct sequences $Z_{1,T}$ and $Z_{2,T}$ satisfying all the conditions in A.1.

Consider index set $J = \{t : z_1(t) \neq z_2(t)\}$. By $Z_{1,T} \neq Z_{2,T}$, we have that $J \neq \varnothing$, and by construction of $Z_{1,T}, Z_{2,T}$, we have that $J \subseteq \{1, \ldots, T_0\}$. We then construct two sequences $W_{1,T}, W_{2,T}$ from $Z_{1,T}, Z_{2,T}$.

- For $t \in J$, $w_1(t) = z_1(t)$, $w_2(t) = z_2(t)$.
- For $t \in \{1, \ldots, T_0\} - J$, $w_1(t) = w_2(t) = y_1$.
- For $T - H \leq t \leq T$, $w_1(t) = w_2(t) = z_1(t)$. (Notice that for $T - H \leq t \leq T$, $z_1(t) = z_2(t)$.)

Denote by $I' = \{1, \ldots, T_0\} - J$, $K' = \{T - H, \ldots, T\}$, and for $i = 1, \ldots, H$, We define the following notations for simplicity of calculation, as in Yu et al. [2026].

- $Q_{1,i} = \sum_{t\in J} \lambda_i(w_1(t), t)$.
- $Q_{2,i} = \sum_{t\in J} \lambda_i(w_2(t), t)$.
- $V_{1,i} = (\sum_{t\in J} \lambda_i(w_1(t), t) v_i(w_1(t), t))/Q_{1,i}$.
- $V_{2,i} = (\sum_{t\in J} \lambda_i(w_2(t), t) v_i(w_2(t), t))/Q_{2,i}$.
- $Q_{3,i} = \sum_{t\in I'} \lambda_i(z_1(t), t)$, which is of the same value if defined on $Z_{2,T}$.
- $V_{3,i} = (\sum_{t\in I'} \lambda_i(z_1(t), t) v_i(z_1(t), t))/Q_{3,i}$, which is of the same value if defined on $Z_{2,T}$.
- $Q_{4,i} = \sum_{t\in I'} \lambda_i(w_1(t), t)$, which is of the same value if defined on $W_{2,T}$.
- $V_{4,i} = (\sum_{t\in I'} \lambda_i(w_1(t), t) v_i(w_1(t), t))/Q_{4,i}$, which is of the same value if defined on $W_{2,T}$.
- $Q_{5,i} = \sum_{t\in K'} \lambda_i(w_1(t), t)$, which is of the same value on all Z, W.
- $V_{5,i} = (\sum_{t\in K'} \lambda_i(w_1(t), t) v_i(w_1(t), t))/Q_{5,i}$, which is of the same value on all Z, W.

We then have the following equations:

- $\sum_{t=1}^{T} \lambda_i(z_1(t), t) = Q_{1,i} + Q_{3,i} + Q_{5,i}$.
- $\sum_{t=1}^{T} \lambda_i(z_2(t), t) = Q_{2,i} + Q_{3,i} + Q_{5,i}$.
- $\sum_{t=1}^{T} \lambda_i(w_1(t), t) = Q_{1,i} + Q_{4,i} + Q_{5,i}$.
- $\sum_{t=1}^{T} \lambda_i(z_2(t), t) = Q_{2,i} + Q_{4,i} + Q_{5,i}$.

- $\frac{\sum_{t=1}^{T}\lambda_i(z_1(t),t)v_i(z_1(t),t)}{\sum_{t=1}^{T}\lambda_i(z_1(t),t)} = \frac{Q_{1,i}V_{1,i}+Q_{3,i}V_{3,i}+Q_{5,i}V_{5,i}}{Q_{1,i}+Q_{3,i}+Q_{5,i}}$.
- $\frac{\sum_{t=1}^{T}\lambda_i(z_2(t),t)v_i(z_2(t),t)}{\sum_{t=1}^{T}\lambda_i(z_2(t),t)} = \frac{Q_{2,i}V_{2,i}+Q_{3,i}V_{3,i}+Q_{5,i}V_{5,i}}{Q_{2,i}+Q_{3,i}+Q_{5,i}}$.
- $\frac{\sum_{t=1}^{T}\lambda_i(w_1(t),t)v_i(w_1(t),t)}{\sum_{t=1}^{T}\lambda_i(w_1(t),t)} = \frac{Q_{1,i}V_{1,i}+Q_{4,i}V_{4,i}+Q_{5,i}V_{5,i}}{Q_{1,i}+Q_{4,i}+Q_{5,i}}$.
- $\frac{\sum_{t=1}^{T}\lambda_i(w_2(t),t)v_i(w_2(t),t)}{\sum_{t=1}^{T}\lambda_i(w_2(t),t)} = \frac{Q_{2,i}V_{2,i}+Q_{4,i}V_{4,i}+Q_{5,i}V_{5,i}}{Q_{1,i}+Q_{4,i}+Q_{5,i}}$.

Without loss of generality, we assume $Q_{1,i} \geq Q_{2,i}$. By calculation, we have

$$\frac{Q_{1,i}V_{1,i}+Q_{4,i}V_{4,i}+Q_{5,i}V_{5,i}}{Q_{1,i}+Q_{4,i}+Q_{5,i}} - \frac{Q_{2,i}V_{2,i}+Q_{4,i}V_{4,i}+Q_{5,i}V_{5,i}}{Q_{2,i}+Q_{4,i}+Q_{5,i}} \tag{15}$$

$$= \frac{(Q_{2,i}-Q_{1,i})(Q_{4,i}(V_{4,i}-V_{2,i})+Q_{5,i}(V_{5,i}-V_{2,i}))}{(Q_{1,i}+Q_{4,i}+Q_{5,i})(Q_{2,i}+Q_{4,i}+Q_{5,i})} + \frac{Q_{1,i}}{Q_{1,i}+Q_{4,i}+Q_{5,i}}(V_{1,i}-V_{2,i}). \tag{16}$$

As $Q_{5,i} \geq M_i \geq \frac{Q_{1,i}+Q_{3,i}+Q_{5,i}}{T}$, we have that

$$\| \frac{(Q_{2,i}-Q_{1,i})(Q_{4,i}(V_{4,i}-V_{2,i})+Q_{5,i}(V_{5,i}-V_{2,i}))}{(Q_{1,i}+Q_{4,i}+Q_{5,i})(Q_{2,i}+Q_{4,i}+Q_{5,i})} \| \tag{17}$$

$$\leq \| \frac{(Q_{2,i}-Q_{1,i})(2Q_{4,i}+2Q_{5,i})}{(Q_{1,i}+Q_{4,i}+Q_{5,i})(Q_{2,i}+Q_{4,i}+Q_{5,i})} \| \tag{18}$$

$$\leq \| \frac{2(Q_{2,i}-Q_{1,i})}{(Q_{1,i}+Q_{4,i}+Q_{5,i})} \| \tag{19}$$

$$\leq \| \frac{2T(Q_{2,i}-Q_{1,i})}{(Q_{1,i}+Q_{3,i}+Q_{5,i})} \| \tag{20}$$

$$= \| 1 - \frac{\sum_{t=1}^{T}\lambda_i(z_2(t),t)}{\sum_{t=1}^{T}\lambda_i(z_1(t),t)} \| \tag{21}$$

$$\leq \| \frac{T\varepsilon^{k+1}}{12T^2} \| \tag{22}$$

$$\leq \frac{\varepsilon^{k+1}}{12T} \tag{23}$$

Similarly, we also have

$$\| \frac{(Q_{2,i}-Q_{1,i})(Q_{3,i}(V_{3,i}-V_{2,i})+Q_{5,i}(V_{5,i}-V_{2,i}))}{(Q_{1,i}+Q_{3,i}+Q_{5,i})(Q_{2,i}+Q_{3,i}+Q_{5,i})} \| \leq \frac{\varepsilon^{k+1}}{12T} \tag{24}$$

Then we have

$$\| \frac{Q_{1,i}}{Q_{1,i}+Q_{3,i}+Q_{5,i}}(V_{1,i}-V_{2,i}) \| \tag{25}$$

$$\leq \| \frac{Q_{1,i}V_{1,i}+Q_{3,i}V_{3,i}+Q_{5,i}V_{5,i}}{Q_{1,i}+Q_{3,i}+Q_{5,i}} - \frac{Q_{2,i}V_{2,i}+Q_{3,i}V_{3,i}+Q_{5,i}V_{5,i}}{Q_{2,i}+Q_{3,i}+Q_{5,i}} \| \tag{26}$$

$$+ \| \frac{(Q_{2,i}-Q_{1,i})(Q_{3,i}(V_{3,i}-V_{2,i})+Q_{5,i}(V_{5,i}-V_{2,i}))}{(Q_{1,i}+Q_{3,i}+Q_{5,i})(Q_{2,i}+Q_{3,i}+Q_{5,i})} \| \tag{27}$$

$$= \| \frac{\sum_{t=1}^{T}\lambda_i(z_1(t),t)v_i(z_1(t),t)}{\sum_{t=1}^{T}\lambda_i(z_1(t),t)} - \frac{\sum_{t=1}^{T}\lambda_i(z_2(t),t)v_i(z_2(t),t)}{\sum_{t=1}^{T}\lambda_i(z_2(t),t)} \| \tag{28}$$

$$+ \| \frac{(Q_{2,i}-Q_{1,i})(Q_{3,i}(V_{3,i}-V_{2,i})+Q_{5,i}(V_{5,i}-V_{2,i}))}{(Q_{1,i}+Q_{3,i}+Q_{5,i})(Q_{2,i}+Q_{3,i}+Q_{5,i})} \| \tag{29}$$

$$\leq \frac{\varepsilon^{k+1}}{6T} + \frac{\varepsilon^{k+1}}{12T} \tag{30}$$

$$= \frac{\varepsilon^{k+1}}{4T} \tag{31}$$

Then we have

$$\|\frac{Q_{1,i}}{Q_{1,i}+Q_{4,i}+Q_{5,i}}(V_{1,i}-V_{2,i})\| \tag{32}$$

$$\leq\|\frac{TQ_{1,i}}{Q_{1,i}+Q_{3,i}+Q_{5,i}}(V_{1,i}-V_{2,i})\| \tag{33}$$

$$\leq\frac{\varepsilon^{k+1}}{4} \tag{34}$$

Thus

$$\|\frac{\sum_{t=1}^{T}\lambda_i(w_1(t),t)v_i(w_1(t),t)}{\sum_{t=1}^{T}\lambda_i(w_1(t),t)}-\frac{\sum_{t=1}^{T}\lambda_i(w_2(t),t)v_i(w_2(t),t)}{\sum_{t=1}^{T}\lambda_i(w_2(t),t)}\| \tag{35}$$

$$=\|\frac{Q_{1,i}V_{1,i}+Q_{4,i}V_{4,i}+Q_{5,i}V_{5,i}}{Q_{1,i}+Q_{4,i}+Q_{5,i}}-\frac{Q_{2,i}V_{2,i}+Q_{4,i}V_{4,i}+Q_{5,i}V_{5,i}}{Q_{2,i}+Q_{4,i}+Q_{5,i}}\| \tag{36}$$

$$\leq\|\frac{Q_{1,i}}{Q_{1,i}+Q_{4,i}+Q_{5,i}}(V_{1,i}-V_{2,i})\|+\|\frac{(Q_{2,i}-Q_{1,i})(Q_{3,i}(V_{3,i}-V_{2,i})+Q_{5,i}(V_{5,i}-V_{2,i}))}{(Q_{1,i}+Q_{3,i}+Q_{5,i})(Q_{2,i}+Q_{3,i}+Q_{5,i})}\| \tag{37}$$

$$\leq\frac{\varepsilon^{k+1}}{4}+\frac{\varepsilon^{k+1}}{12T} \tag{38}$$

$$\leq\frac{\varepsilon^{k+1}}{2} \tag{39}$$

This indicated that $W_{1,T}, W_{2,T}$ have very close representations.

Now we show that $F(W_{1,T})$ and $F(W_{2,T})$ are at least separated by ε.

It is clear that $F(X_T) = \min_{1\leq s,t\leq T}\|x(s)+x(t)\|_2^2$ if the entire X_T lies on $\mathbb{S}^2$, which both $W_{1,T}, W_{2,T}$ satisfies. As $w_1(T) = w_2(T) = b_1$ is the only point on each sequence in Q_0, and the rest of the sequence are in $-Q_0$, it is clear that $F(W_{i,t}) = \min_{1\leq s\leq T-1}\|w_i(t)-(-b_1)\|_2^2$. Furthermore, by the fact that $Y\cap B(-b_1,L)=\varnothing$, this further reduces to $F(W_{i,t}) = \min_{s\in J}\|w_i(t)-(-b_1)\|_2^2$.

By the construction of U_t and $Z_{1,T}, Z_{2,T}$, let $j^* = \min_{t\in J}$, we have that $F(W_{i,t}) = \|w_i(j^*)-(-b_1)\|_2^2 = \|z_i(j^*)-(-b_1)\|_2^2$. From $z_1(j^*)\neq z_2(j^*)$, we have that $|\|z_1(j^*)-(-b_1)\|_2^2-\|z_2(j^*)-(-b_1)\|_2^2|\geq 4\varepsilon$. From $\delta(M_1)\leq\varepsilon$, we have that $|M_1(W_{1,T})-M_1(W_{2,T})|\geq|F(W_{1,T})-F(W_{2,T})|-|F(W_{1,T})-M_1(W_{1,T})|-|F(W_{2,T})-M_1(W_{2,T})|\geq 4\varepsilon-\varepsilon-\varepsilon=2\varepsilon$.

Then by Lemma 4 in Yu et al. [2026], which states that for two vectors $v_1, v_2\in\mathbb{R}^n$. If

$$\|v_1-v_2\|_2\leq A \quad\text{and}\quad \|\hat{F}(v_1)-\hat{F}(v_2)\|\geq B,$$

where $\hat{F}:\mathbb{R}^n\to\mathbb{R}^m$ is a two-layer feed-forward network satisfying with Tanh-activation and bounded weights, then $\hat{F}$ must use at least

$$\Omega\left(\frac{B}{A\sqrt{n}}\right)$$

parameters. Having $A\leq\frac{\varepsilon^{k+1}}{2}$, and $B\geq\varepsilon$

Thus the parameter count is lower bounded by $\Omega(\frac{1}{\varepsilon^k})$.

Second Part of Theorem 1 We directly construct this two-layer transformer.

1. $x_1(t) = \hat{x}(t) = P_\phi(x(t),t) = (\frac{x(t)}{3},0)\in\mathbb{R}^6$. This embedding requires only $O(1)$ parameters.
2. $W_{Q,1,1}^T W_{K,1,1} = \begin{pmatrix}-I_3 & 0\\ 0 & 0\end{pmatrix}\in\mathbb{R}^{6\times 6}$. We then have that $(W_{Q,1,1}x_1(s))^T W_{K,1,1}x_1(t) = -\frac{1}{9}x(s)^T x(t)$.

3. $W_{V,1,1} = I_6$, $W_{O,1} = \begin{pmatrix} 0 & 0 \\ I_3 & 0 \end{pmatrix}$. We then have $x_1'(t) = (\frac{1}{3}x(t), v(t)) \in \mathbb{R}^6$, where $v(t)$ is the actual vector obtained from softmax attention. By the fact that log attention score $(W_{Q,1,1}x_1(s))^T W_{K,1,1}x_1(t) = -\frac{1}{9}x(s)^T x(t)$ and we can apply a scaling factor $\beta > 0$ to this term in soft attention, we have that when $\beta \to \infty$, $x(t)^T v(t) \to \frac{1}{9}\min_{1\le s\le T} x(s)^T x(t)$ uniformly for all $X_T \in \mathcal{X}^T$. We set β to be large enough such that $|x(t)^T v(t) - \frac{1}{9}\min_{1\le s\le T} x(s)^T x(t)| \le \frac{\varepsilon}{54}$.

4. Denote by $G : [-1,1]^6 \to [-1,1]^6$ the following function: $G((x,y)) = (x^T y, \frac{1}{2}, 0, 0, 0, 0)$, where $(x,y) \in \mathbb{R}^6, x, y \in \mathbb{R}^3$. Then there exists a two-layer Tanh-activated FFN F_1 with at most $O(1/\varepsilon^1)$ parameters such that $|F_1((x,y)) - G((x,y))| \le \frac{\varepsilon}{54}$. (As we can rewrite $x_i y_i = \frac{1}{4}[(x_i + y_i)^2 - (x_i - y_i)^2]$, this approximation of inner product is transformed to one dimensional FFN approximation problem, therefore $O(1/\varepsilon^1)$ approximation rate can be achieved by FFNs.)

5. Write $x_2(t) = (a(t), \frac{1}{2}, 0, 0, 0, 0) \in \mathbb{R}^6$. Let $W_{Q,1,2}^T W_{K,1,2} = A \in \mathbb{R}^{6\times 6}$, where $A_{2,1} = -2$ is the only non-zero element of A. Thus we have that $(W_{Q,1,2}c_2)^T W_{K,1,1}x_2(t) = -a(t)$.

6. $W_{V,1,2} = I_6$, $W_{O,2} = \begin{pmatrix} 0 & 0 \\ I_3 & 0 \end{pmatrix}$. With the scaling factor $\beta > 0$, we also have a large enough β such that the 4-th element of c_2' satisfies that $|(c_2')_4 - \min_{1\le t\le T} a(t)| \le \frac{\varepsilon}{54}$. Then the output feed-forward block maps c_2' to $2 + 18(c_2')_4$ with $O(1)$ parameters.

7. The total error is

$$
\begin{aligned}
&|M_2(X_T) - F(X_T)| && (40)\\
\le& 18[\max_{1\le t\le T}(|x(t)^T v(t) - \frac{1}{9}\min_{1\le s\le T} x(s)^T x(t)| + |a(t) - \frac{1}{9}\min_{1\le s\le T} x(s)^T x(t)|) && (41)\\
&+|(c_2')_4 - \min_{1\le t\le T} a(t)|] && (42)\\
\le& \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + \frac{\varepsilon}{3} && (43)\\
\le& \varepsilon && (44)
\end{aligned}
$$

By this construction we have proven the second part of Theorem 1.

□

Remark. If we choose $\mathcal{X} = [0,1]^3$, we can also choose a similar target for Theorem 1 to hold.

A.2 Formal statement and proof of Theorem 2

It suffices to show this statement in the scalar case with $\mathcal{X} = [0,1]$ ($X_T \in [0,1]^T$). For a given X_T, there exists a permutation σ such $x(\sigma(1)) \ge x(\sigma(2)) \ge \ldots, \ge x(\sigma(T))$. Then for a fixed k, define $F_k : [0,1]^T \to [0,1]$ to be $F_k(X_T) = x(\sigma(k))$.

We have the following theorem showing that softmax attention cannot retrieval the k-th largest value efficiently.

Theorem 5 (Formal Version of Theorem 2)**.** *Let $f_1 : [0,1] \to [0,1]^n$ and $f_2 : [0,1]^n \to [0,1]$ be two-layer Tanh-activated Feed forward Neural Networks, with weights bounded by 1. For any non-decreasing $\rho : [0,1] \to \mathbb{R}$, we have a softmax attention $M : X_T \to [0,1]$ of the following form:*

$$
M(X_T) = f_2\left(\frac{\sum_{t=1}^T [\exp(\rho(x(t)))f_1(x(t))]}{\sum_{t=1}^T \exp(\rho(x(t)))}\right) \tag{45}
$$

If $\max_{X_T \in \mathcal{X}^T} |M(X_T) - F_k(X_T)| \le \varepsilon$ and $k \ge 2$, then f_1 and f_2 have parameter count at least $\Omega(\varepsilon^{-o})$, where $o = \frac{T-k}{n+1} - 1$.

Proof Sketch: To prove Theorem 2, we follow a similar idea to the first part of the proof of Theorem 1.

Proof. Let

$$m := T - k + 1.$$

The proof reduces the k-th largest-value task to a maximum-retrieval task over m free coordinates. The remaining $k-1$ coordinates will be fixed at value 1. Since $k \geq 2$, at least one such fixed top coordinate is available.

Define

$$w(x) := \exp(\rho(x)).$$

Since ρ is non-decreasing, for every $x \in [0,1]$,

$$0 < \frac{w(x)}{w(1)} \leq 1.$$

Set

$$\lambda(x) := \frac{w(x)}{w(1)}, \qquad a(x) := \lambda(x) f_1(x) \in [0,1]^n, \qquad b(x) := \lambda(x) \in (0,1].$$

Multiplying all attention weights by the same positive constant $w(1)^{-1}$ does not change the softmax average. Hence the attention representation can be written as

$$A(X_T) := \frac{\sum_{t=1}^{T} a(x(t))}{\sum_{t=1}^{T} b(x(t))} \in [0,1]^n.$$

We now construct many candidate subsequences. Choose $\varepsilon > 0$ sufficiently small, for instance

$$0 < \varepsilon < \frac{1}{64m}.$$

Let

$$\Delta := 4\varepsilon, \qquad N := \left\lfloor \frac{1}{16m\varepsilon} \right\rfloor.$$

For each $j = 1, \ldots, m$, define

$$\alpha_j := \frac{j-1}{2m},$$

and the grid

$$G_j := \{\alpha_j + q\Delta : \ q = 1, \ldots, N\}.$$

For sufficiently small ε, each $G_j \subset [0, 1/2]$. Moreover, the grids are ordered: if $i < j$, then every point of G_i is smaller than every point of G_j. Also, two distinct points in the same G_j differ by at least

$$\Delta = 4\varepsilon.$$

Consider all grid subsequences

$$z = (z_1, \ldots, z_m) \in G_1 \times \cdots \times G_m.$$

There are N^m such subsequences. For each z, define

$$S(z) := \left(\sum_{j=1}^{m} a(z_j), \sum_{j=1}^{m} b(z_j) \right) \in \mathbb{R}^{n+1}.$$

Because $a(z_j) \in [0,1]^n$ and $b(z_j) \in (0,1]$, we have

$$S(z) \in [0, m]^{n+1}.$$

Let

$$\eta := 4m\, N^{-m/(n+1)}.$$

Partition the cube $[0, m]^{n+1}$ into axis-aligned cubes of side length η. The number of cubes is at most

$$\left(\frac{m}{\eta} + 1\right)^{n+1} = \left(\frac{1}{4}N^{m/(n+1)} + 1\right)^{n+1}.$$

For N large enough, this number is strictly smaller than N^m. Hence, by the pigeonhole principle, there exist two distinct subsequences

$$z = (z_1, \ldots, z_m), \qquad z' = (z'_1, \ldots, z'_m)$$

such that

$$\left\| \sum_{j=1}^{m} a(z_j) - \sum_{j=1}^{m} a(z'_j) \right\|_\infty \leq \eta,$$

and

$$\left| \sum_{j=1}^{m} b(z_j) - \sum_{j=1}^{m} b(z'_j) \right| \leq \eta.$$

Let

$$J := \{j \in [m] : z_j \neq z'_j\}.$$

Since $z \neq z'$, the set J is nonempty. We now build two full input sequences $X, Y \in [0, 1]^T$. For the first $k - 1$ positions, set

$$x(1) = \cdots = x(k-1) = 1, \qquad y(1) = \cdots = y(k-1) = 1.$$

For the remaining $m = T - k + 1$ positions, indexed by $j = 1, \ldots, m$, set

$$x(k - 1 + j) := \begin{cases} z_j, & j \in J, \\ 0, & j \notin J, \end{cases}$$

and

$$y(k - 1 + j) := \begin{cases} z'_j, & j \in J, \\ 0, & j \notin J. \end{cases}$$

We first show that the targets are separated. Since the first $k - 1$ entries of both X and Y equal 1, and all remaining entries lie in $[0, 1/2]$, the k-th largest entry is the maximum among the remaining m entries. Let

$$j_* := \max J.$$

Because the grids $G_1, \ldots, G_m$ are strictly ordered, the largest nonzero remaining entry of X is z_{j_*}, and the largest nonzero remaining entry of Y is z'_{j_*}. Therefore

$$F_k(X) = z_{j_*}, \qquad F_k(Y) = z'_{j_*}.$$

Since $z_{j_*}, z'_{j_*} \in G_{j_*}$ are distinct grid points,

$$|F_k(X) - F_k(Y)| = |z_{j_*} - z'_{j_*}| \geq \Delta = 4\varepsilon.$$

Next we show that the attention representations of X and Y are very close. Define

$$A_X := \sum_{t=1}^{T} a(x(t)), \qquad B_X := \sum_{t=1}^{T} b(x(t)),$$

and similarly A_Y, B_Y. Since $b(1) = 1$ and $k \geq 2$,

$$B_X \geq k - 1, \qquad B_Y \geq k - 1.$$

Moreover, the coordinates where $j \notin J$ are identical in X and Y, and hence cancel in the difference. Thus

$$A_X - A_Y = \sum_{j \in J} \big(a(z_j) - a(z'_j)\big) = \sum_{j=1}^{m} \big(a(z_j) - a(z'_j)\big),$$

so

$$\|A_X - A_Y\|_\infty \le \eta.$$

Similarly,

$$|B_X - B_Y| \le \eta.$$

The attention representations are

$$A(X) = \frac{A_X}{B_X}, \qquad A(Y) = \frac{A_Y}{B_Y}.$$

For each coordinate $\ell = 1, \dots, n$,

$$\left| \frac{(A_X)_\ell}{B_X} - \frac{(A_Y)_\ell}{B_Y} \right| \le \frac{|(A_X - A_Y)_\ell|}{B_X} + (A_Y)_\ell \left| \frac{1}{B_X} - \frac{1}{B_Y} \right|.$$

Since $0 \le f_1(x)_\ell \le 1$, we have $(A_Y)_\ell \le B_Y$. Therefore

$$\begin{aligned} \left| \frac{(A_X)_\ell}{B_X} - \frac{(A_Y)_\ell}{B_Y} \right| &\le \frac{\eta}{k-1} + B_Y \frac{|B_X - B_Y|}{B_X B_Y} \\ &\le \frac{\eta}{k-1} + \frac{\eta}{k-1} = \frac{2\eta}{k-1}. \end{aligned}$$

Hence

$$\|A(X) - A(Y)\|_\infty \le \frac{2\eta}{k-1},$$

and therefore

$$\|A(X) - A(Y)\|_2 \le \frac{2\sqrt{n}\,\eta}{k-1}.$$

Since M is assumed to ε-approximate F_k, we have

$$|M(X) - F_k(X)| \le \varepsilon, \qquad |M(Y) - F_k(Y)| \le \varepsilon.$$

Thus

$$\begin{aligned} |M(X) - M(Y)| &\ge |F_k(X) - F_k(Y)| - |M(X) - F_k(X)| - |M(Y) - F_k(Y)| \\ &\ge 4\varepsilon - 2\varepsilon = 2\varepsilon. \end{aligned}$$

But

$$M(X) = f_2(A(X)), \qquad M(Y) = f_2(A(Y)).$$

Therefore

$$|f_2(A(X)) - f_2(A(Y))| \ge 2\varepsilon.$$

It remains to convert this separation into a parameter lower bound for f_2. Write the two-layer tanh network f_2 as

$$f_2(u) = \sum_{r=1}^{p} \alpha_r \tanh(\theta_r^\top u + \beta_r) + \gamma,$$

where p is the hidden width. Since all weights are bounded by 1,

$$|\alpha_r| \le 1, \qquad \|\theta_r\|_2 \le \sqrt{n}.$$

Because $\tanh$ is 1-Lipschitz,

$$\begin{aligned} |f_2(u) - f_2(v)| &\le \sum_{r=1}^{p} |\alpha_r| \left| \tanh(\theta_r^\top u + \beta_r) - \tanh(\theta_r^\top v + \beta_r) \right| \\ &\le \sum_{r=1}^{p} |\theta_r^\top (u - v)| \\ &\le \sum_{r=1}^{p} \|\theta_r\|_2 \|u - v\|_2 \\ &\le p\sqrt{n}\, \|u - v\|_2. \end{aligned}$$

Applying this with $u = A(X)$ and $v = A(Y)$, we obtain

$$2\varepsilon \le p\sqrt{n}\,\|A(X) - A(Y)\|_2 \le p\sqrt{n}\,\frac{2\sqrt{n}\,\eta}{k-1}.$$

Hence

$$p \ge \frac{(k-1)\varepsilon}{n\eta}.$$

Since the parameter count P_2 of f_2 is at least its hidden width p,

$$P_2 \ge \frac{(k-1)\varepsilon}{n\eta}.$$

Finally, recall that

$$\eta = 4m\,N^{-m/(n+1)}.$$

Since

$$N = \left\lfloor \frac{1}{16m\varepsilon} \right\rfloor = \Theta(\varepsilon^{-1}),$$

we have

$$\eta = O\Big(\varepsilon^{m/(n+1)}\Big).$$

Therefore

$$P_2 \ge C_{T,k,n}\,\varepsilon^{1-m/(n+1)} = C_{T,k,n}\,\varepsilon^{-\left(\frac{m}{n+1}-1\right)}.$$

Since $m = T - k + 1$, this gives

$$P_2 = \Omega\Big(\varepsilon^{-\left(\frac{T-k+1}{n+1}-1\right)}\Big).$$

The total parameter count of (f_1, f_2) is at least P_2, so the same lower bound holds for the combined parameter count.

This proves the theorem.

□

Remark. We choose ρ to be non-decreasing in the spirit that $x(\sigma(k))$ doesn't happen to be the maximizer of another function.

A.3 Justification of the Parameter Cost in Section 5.2

We propose the formal version of Theorem 3 to justify the choice of parameter cost in Section 5.2.

Theorem 6 (Formal Version of Theorem 3)**.** *For sequence X_T, let T_0 be a integer such that $1 \le T_0 \le T$. Then for two layer Tanh-activated neural networks $f_1 : \mathcal{X}^{T_0} \to [0,1]^n$ and $f_2 : [0,1]^n \to \mathcal{X}^{T_0}$, let $I \subseteq [T]$ be an index set with $|I| = T_0$. We have the following:*

- *For a fixed $p \ge 1$, there exists f_1, f_2 with total parameter count of order $O(\varepsilon^{-\frac{T_0 d}{n}})$ such that $\|X(I) - f_2 \circ f_1(X(I))\|_{L_p} \le \varepsilon$.*
- *If $\|X(I) - f_2 \circ f_1(X(I))\|_{L_\infty} \le \varepsilon$ and the norm of the weights of f_1 and f_2 are bounded by B, then f_1, f_2 must have total parameter count of order $O(\varepsilon^{\min(1-\frac{T_0 d}{n},0)})$.*

Explanation: In practical Transformers, information associated with retrieved positions is not represented explicitly, but is instead embedded in the token states $x_l(t) \in \mathbb{R}^{E_l}$. Recovering this information, namely $X(I(t,l))$, generally requires nontrivial decoding by the feed-forward layers. Theorem 3 have demonstrated that to achieve ε-accurate encoding and decoding, a parameter cost exponential to $|I(t,l)|/E_l$ is required. It is also shown that $O(1/\varepsilon^{|I(t,l)|d/E_l})$ parameters are enough for such encoding and decoding. Therefore $O(1/\varepsilon^{|I(t,l)|d/E_l})$ is a reasonable parameter cost associated to the processing of information set $I(t,l)$.

We then consider proving Theorem 3.

Proof Sketch For the first part of Theorem 3, we first construct two Heaviside function activated FFNs $\hat{f}_1, \hat{f}_2$ to store the information of $X(I)$ using binary representation up to $\varepsilon/1$-accuracy. Then we replace each Heaviside unit with Tanh unit to derive the ε approximation in L_p norm. For the second part, we use Pigeonhole Principle to complete the argument.

Proof. **First Part of Theorem 3** We first consider the case $T_0 d > n$. When we say Heaviside function in this proof, we refer to $H(x) = \mathbf{1}_{\{\mathbf{x} \geq \mathbf{0}\}}$.

Construction of Encoding function $\hat{f}_1$. Let $m = T_0 d$, and identify $X(I) \in \mathcal{X}^{T_0}$ with a vector

$$V = (V_1, \ldots, V_m) \in [0,1]^m.$$

Choose

$$L = \left\lceil \log_2 \frac{2m}{\varepsilon} \right\rceil.$$

For each coordinate V_j, let

$$V_j^{(L)} = \sum_{\ell=1}^{L} 2^{-\ell} b_{j,\ell}(V_j), \qquad b_{j,\ell}(V_j) \in \{0,1\},$$

be its L-bit binary truncation. Then

$$\|V - V^{(L)}\|_\infty \leq 2^{-L} \leq \frac{\varepsilon}{2m}, \qquad \|V - V^{(L)}\|_p \leq m^{1/p} \frac{\varepsilon}{2m} \leq \frac{\varepsilon}{2}.$$

For each fixed coordinate V_j, the map

$$V_j \longmapsto (b_{j,1}, \ldots, b_{j,L})$$

can be implemented by a two-layer Heaviside network with $O(2^L)$ parameters. Indeed, the L-bit truncation partitions $[0,1]$ into 2^L dyadic intervals, and the binary vector $(b_{j,1}, \ldots, b_{j,L})$ is constant on each such interval. A two-layer Heaviside network can represent this piecewise-constant map by using threshold units to identify the dyadic intervals and then assigning to each interval its corresponding binary code. Therefore, encoding all $m = T_0 d$ coordinates requires

$$O(m 2^L)$$

parameters.

Since

$$L = \left\lceil \log_2 \frac{2m}{\varepsilon} \right\rceil,$$

we have $2^L = O(m/\varepsilon)$. Hence, for fixed T_0 and d, the encoder can be implemented by a two-layer Heaviside network $\widehat{f}_1 : [0,1]^m \to [0,1]^n$ with

$$O(m 2^L) = O\left(\frac{1}{\varepsilon}\right)$$

parameters. The resulting mL binary digits are then distributed among the n latent coordinates, with each coordinate storing at most

$$\left\lceil \frac{mL}{n} \right\rceil$$

digits through its own binary expansion.

Construction of Decoding Function $\hat{f}_2$. Next we construct the decoder $\widehat{f}_2$.
Recall that the encoder $\widehat{f}_1$ allocates the mL binary digits among the n latent coordinates, with each latent coordinate storing at most

$$q := \left\lceil \frac{mL}{n} \right\rceil$$

digits through a binary expansion. The decoder $\widehat{f}_2$ is constructed according to this fixed allocation rule. For each latent coordinate, recovering its assigned binary digits amounts to decoding a q-bit

binary expansion. This is again a piecewise-constant map on $[0,1]$ with at most 2^q dyadic intervals, and hence can be implemented by a two-layer Heaviside network with $O(2^q)$ parameters. Applying this to all n latent coordinates and then rearranging the recovered bits according to the allocation rule of $\widehat{f}_1$, the decoder recovers all digits

$$\{b_{j,\ell} : 1 \le j \le m,\ 1 \le \ell \le L\}.$$

It then outputs

$$V_j^{(L)} = \sum_{\ell=1}^{L} 2^{-\ell} b_{j,\ell}, \qquad j = 1, \dots, m.$$

Therefore,

$$\widehat{f}_2 \circ \widehat{f}_1(V) = V^{(L)}.$$

By the choice of

$$L = \left\lceil \log_2 \frac{2m}{\varepsilon} \right\rceil,$$

we have

$$\|V - V^{(L)}\|_\infty \le 2^{-L} \le \frac{\varepsilon}{2}.$$

Moreover,

$$2^q = 2^{\lceil \frac{mL}{n} \rceil} \le 2\left(2^L\right)^{m/n}.$$

Since $2^L = O(1/\varepsilon)$, the decoder $\widehat{f}_2$ has parameter count

$$O\Big(2^{\lceil \frac{mL}{n} \rceil}\Big) = O\Big(\varepsilon^{-m/n}\Big).$$

We then consider replacing these Heaviside activation functions with Tanh activation functions. We first prove the following lemma regarding such replacements.

Lemma 1 (Replacing Heaviside Unit with Tanh Unit)**.** *Let $p > 1$, let $a < 0 < b$, and let μ be a finite positive measure on $[a,b]$. Let $H(x) = \mathbf{1}_{\{x \ge 0\}}$. Then for every $\delta > 0$, there exist $A, B, C, D \in \mathbb{R}$ such that*

$$\big\|H - \big(A \tanh(Bx + C) + D\big)\big\|_{L_p(\mu)} < \delta.$$

Proof. It suffices to consider functions of the form

$$\phi_{B,C}(x) := \frac{1 + \tanh(Bx + C)}{2}.$$

Let $M := \mu([a,b]) < \infty$. If $M = 0$, the claim is trivial, so assume $M > 0$.

Since μ is finite,

$$\mu\big((-\eta, \eta) \setminus \{0\}\big) \to 0 \qquad \text{as } \eta \downarrow 0.$$

Choose $\eta > 0$ such that

$$\mu\big((-\eta, \eta) \setminus \{0\}\big) < \frac{\delta^p}{3}.$$

Next choose $C > 0$ such that

$$\left|1 - \frac{1 + \tanh C}{2}\right|^p \mu(\{0\}) < \frac{\delta^p}{3}.$$

Finally, choose $R > 0$ sufficiently large so that

$$\left(\frac{1 + \tanh(-R)}{2}\right)^p M + \left(1 - \frac{1 + \tanh R}{2}\right)^p M < \frac{\delta^p}{3},$$

and then choose $B > 0$ large enough that

$$-B\eta + C \le -R, \qquad B\eta + C \ge R.$$

For $x \le -\eta$, we have $Bx + C \le -R$, and hence $\phi_{B,C}(x)$ is uniformly small. For $x \ge \eta$, we have $Bx + C \ge R$, and hence $\phi_{B,C}(x)$ is uniformly close to 1. On $(-\eta, \eta) \setminus \{0\}$, the pointwise error is bounded by 1, while at $x = 0$ it is controlled by the choice of C. Therefore,

$$\begin{aligned}\|H - \phi_{B,C}\|_{L_p(\mu)}^p &\le \int_{[a,-\eta]} |\phi_{B,C}(x)|^p \, d\mu(x) + \int_{[\eta,b]} |1 - \phi_{B,C}(x)|^p \, d\mu(x) \\ &\quad + \mu\big((-\eta, \eta) \setminus \{0\}\big) + |1 - \phi_{B,C}(0)|^p \mu(\{0\}) \\ &< \delta^p.\end{aligned}$$

Thus $\|H - \phi_{B,C}\|_{L_p(\mu)} < \delta$. Taking

$$A = \frac{1}{2}, \qquad D = \frac{1}{2}$$

gives the desired form. □

We now pass from Heaviside networks to Tanh networks. Let

$$\widehat{F} := \widehat{f}_2 \circ \widehat{f}_1$$

be the Heaviside network constructed above, and let N_0 be the total number of Heaviside units in $\widehat{F}$. We replace these Heaviside units one at a time.

At any intermediate stage, if not all Heaviside units have been replaced, choose a Heaviside unit whose output does not feed into any other remaining Heaviside unit. Such a unit always exists by the acyclicity of the feed-forward computation graph. Writing the output of this unit as

$$H(z),$$

where z is an affine function of the previous layer outputs, the part of the network after this unit is a Tanh/affine network, and is therefore continuous, indeed Lipschitz, with respect to this scalar input on the relevant compact range. Denote this downstream map by G, and let L_G be a Lipschitz constant.

Let ν be the pushforward of the input distribution under z. By the preceding lemma, for any $\delta > 0$, there exist $A, B, C, D \in \mathbb{R}$ such that

$$\|H - (A \tanh(B \cdot + C) + D)\|_{L_p(\nu)} < \delta.$$

Choose δ sufficiently small so that

$$L_G \delta \le \frac{\varepsilon}{2N_0}.$$

Replacing this Heaviside unit by

$$A \tanh(Bz + C) + D$$

therefore changes the final network output by at most

$$\frac{\varepsilon}{2N_0}$$

in L_p.

Repeating this procedure for all N_0 Heaviside units yields Tanh-activated networks f_1, f_2 satisfying

$$\|f_2 \circ f_1 - \widehat{f}_2 \circ \widehat{f}_1\|_{L_p} \le \sum_{r=1}^{N_0} \frac{\varepsilon}{2N_0} = \frac{\varepsilon}{2}.$$

Since the Heaviside construction gives

$$\|V - \widehat{f}_2 \circ \widehat{f}_1(V)\|_{L_p} \le \frac{\varepsilon}{2},$$

the triangle inequality implies

$$\|V - f_2 \circ f_1(V)\|_{L_p} \le \|V - \widehat{f}_2 \circ \widehat{f}_1(V)\|_{L_p} + \|\widehat{f}_2 \circ \widehat{f}_1(V) - f_2 \circ f_1(V)\|_{L_p} \le \varepsilon.$$

Thus the desired L_p-approximation is obtained by Tanh-activated networks.

When $n \geq T_0 d$, we just let $f_1(V) = (V, 0) \in \mathbb{R}^n$, and $f_2((U, 0)) = U$, then we can achieve exact representation with $O(1)$ parameters.

Second Part of Theorem 3

Let $m = T_0 d$. The case $m \leq n$ only gives the trivial lower bound $\Omega(1)$, so assume $m > n$. By the standard packing argument for $f_1 : [0,1]^m \to [0,1]^n$, there exist $V_1, V_2 \in [0,1]^m$ such that

$$\|V_1 - V_2\|_\infty \geq 4\varepsilon, \qquad \|f_1(V_1) - f_1(V_2)\|_2 \leq C\varepsilon^{m/n}.$$

Since $\|V - f_2 \circ f_1(V)\|_{L_\infty} \leq \varepsilon$, we have

$$\begin{aligned}\|f_2(f_1(V_1)) - f_2(f_1(V_2))\|_2 \geq \|V_1 - V_2\|_\infty - \|V_1 - f_2(f_1(V_1))\|_\infty \\ - \|V_2 - f_2(f_1(V_2))\|_\infty \geq 2\varepsilon.\end{aligned}$$

Applying the Lemma 4 from Yu et al. [2026] to f_2, with

$$A = C\varepsilon^{m/n}, \qquad B = 2\varepsilon,$$

shows that f_2 must have at least

$$\Omega\Big(\frac{B}{A\sqrt{n}}\Big) = \Omega\Big(\varepsilon^{1-m/n}\Big) = \Omega\Big(\frac{1}{\varepsilon^{m/n-1}}\Big)$$

parameters. Therefore the total parameter count of f_1, f_2 is at least

$$\Omega\Big(\varepsilon^{1-\frac{T_0 d}{n}}\Big),$$

which is the desired lower bound in the nontrivial regime $T_0 d > n$. □

Remark. We can replace Tanh with any 1-Lipschitz sigmoidal activation function to obtain the same results following similar proof strategy.

B Derivation of Predictions

B.1 Derivation of Intrinsic Dimension Phenomena in Two-Layer Transformers

Derivation of Proposition 1 To prove the upper bound, we construct the following D Trees of Comparison $TR_1, \ldots, TR_D$.

1. For TR_i, each leaf is associated with an ordered index set t_1, t_2 for $1 \leq t_1, t_2 \leq T$, thus TR_i has T^2 leaves. And for each internal node of TR_i, its corresponding f_{in} is defined as $f_{in}(X[\{s_1, s_2\}]) = x(s_1)^T A_i x(s_2)$. Thus $I_{fi}(X_T, TR_i) = \arg\max_{\{s_1,s_2\}} x(s_1)^T A_i x(s_2)$. Then we have $I_F(X_T) \subseteq \bigcup_{i=1}^D I_{fi}(X_T, TR_i)$.

With this upper bound we also have $\beta' \leq \beta_1 \leq 2$. Then following a similar argument in D.1, we can show that each index set $\{t_1, t_2\}$ for $t_1 \neq t_2$ must be on the leaf of a Tree of Comparison. This gives a lower bound $\Omega(T^2)$ for the Number of Comparison, and hence $\beta_1 = \beta' = 2$.

As the target F have $\beta_1 = \beta' = 2$. Using only max position aggregation, we have that $I(t,1) \subseteq \bigcup_{i=1}^{h_1} I(s_i(t), 0) \cup I(t, 0)$. As $|I(s,0)| = |I(t,0)| = 1$, we have that $|I(t,1)| \leq (h_1 + 1)$. Similarily we have $|I(T+1, 1)| \leq h_1$. Then $|I(T+1, 2)| \leq \sum_{i=1}^{h_2} |I(s_i(T+1), 1)| + |I(T+1, 1)| \leq h_1 h_2 + h_2 + h_1 < (h_1 + 1)(h_2 + 1)$. Substituting all terms into the formula for Number of Comparison, in the regime of $h_1, h_2 \ll T$, we have

$$\sum_{t=1}^{T}\sum_{l=1}^{L-1}\Big(|I(t,l)|^{\beta'} - 1 + \sum_{i=1}^{h_l}(T-1)\Big) + \sum_{l=1}^{L}\Big(|I(T+1,l)|^{\beta'} - 1 + \sum_{i=1}^{h_l}(T-1)\Big) \tag{46}$$

$$\leq \sum_{t=1}^{T}\Big(|I(t,1)|^2 - 1 + \sum_{i=1}^{h_1}(T-1)\Big) + \sum_{l=1}^{2}\Big(|I(T+1,l)|^2 - 1 + \sum_{i=1}^{h_l}(T-1)\Big) \tag{47}$$

$$\leq T(h_1+1)^2 - T + h_1 T(T-1) + h_1^2 + (h_1+1)^2(h_2+1)^2 - 2 + 2(T-1) \tag{48}$$

$$= h_1 T^2 + O(T) \tag{49}$$

Therefore we need $h_1 \geq D$ to have large enough Number of Comparison.

To show that the (D, D) heads model exists, we construct an explicit **InfoFlow** model with $h_1 = h_2 = D$ heads that learns the target. We set

$$I(t,1) = \{s_i : s_i = \arg\max_{1 \leq s \leq T} x(s)^\top A_i x(t)\} \cup \{t\} \tag{50}$$

and

$$I(T+1,2) = \{(t, s_i(t)) : t \in \bigcup_{i=1}^{D} \{\arg\max_{1 \leq t \leq T} x(s_i(t))^\top A_i x(t)\}\}. \tag{51}$$

We have $I_F(X_T) \subseteq I(T+1,2)$, confirming that the two-layer Transformer with (D, D) heads approximates F effectively.

B.2 Derivation of Approximation Hardness of Higher-Order Retrieval Tasks

Derivation of Proposition 2 Suppose the target F have Number of Comparison lower bounded by $C_1 T^{\beta'}$ for a positive constant C_1 (This is valid by the definition of Order of Comparison β'). Let $M = \max_{t,l} |I(t,l)|$, we have an estimation for the Number of Comparison of the model (in the regime of $H, L, E \ll T$):

$$\sum_{t=1}^{T} \sum_{l=1}^{L-1} \Big(|I(t,l)|^{\beta_1} - 1 + \sum_{i=1}^{h_l} (T-1) \Big) + \sum_{l=1}^{L} \Big(|I(T+1,l)|^{\beta_1} - 1 + \sum_{i=1}^{h_l} (T-1) \Big) \tag{52}$$

$$\leq \sum_{t=1}^{T} \sum_{l=1}^{L-1} \Big(M^{\beta_1} - 1 + H(T-1) \Big) + \sum_{l=1}^{L} \Big(M^{\beta_1} - 1 + H(T-1) \Big) \tag{53}$$

$$\leq T(L-1)M^{\beta_1} + T(L-1)H(T-1) + LM^{\beta_1} + LH(T-1) \tag{54}$$

$$\leq TLM^{\beta_1} + HLT^2 + LM^{\beta_1} + LHT \tag{55}$$

$$= TLM^{\beta_1} + HLT^2 + O(M^{\beta_1} + T) \tag{56}$$

If $TLM^{\beta_1} + HLT^{\beta_1} + O(M^{\beta_1} + T) \geq C_1 T^{\beta'}$ holds for $T \gg HL$, we need $TLM^{\beta_1} \geq C_2 T^{\beta'}$ for positive C_2, thus we have $M \geq C T^{\frac{\beta'-1}{\beta_1}} / L$ for positive constant C. This leads to a parameter cost of $O(1/\varepsilon^{M/E}) = O(1/\varepsilon^{CT^{\frac{\beta'-1}{\beta_1}}/LE})$.

C Experimental Details and Results

C.1 Max-Position Retrieval in Trained Two-Layer Transformers

Setup. We train two-layer Transformers on the generalized D-retrieval target with intrinsic dimension $D = 6$, sequence length $T = 64$, and feed-forward hidden width $w_1 = 384$. We consider six configurations of head counts (h_1, h_2) and corresponding embedding dimensions $E_1 = h_1 n_1$, with per-head embedding dimension n_1 fixed such that E_1 is proportional to h_1. For each configuration, we train the model to convergence and record the best random seed. To assess whether the trained model retains the information of the argmax token at each head, we fit a linear map $g_i : \mathbb{R}^{E_1} \to \mathbb{R}^d$ by regression for each head $i = 1, \ldots, h_1$, and measure the normalized recovery error

$$\mathcal{E}_i = \mathbb{E}_{X_T} \mathbb{E}_t \frac{|g_i(x'_1(t)) - x(s_i(t))|^2}{|x(s_i(t))|^2}, \tag{57}$$

where $s_i(t) = \arg\max_s A_i(x(t), x(s))$ is the argmax position of head i at position t. We report $\mathcal{E}_i$ for both the initialized and trained model, as well as the ratio of trained to initialized error, which measures the degree to which training improves argmax position recovery.

Results. The recovery errors for each configuration are reported in Tables 2 to 7. In all configurations, training reduces the recovery error substantially compared to initialization, with trained-to-initialized ratios ranging from 0.08 to 0.84. The reduction is most pronounced in configurations where the number of first-layer heads h_1 matches or exceeds the intrinsic dimension $D = 6$, consistent with the InfoFlow prediction that $h_1 \geq D$ heads are needed for effective approximation of the target.

Table 2: Recovery errors for $h_1 = 6$, $h_2 = 6$, embedding dimension $E_1 = 36$, seed 12.

	Head 1	Head 2	Head 3	Head 4	Head 5	Head 6
Initialized	0.3497	0.3546	0.3233	0.3738	0.4851	0.3496
Trained	0.0302	0.0323	0.0310	0.0305	0.0315	0.0317
Trained/Initialized	0.0863	0.0910	0.0958	0.0817	0.0650	0.0906

Table 3: Recovery errors for $h_1 = 5$, $h_2 = 5$, embedding dimension $E_1 = 40$, seed 9.

	Head 1	Head 2	Head 3	Head 4	Head 5
Initialized	0.3496	0.3711	0.3676	0.3113	0.3836
Trained	0.1816	0.1735	0.2433	0.2373	0.2350
Trained/Initialized	0.5195	0.4676	0.6618	0.7624	0.6126

Discussion. The results across all configurations consistently show that training improves the recoverability of argmax position information from the post-attention token state. The improvement is most pronounced when $h_1 \geq D$, where the trained-to-initialized ratio drops to as low as 0.08, indicating near-perfect recovery after training. When $h_1 < D$, the improvement is more modest, which is consistent with the InfoFlow prediction that insufficient head count prevents the model from independently tracking each of the D relevant argmax positions.

C.2 Second-Largest Position Retrieval in Trained Two-Layer Transformers

Setup. Using the same experimental configurations as Section C, we assess whether trained two-layer Transformers retain the information of the token attaining the second-largest attention score at each head.

Results. The recovery errors for each configuration are reported in Tables 5 to 11.

Discussion. In sharp contrast to the argmax recovery experiment, training does not improve the ability to recover the second-largest position information across any configuration. The trained-to-initialized ratios are uniformly close to 1.0 across all heads and configurations, indicating that the trained model retains no more information about the second-largest attention position than an untrained model. This provides empirical confirmation that softmax attention concentrates exclusively on the argmax position, and that the max-position retrieval mechanism identified in Theorem 2 and Section 4 is not merely a theoretical phenomenon but is precisely what emerges in trained networks.

C.3 Specific Position Aggregation Experiment

Setup. We consider sequences $X_T = (x(1), \ldots, x(T))$ with $T = 10$ and token dimension $d = 2$, where each token $x(t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$. The target function is

$$F(X_T) = x(1) + x(2) + x(3) \in \mathbb{R}^d, \tag{58}$$

namely the sum of the first three tokens, which is independent of the remaining $T - 3$ tokens and of the specific values at positions 1, 2, 3.

Model. We train a one-layer one-head Transformer with learned positional encoding. Each token $x(t)$ is first mapped to a d_{model}-dimensional embedding via a linear layer, and a learned positional embedding $p(t) \in \mathbb{R}^{d_{\text{model}}}$ is added. A learnable CLS token, initialized to zero in embedding space and assigned positional embedding $p(T + 1)$, is appended to the sequence. The model applies one round of single-head attention with a residual connection and layer normalization, then reads out the CLS token through a two-layer MLP to produce the prediction $\hat{F}(X_T) \in \mathbb{R}^d$.

Training. We train on $N = 50{,}000$ i.i.d. samples with MSE loss using AdamW with learning rate 10^{-3} and weight decay 10^{-2}, with a cosine annealing schedule over 1000 epochs followed by 1000 epochs at learning rate 10^{-4}. Validation is performed on a held-out set of $10{,}000$ samples.

Table 4: Recovery errors for $h_1 = 5$, $h_2 = 6$, embedding dimension $E_1 = 40$, seed 8.

	Head 1	Head 2	Head 3	Head 4	Head 5
Initialized	0.3167	0.3248	0.3413	0.3234	0.3261
Trained	0.2099	0.2531	0.1160	0.1382	0.1405
Trained/Initialized	0.6629	0.7793	0.3400	0.4273	0.4308

Table 5: Second-largest position recovery errors for $h_1 = 6$, $h_2 = 6$, embedding dimension $E_1 = 36$, seed 12.

	Head 1	Head 2	Head 3	Head 4	Head 5	Head 6
Initialized	0.4085	0.4126	0.3843	0.4303	0.5318	0.4086
Trained	0.4061	0.4154	0.4176	0.4191	0.4132	0.4148
Trained/Initialized	0.9940	1.0067	1.0864	0.9739	0.7770	1.0151

Results. The model achieves near-zero validation MSE. Inspecting the learned attention weights, the CLS token's mean attention over the validation set concentrates with weight $1/3$ on each of positions 1, 2, 3 and weight 0 on all remaining positions. This pattern is stable across samples regardless of token values, demonstrating that the model implements position-specific selection purely through the positional encoding rather than through content-dependent attention. The attention weights are shown in Figure 4.

C.4 Intrinsic Dimension Experiment

Target. We train two-layer Transformers on the permutation-invariant target

$$F(X_T) = \sum_{i=1}^{D} \max_{1 \le s,t \le T} x(s)^\top A_i x(t), \tag{59}$$

where $A_i \in \mathbb{R}^{d \times d}$ are distinct randomly generated matrices fixed throughout training. We consider intrinsic dimensions $D \in \{2, 3, 4, 5, 6\}$ with sequence length $T = 64$ and token dimension $d = 4$.

Model. For each value of D, we train two-layer Transformers with three head configurations: $(h_1, h_2) = (D, D)$, $(h_1, h_2) = (D-1, D)$, and $(h_1, h_2) = (D, D-1)$. The total embedding dimension is fixed at $E_1 = \lceil \frac{6D}{h_1} \rceil h_1$, $E_2 = \lceil \frac{6D}{h_2} \rceil h_2$. The feed-forward hidden width is $w_1 = 16D$. No positional encoding is used, consistent with the permutation-invariant nature of the target.

Training. Each configuration is trained for 16 independent runs with different random seeds. We use the AdamW optimizer with learning rate 10^{-3} and weight decay 10^{-2}, with a cosine annealing schedule. Training is performed on $N = 36{,}000$ i.i.d. samples drawn from $x(t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$, with a held-out validation set of $4{,}000$ samples. We report the best validation NMSE across training, averaged over all 16 seeds.

Results. The mean best validation NMSE for each configuration and value of D is reported in Table 12. Results are also shown in Figure 1a.

An ablation with wider FFN of $w_1 = 32D$ instead of $w_1 = 16D$ is reported in Table 13.

Across all values of D, the (D, D) configuration achieves substantially lower NMSE than the $(D-1, D)$ and $(D, D-1)$ configurations, with a gap of two to three orders of magnitude for $D \le 5$. For $D \ge 6$, the gap narrows but remains consistent, which we attribute to the increased difficulty of training with larger head counts within the fixed sequence length $T = 64$.

C.5 Triangle-Center Experiment

Target. We train two-layer Transformers on the triangle-center target

$$F(X_T) = \min_{1 \le t_1, t_2, t_3 \le T} \|x(t_1) + x(t_2) + x(t_3)\|_2^2, \tag{60}$$

Table 6: Second-largest position recovery errors for $h_1 = 5$, $h_2 = 5$, embedding dimension $E_1 = 40$, seed 9.

	Head 1	Head 2	Head 3	Head 4	Head 5
Initialized	0.4093	0.4272	0.4249	0.3746	0.4390
Trained	0.4095	0.4099	0.4097	0.4082	0.4083
Trained/Initialized	1.0005	0.9594	0.9641	1.0895	0.9299

Table 7: Second-largest position recovery errors for $h_1 = 5$, $h_2 = 6$, embedding dimension $E_1 = 40$, seed 8.

	Head 1	Head 2	Head 3	Head 4	Head 5
Initialized	0.3779	0.3852	0.4013	0.3849	0.3873
Trained	0.4018	0.4038	0.4085	0.4050	0.4068
Trained/Initialized	1.0631	1.0483	1.0180	1.0523	1.0505

with token dimension $d = 2$ and sequence lengths $T \in \{3, 4, 5, 6, 8, 10, 12, 16, 24, 32, 48, 64\}$. Each token $x(t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 2I_2)$, clamping min norm at 0.2 and max norm at 4.

Model. We train three two-layer Transformer configurations of increasing size: $(h_1, h_2) = (2, 2)$ with embedding dimension $E = 24$, $(h_1, h_2) = (4, 4)$ with embedding dimension $E = 24$, and $(h_1, h_2) = (4, 4)$ with embedding dimension $E = 48$. The feed-forward hidden width is $w_1 = 32$ for all configurations. No positional encoding is used.

Training. Each configuration and sequence length is trained for 2 independent runs with different random seeds. We use the AdamW optimizer with learning rate 10^{-3} and weight decay 10^{-2}, with a cosine annealing schedule. Training is performed on $N = 90{,}000$ i.i.d. samples, with a held-out validation set of $10{,}000$ samples. We report the best validation NMSE across training.

Results. The mean best validation NMSE for each configuration and sequence length is reported in Table 14. Results are also shown in Figure 1b.

An ablation with wider and deeper Transformer Models is reported in Table 15.

All three two-layer configurations exhibit rapidly growing NMSE as T increases, with NMSE exceeding 0.84 at $T = 16$ and approaching 1.0 for $T \geq 32$. These three configurations perform almost identically across all values of T, indicating that increasing the number of heads or the embedding dimension does not improve approximation within the tested range, consistent with the **InfoFlow** prediction. With deeper and larger models, as shown in Table 15, we can still see rapidly growing NMSE as T increases even though these deeper models exhibit smaller NMSE for smaller T. This is also consistent with the **InfoFlow** prediction.

Remark. NMSE, namely Normalized Mean Square Error, is equal to 1 if the model predicts the best constant value regardless of input ($M(X_T) \equiv c$ for constant c independent of X_T), which can be achieved by Transformers of any size. NMSE approaching 1 means that Transformer is not performing better than a constant predictor, which shows that Transformers are not approximating the target efficiently.

Table 8: Second-largest position recovery errors for $h_1 = 5$, $h_2 = 7$, embedding dimension $E_1 = 40$, seed 15.

	Head 1	Head 2	Head 3	Head 4	Head 5
Initialized	0.3854	0.3954	0.3863	0.4124	0.3869
Trained	0.4044	0.4077	0.4017	0.4109	0.3996
Trained/Initialized	1.0493	1.0313	1.0399	0.9963	1.0329

Table 9: Second-largest position recovery errors for $h_1 = 6$, $h_2 = 5$, embedding dimension $E_1 = 36$, seed 1.

	Head 1	Head 2	Head 3	Head 4	Head 5	Head 6
Initialized	0.4025	0.3800	0.4280	0.4466	0.4230	0.3976
Trained	0.4045	0.4027	0.4032	0.4042	0.4028	0.4085
Trained/Initialized	1.0049	1.0598	0.9420	0.9049	0.9524	1.0272

Table 10: Second-largest position recovery errors for $h_1 = 7$, $h_2 = 5$, embedding dimension $E_1 = 42$, seed 13.

	Head 1	Head 2	Head 3	Head 4	Head 5	Head 6	Head 7
Initialized	0.3813	0.3905	0.4137	0.3863	0.3812	0.4053	0.3964
Trained	0.4001	0.4026	0.4046	0.4042	0.3996	0.4067	0.4029
Trained/Initialized	1.0493	1.0310	0.9781	1.0461	1.0483	1.0034	1.0163

Table 11: Second-largest position recovery errors for $h_1 = 7$, $h_2 = 7$, embedding dimension $E_1 = 42$, seed 10.

	Head 1	Head 2	Head 3	Head 4	Head 5	Head 6	Head 7
Initialized	0.4740	0.3949	0.3801	0.3922	0.3955	0.4124	0.4001
Trained	0.4062	0.4148	0.4147	0.4054	0.4098	0.3903	0.3868
Trained/Initialized	0.8569	1.0506	1.0909	1.0337	1.0364	0.9464	0.9668

Table 12: Mean best validation NMSE for the intrinsic dimension experiment with $T = 64$, averaged over 16 random seeds.

D	(D, D)	$(D-1, D)$	$(D, D-1)$
2	5.89×10^{-5}	5.89×10^{-2}	3.67×10^{-2}
3	6.37×10^{-5}	2.64×10^{-2}	2.07×10^{-2}
4	6.92×10^{-5}	1.39×10^{-2}	1.30×10^{-2}
5	4.69×10^{-4}	4.46×10^{-3}	7.19×10^{-3}
6	1.99×10^{-3}	6.26×10^{-3}	7.67×10^{-3}

Table 13: Best validation NMSE (mean $\pm$ standard deviation over 16 seeds) for the intrinsic dimension experiment with $T = 64$ and FFN hidden width $32D$. This is an ablation with wider FFN than the $w_1 = 16D$ width results used in the plot.

D	(D, D)	$(D-1, D)$	$(D, D-1)$
2	$4.71 \times 10^{-5} \pm 6.41 \times 10^{-6}$	$5.55 \times 10^{-2} \pm 4.65 \times 10^{-3}$	$3.39 \times 10^{-2} \pm 1.25 \times 10^{-3}$
3	$4.93 \times 10^{-5} \pm 3.19 \times 10^{-6}$	$2.43 \times 10^{-2} \pm 9.24 \times 10^{-3}$	$2.02 \times 10^{-2} \pm 3.82 \times 10^{-3}$
4	$5.67 \times 10^{-5} \pm 2.98 \times 10^{-6}$	$1.46 \times 10^{-2} \pm 4.25 \times 10^{-3}$	$1.36 \times 10^{-2} \pm 4.26 \times 10^{-3}$
5	$5.81 \times 10^{-4} \pm 2.02 \times 10^{-3}$	$4.21 \times 10^{-3} \pm 2.07 \times 10^{-3}$	$7.81 \times 10^{-3} \pm 7.59 \times 10^{-4}$
6	$1.99 \times 10^{-3} \pm 4.50 \times 10^{-3}$	$7.87 \times 10^{-3} \pm 3.53 \times 10^{-3}$	$9.81 \times 10^{-3} \pm 4.49 \times 10^{-3}$

Table 14: Mean best validation NMSE for the triangle-center experiment, averaged over 2 random seeds.

T	$(2,2)$, $E=24$	$(4,4)$, $E=24$	$(4,4)$, $E=48$
3	1.31×10^{-3}	9.59×10^{-4}	6.20×10^{-4}
4	4.58×10^{-3}	3.81×10^{-3}	3.03×10^{-3}
5	1.12×10^{-2}	9.51×10^{-3}	8.41×10^{-3}
6	2.12×10^{-2}	2.00×10^{-2}	1.84×10^{-2}
8	8.61×10^{-2}	7.20×10^{-2}	7.95×10^{-2}
10	2.42×10^{-1}	2.27×10^{-1}	2.13×10^{-1}
12	4.32×10^{-1}	4.60×10^{-1}	4.52×10^{-1}
16	8.43×10^{-1}	8.53×10^{-1}	8.56×10^{-1}
24	9.44×10^{-1}	9.44×10^{-1}	9.44×10^{-1}
32	9.57×10^{-1}	9.58×10^{-1}	9.59×10^{-1}
48	9.79×10^{-1}	9.79×10^{-1}	9.79×10^{-1}
64	9.85×10^{-1}	9.85×10^{-1}	9.85×10^{-1}

Table 15: Best validation NMSE (mean $\pm$ standard deviation over 4 seeds) for the triangle-center experiment with deeper and larger Transformers.

T	$L=3,\ E=36$	$L=4,\ E=48$
3	$7.88\times10^{-4}\pm9.52\times10^{-4}$	$6.73\times10^{-4}\pm8.05\times10^{-4}$
6	$4.42\times10^{-3}\pm3.40\times10^{-4}$	$4.02\times10^{-3}\pm1.96\times10^{-4}$
8	$2.13\times10^{-2}\pm1.72\times10^{-3}$	$1.64\times10^{-2}\pm1.12\times10^{-3}$
16	$6.73\times10^{-1}\pm4.04\times10^{-2}$	$6.96\times10^{-1}\pm4.87\times10^{-2}$
32	$9.61\times10^{-1}\pm1.57\times10^{-3}$	$9.61\times10^{-1}\pm1.72\times10^{-3}$
64	$9.83\times10^{-1}\pm6.24\times10^{-4}$	$9.84\times10^{-1}\pm5.02\times10^{-4}$

C.6 bAbI Question-Answering Experiment

Tasks and arity. We use Tasks 1, 2, and 3 of the bAbI question-answering benchmark [Weston et al., 2015]. Each example includes an annotation of the facts needed to answer its question. Tasks 1, 2, and 3 require one, two, and three supporting facts, respectively; we call this number the *arity*. Under the retrieval-task interpretation used by **InfoFlow**, arity is the number of input positions whose information must be combined to produce the answer.

Input construction. Each fact is represented as one input token, so T is the number of candidate facts searched by the model. We vary $T \in \{20, 40, 80, 128\}$ by inserting answer-preserving distractor facts. The entity pool is held fixed (6 identities) so that increasing T changes the amount of search without changing the arity of the question or the number of candidate entities.

Model and evaluation. We train each Transformer from scratch with $h = 4$ attention heads, embedding width $E = 16$, and depth $L \in \{1, 2, 3, 4\}$, with 5 answer-preserving padding variants per source example. Training this architecture is *bimodal*: for a fixed configuration, a random seed either converges to the retrieval circuit (training accuracy $\gtrsim 0.95$) or remains near the plateau induced by the architecture's default initialization, with little in between. A best-of-few statistic can therefore substantially overstate how reliably a configuration trains. We report, over $N = 10$ random seeds per cell, the *solve rate* — the fraction of seeds reaching training accuracy above 0.9 — as our primary statistic (Table 16), together with the mean $\pm$ standard deviation of training accuracy across the same 10 seeds (Table 17). Arity 1 is included as a trainability control and is deterministic across seeds ($N = 3$; training accuracy 1.00, zero variance), so it is omitted from the bimodality statistics.

Table 16: Solve rate on bAbI Tasks 2–3: fraction of $N = 10$ seeds reaching training accuracy > 0.9, at $h = 4$, $E = 16$. Arity 1 always solves (10/10, $N = 3$; deterministic) and is omitted. Columns give the number T of candidate facts.

Arity 2

$L \backslash T$	20	40	80	128
1	0/10	0/10	0/10	0/10
2	10/10	10/10	9/10	7/10
3	10/10	10/10	10/10	7/10
4	10/10	10/10	10/10	9/10

Arity 3

$L \backslash T$	20	40	80	128
1	0/10	0/10	0/10	0/10
2	1/10	7/10	2/10	0/10
3	6/10	9/10	8/10	2/10
4	6/10	10/10	9/10	4/10

Table 17: Mean $\pm$ standard deviation of training accuracy across the same $N = 10$ seeds as Table 16. This is a supplementary summary; because the underlying distribution is bimodal (Table 16), mean and std can understate how often a configuration fully solves the task and should be read alongside, not instead of, the solve rate.

Arity 2

$L \backslash T$	20	40	80	128
1	.40±.08	.53±.08	.56±.09	.48±.06
2	.99±.00	.99±.01	.95±.12	.83±.26
3	1.00±.00	1.00±.00	1.00±.00	.86±.22
4	1.00±.00	1.00±.00	1.00±.00	.95±.14

Arity 3

$L \backslash T$	20	40	80	128
1	.45±.01	.39±.03	.53±.01	.48±.03
2	.53±.15	.84±.20	.69±.14	.53±.09
3	.80±.23	.92±.19	.91±.12	.56±.23
4	.80±.24	1.00±.00	.93±.11	.68±.25

Results. Arity 1 is fitted at every tested depth and sequence length on every seed (deterministic; omitted from the tables above). At depth $L = 1$, arity 2 and arity 3 are never solved at any T (solve rate $0/10$ throughout, training accuracy capped around 0.4–0.7): a single attention layer cannot chain the multi-hop retrieval either task requires, regardless of context length. This is the depth floor predicted by the staged pairwise retrieval underlying Theorem 1.

At $L \geq 2$, arity 2 solves reliably: solve rate is 9–$10/10$ at every depth for $T \leq 80$, and remains comparatively high at $T = 128$ (7–$9/10$). At $L = 2$, arity 3 solves on only $1/10$ seeds at $T = 20$ and $2/10$ at $T = 80$ despite reaching $7/10$ at $T = 40$. We attribute this to the optimization/data-density sensitivity noted for other short-sequence cells rather than to an arity effect. The pattern that survives across depths and is monotone in T is the collapse at $T = 128$: solve rate there falls to $0/10$ ($L = 2$), $2/10$ ($L = 3$), and $4/10$ ($L = 4$), while arity 2 at the same $T = 128$ stays at 7–$9/10$. Increasing depth

from $L = 2$ to $L = 3, 4$ raises the solve rate at intermediate T (e.g. $T = 80$: $2/10 \to 8/10 \to 9/10$), but still collapse for larger T such as $T = 128$. This is consistent with Proposition 2 and the triangle-center example (8): the arity-3 decode cost degrades with a growing exponent in $T^{2/3}/(LE)$, so additional depth L (at fixed width E) delays but cannot remove inefficiency at large-T, whereas arity 2's cost at this width shows no comparable T-dependence over the range tested.

D Explanations for Number of Comparison 5.3

For the generalized D-retrieval target $F(X_T) = F_0(\oplus_{i=1}^{D} \max_{1\le t\le T} f_i(x(t)))$, it has $\beta' = \beta_1 = 1$, and the Number of Comparison of F is upper bounded by $D(T-1)$ and lower bounded by $D(T-D)$. For the target $F(X_T) = \min_{1\le t_1,t_2,t_3\le T} \|x(t_1) + x(t_2) + x(t_3)\|_2^2$, we can find that it has $\beta' = \beta_1 = 3$, and its Number of Comparison is upper bounded by $T^3 - 1$, also lower bounded by $\Omega(T^3)$.

D.1 Explanations for the Examples

Generalized D-retrieval Target We first show the upper bound by the following construction, which is intuitive.

1. It is clear that $|I_F(X_T)| \le D, a.e.$, so we construct D Trees of Comparison.
2. For $j = 1, \dots, D$, $\mathcal{T}_j$ have exactly T leaves, each leaf being a single-element index set $\{t\}$ for $t = 1, \dots, T$. And $f_{in}(x) = f_j(x)$ for all internal nodes of $\mathcal{T}_j$. Thus $I_{fi}(X_T, \mathcal{T}_j) = \arg\max_{1\le t\le T} f_j(x(t))$.
3. We then have $I_F(X_T) = \bigcup_{j=1}^{T} I_{fi}(X_T, \mathcal{T}_j)$.
4. These D Trees of Comparison each has T leaves, thus it upper bounds the Number of Comparison of F by $D(T-1)$.

Assuming that each $f_i(z)$ obtains maximum at different z_i, and non-degeneracy condition as in Yu et al. [2026]. We then show that the Number of Comparison is lower bounded by $D(T-D)$.

1. It is clear that $\beta_1 = 1$. As all t's can appear in $I_F(X_T)$, so $\{t\}$ has to be some leaf of some Tree. As there are at most D Trees, We have $N' \ge T - D$, thus $\beta' = 1$.
2. As $|I_F(X_T)| = D, a.e.$ and $|I_{fi}(X_T, \mathcal{T}_j)| = 1$, thus we need exactly D Trees of Comparisons.
3. For each $\mathcal{T}_j$, denote by $I_j \subseteq [T]$ the union of index sets on all its leaves. And Number of Comparison is calculated with $\sum_{j=1}^{D}(|I_j| - 1)$.
4. Assume that $|I_j| \le T-D$ for some j. Then suppose $t_1, \dots, t_D \notin I_j$. Then let $x(t_i) = z_i$ for $i = 1, \dots, D$, we have that $I_F(X_T) = \{t_1, \dots, t_D\}$. However, $[I_F(X_T) \cap I_{fi}(X_T, \mathcal{T}_j)] \subseteq [I_F(X_T) \cap I_j] = \varnothing$. Then we have that
$$
\begin{aligned}
&|I_F(X_T) \cap \bigcup_{i=1}^{T} I_{fi}(X_T, \mathcal{T}_i)| && (61)\\
=&|I_F(X_T) \cap \bigcup_{i\ne j} I_{fi}(X_T, \mathcal{T}_i)| && (62)\\
\le&|\bigcup_{i\ne j} I_{fi}(X_T, \mathcal{T}_i)| && (63)\\
\le&D - 1 && (64)\\
<&|I_F(X_T)| && (65)
\end{aligned}
$$
which contradicts with $|I_F(X_T) = \bigcup_{i=1}^{T} I_{fi}(X_T, \mathcal{T}_i)|$.
5. Then we have $|I_j| \ge T - D + 1$ for all $j = 1, \dots, D$, and thus $N' \ge D(T - D + 1) - D = D(T - D)$.

We then successfully showed that the Number of Comparison of $F(X_T) = F_0(\oplus_{i=1}^{D} \max_{1\le t\le T} f_i(x(t)))$ is $DT + O(1)$.

Triangle-Center Target For triangle-center target $F(X_T) = \min_{1\le t_1,t_2,t_3\le T} \|x(t_1) + x(t_2) + x(t_3)\|_2^2$, we show that it has $\beta_1 = 3$ and $\beta' = 3$, with Number of Comparison upper bounded by $T^3 - 1$ and lower bounded by $\Omega(T^3)$ (Or more specifically, $\frac{T^3}{6} + O(T^2)$). (For this simplicity we consider $\mathcal{X} = [-1, 1]^d$ for this case, where similar results can be transformed back to $\mathcal{X}^T = [0, 1]^T$.)

We first show that its Number of Comparison is upper bounded by $T^3 - 1$.

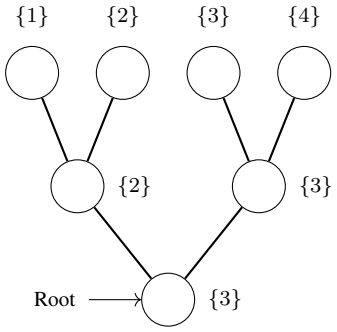
1. We construct one Tree of Comparison, with each leaf being an ordered set (t_1, t_2, t_3) that allows repetition. There are T^3 choices of such ordered set.
2. For all internal nodes, they have the same $f_{in}(X[I]) = -\| \sum_{t\in I} x(t)\|_2^2$. Thus at the root, we have $I_{fi}(X_T, \mathcal{T}) = \arg\max_{I\subseteq[T],|I|=3} \| \sum_{t\in I} x(t)\|_2^2 = I_F(X_T)$.

With the above construction, we have that the Order and Dimension of Comparison upper bounded by 3, namely $\beta' \leq \beta_1 \leq 3$.

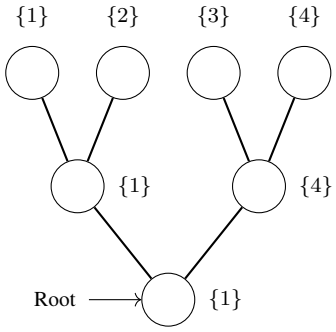
We then show the lower bound. Consider $\mathcal{X}^T = [-1, 1]^{Td} \subset \mathbb{R}^{Td}$ as part of the Td-dimensional Euclidean Space. Now for two given triple pairs (unordered) of indices $I_0 = (t_1, t_2, t_3)$ and $J_0 = (s_1, s_2, s_3)$ where t_i, s_j are pairwise distinct (We assume $T > 6$). And denote by f the function $f(X[I]) = \| \sum_{t\in I} x(t)\|_2^2$ for index set I. We define the following sets:

1. $\Gamma'' = \{X_T \in \mathcal{X}^T : f(X[I_0]), f(X[J_0]) < f(X[I'])$ for all $I' \neq I_0, J_0$; $|I'| = 3\}$. Γ is the set where $X[I_0]$ and $X[J_0]$ obtain the minimum two for f. Under the uniform measure μ on $\mathcal{X}^T = [-1, 1]^{Td}$, we have $\mu(\Gamma'') > 0$.
2. $\Gamma''_l = \{X_T \in \mathcal{X}^T : f(X[I_0]) < f(X[J_0]) < f(X[I'])$ for all $I' \neq I_0, J_0$; $|I'| = 3\}$.
   $\Gamma''_r = \{X_T \in \mathcal{X}^T : f(X[J_0]) < f(X[I_0]) < f(X[I'])$ for all $I' \neq I_0, J_0$; $|I'| = 3\}$.
   We have that $\Gamma''_r + \Gamma''_l = \Gamma''$; $\mu(\Gamma''_l) = \mu(\Gamma''_r) = \frac{\mu(\Gamma'')}{2} > 0$.
3. $\gamma'' = \{X_T \in \mathcal{X}^T : f(X[I_0]) = f(X[J_0]) < f(X[I'])$ for all $I' \neq I_0, J_0$; $|I'| = 3\} \subset \Gamma''$ is a $Td - 1$ dimensional manifold in Γ''. As γ'' is in $\mathbb{R}^{Td}$, we can define the area measure ν on γ'', and we have $\nu(\gamma'') > 0$.
4. By symmetry, we can find a connected component $\Gamma \subseteq \Gamma''$ that is separated into two pieces Γ_l, Γ_r such that $\mu(\Gamma) = 2\mu(\Gamma_l) = 2\mu(\Gamma_r) > 0$, and $\gamma = \gamma'' \cap \Gamma$ also have $\nu(\gamma) > 0$.

Now for each pairs of unordered triple I', J' that is not the same as I_0, J_0, let $\gamma' = \{X_T \in \mathcal{X}^T : g(X[I_1]) = g(X[J_1])\} \cap \Gamma$ for non-degenerate g (namely γ' have dimension $Td - 1$). As $\{I', J'\} \neq \{I_0, J_0\}$, we have that $\gamma' \cap \gamma$ is a manifold of dimension $Td - 2$ (As we cannot have $I_0 \subseteq I', J_0 \subseteq J'$ together), thus $\nu(\gamma' \cap \gamma) = 0$. Suppose each γ' separates Γ into Γ'_l and Γ'_r, and we have a finite number M of such pairs $(I', J', g, \gamma', \Gamma'_l, \Gamma'_r)$, we have that $\nu[(\cup\gamma') \cap \gamma] = 0$.
By continuity and and compactness of each γ', as $\cup\gamma'$ cannot interpolate γ in a dense manner, thus if we use all the γ''s to split Γ into (at most) 2^M subsets (In the sense whether a point in Γ lies on the "left" or "right" of γ', and one such subset is for points on the same side for all γ'.) We call these subsets $\Gamma_1, \ldots, \Gamma_K$ for $K = 2^M$. Then we have that $\min_{I_k\subseteq[K]} \mu(\Gamma_l \triangle (\bigcup_{i\in I_k} \Gamma_i)) = \min_{I_k\subseteq[K]} \mu(\Gamma_r \triangle (\bigcup_{i\in I_k} \Gamma_i)) > 0$.
This shows that we cannot use many comparisons of other pairs (I', J') to exactly characterize the comparison of (I_0, J_0). Thus when $\beta_1 = 3$, and that the Tree of Comparison have finite leaves, (Thus finite number of possible pairwise comparison performed, where $I_{fi}(X_T, \mathcal{T})$ depends only on the comparison results in the form of separation.) I_0 and J_0 need to be on the leaves of a Tree of Comparison, namely the comparison of I_0, J_0 should be among the possible pairwise comparisons. By extending this argument to all I_0, J_0 pair, we have that all unordered $I_0 = \{t_1, t_2, t_3\}$ with distinct t_1, t_2, t_3 should be on the leaf of at least one Tree of Comparison. This gives a total number of $\binom{T}{3}$ leaves. With $|I_F(X_T)| \leq D_0 = 3, a.e.$, we have the Number of Comparison of F lower bounded by $\binom{T}{3} - 3$. Thus the Number of Comparison of F is lower bounded by $\frac{T^3}{6} + O(T^2)$.



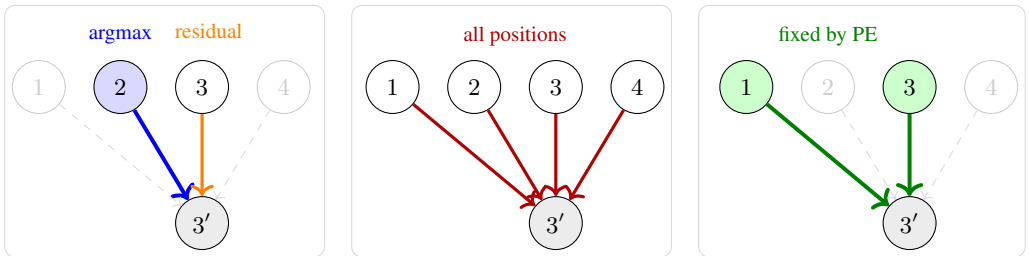
(a) $\mathcal{T}_1$ with $f_{\{in\}}(x) = x$ at all internal nodes, finding the maximum. The root gives $\{3\} = I_{\{f\}}(X_T, \mathcal{T}_1)$, the position of the maximum value.



(b) $\mathcal{T}_2$ with $f_{\{in\}}(x) = -x$ at all internal nodes, finding the minimum. The root gives $\{1\} = I_{\{f\}}(X_T, \mathcal{T}_2)$, the position of the minimum value.

Figure 2: Two comparison trees for the target $F(X_T) = \max_{1 \le t \le T} x(t) + \min_{1 \le t \le T} x(t)$ with $D = 2$, illustrated on the input $x(1) = 0.1$, $x(2) = 0.3$, $x(3) = 0.4$, $x(4) = 0.2$ with $T = 4$. Each leaf corresponds to one input position, and each internal node retains the index of the winning position after comparison. The root of each tree gives the active index set $I_{fi}(X_T, \mathcal{T})$.

E Figures



I: Max-position **II**: Global aggregation **III**: Specific position

Figure 3: Illustration of the three information propagation mechanisms in attention layers, shown for a sequence of four tokens. The output token $3'$ is shown at the bottom of each subfigure, with arrows indicating which input tokens contribute to it. **I**: Max-position retrieval aggregates information from the argmax position (blue, shaded) via attention and from its own position via the residual connection (orange). **II**: Global aggregation aggregates information from all input positions. **III**: Specific position aggregation aggregates information from a fixed set of positions (green, shaded) determined by positional encoding alone, independent of token content. Faded tokens with dashed arrows indicate positions that do not contribute.

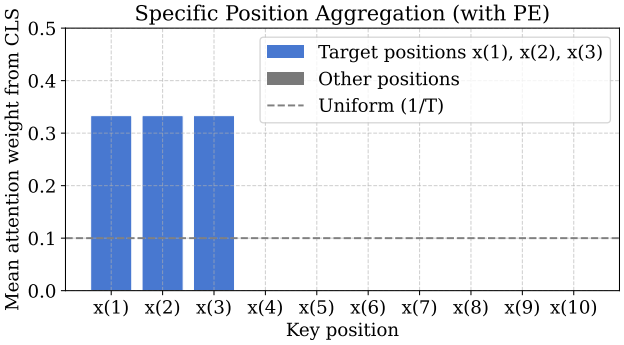


Figure 4: Mean CLS attention weight per input position, averaged over the validation set, for a Transformer trained with positional encoding on $F(X_T) = x(1)+x(2)+x(3)$. Attention concentrates entirely on positions 1, 2, 3 with weight $1/3$ each, while all other positions receive zero weight, confirming position-specific selection independent of token content. This is the figure for the specific position aggregation experiment in C.3

Illustrative example of InfoFlow. We illustrate the two-step procedure on a concrete example. Consider the target $F(X_T) = \min_{1\le s,t\le T} x(t)^\top x(s)$ and a two-layer Transformer with one head per layer. Let the input sequence be $x(1) = (0, -1)$, $x(2) = (0.7, 0.7)$, $x(3) = (0, 1)$, $x(4) = (-0.2, -0.9)$, so that $T = 4$.

We apply update rule (1) max position aggregation at both layers. At layer 0, the information sets are initialized as $I(t, 0) = \{t\}$ for $t = 1, 2, 3, 4$ and $I(5, 0) = \varnothing$. For the first layer, letting the attention score function be $A_{1,1}(X[I(t, 0)], X[I(s, 0)]) = \min_{t'\in I(t,0), s'\in I(s,0)} x(t')^\top x(s')$, we compute the argmax positions

$$s_{1,0}(1) = 3, \quad s_{1,0}(2) = 4, \quad s_{1,0}(3) = 1, \quad s_{1,0}(4) = 3. \tag{66}$$

Applying the max position aggregation rule gives

$$I(1, 1) = \{1, 3\}, \quad I(2, 1) = \{2, 4\}, \quad I(3, 1) = \{1, 3\}, \quad I(4, 1) = \{3, 4\}. \tag{67}$$

For the second layer, the classification token at position 5 reads out from the sequence. Letting the attention score function be $A_{2,1}(X[I(5, 1)], X[I(s, 1)]) = \min_{s_1,s_2\in I(s,1)} x(s_1)^\top x(s_2)$, we obtain $s_{1,1}(5) = 1$, and therefore

$$I(5, 2) = I(s_{1,1}(5), 1) = I(1, 1) = \{1, 3\}. \tag{68}$$

The active index set of the target is $I_F(X_T) = \{1, 3\}$, since the minimum $\min_{s,t} x(t)^\top x(s)$ is attained at $(s, t) = (1, 3)$. For this X_T we have $I_F(X_T) \subseteq I(5, 2)$. The evolution of information sets across layers is illustrated in Figure 5.

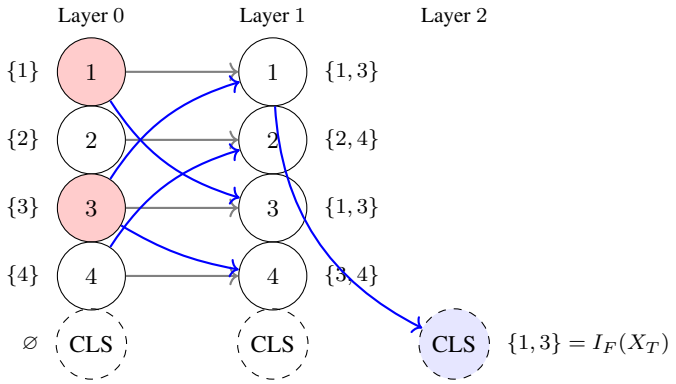


Figure 5: Illustration of InfoFlow on the target $F(X_T) = \min_{1\le s,t\le T} x(t)^\top x(s)$ with a two-layer one-head-per-layer Transformer. Each node shows the information set $I(t, l)$ at position t and layer l. Blue arrows indicate argmax information flow selected by each attention head. Gray arrows indicate residual connections preserving the token's own information. Red-shaded nodes at layer 0 indicate the active index set $I_F(X_T) = \{1, 3\}$. After two layers, the classification token accumulates $I(5, 2) = \{1, 3\} = I_F(X_T)$, confirming that the InfoFlow model learns the target.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The introduction has discussed main contribution.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a limitation section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All proofs are presented in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiment setting details are presented in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: The experiment settings are clearly provided in the appendix, which can be easily reproduced.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed settings are provided in the appendix.a

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experiments are performed over multiple seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The experiments does not require strong computation powers.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: NeurIPS Code of Ethics are followed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Do not have such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: Did not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: There is no release of new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: There is no crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No involve of crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Core methods are original.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.