HDT: Hierarchical Document Transformer

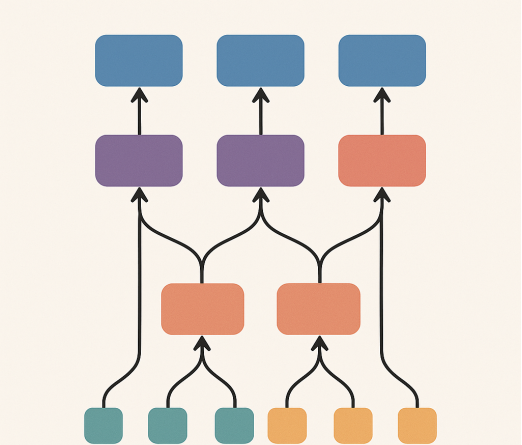
Haoyu He; Markus Flicke; Jan Buchmann; Iryna Gurevych; Andrea Geiger

University of Tübingen, Tübingen AI Center, Technical University of Darmstadt and Hessian Center for AI (hessian.AI)

Sep, 2025

Presented by Scott Sun from Duke B&B

ChatGPT art



Introduction

HDT is a sparse Transformer architecture tailored for structured hierarchical (long) documents. Auxiliary *anchor tokens* are introduced during the text tokenization, and the novel *sparse attention kernel* leads to faster convergence, higher sample efficiency and better representation learning.

Goal: 1) improving sample efficiency & generalization by **imposing the document's hierarchy structure as an inductive bias**; 2) adapting the attention to the hierarchy structure to obtain **sparse representation, which has reduced time & space complexity** (beneficial to long text learning)

- **Flash Attention:** Reduce memory I/O operations between HBM and SRAM so that the attention matrices are efficiently computed in SRAM
- **Hierarchical Structure:** Words $\rightarrow$ Sentences $\rightarrow$ Sections $\rightarrow$ Document
  This structure is largely ignored by many existing LM, which typically consider text as a "flat" sequence of tokens.

Background: Standard Transformer

For a single head in MHA, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n \times d_k}$,

$$\boldsymbol{A} = \frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}} \quad \boldsymbol{S} = \text{softmax}(\boldsymbol{A}) \quad \boldsymbol{O} = \boldsymbol{S}\boldsymbol{V} \tag{1}$$

Attention & causal masks are applied during the softmax operation. Then, $\boldsymbol{O} \in \mathbb{R}^{n \times d_k}$ from all heads are concatenated and passed another linear projection...The time & memory complexity is $O(n^2)$

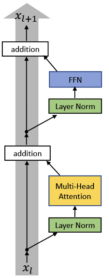


Figure: Transformer block [Xiong et al, 2020] that almost everyone agrees in 2025, except for Grok-1 (maybe its descendants? Only OpenAI and xAI know.)

Background: Hierarchical Attention Transformer

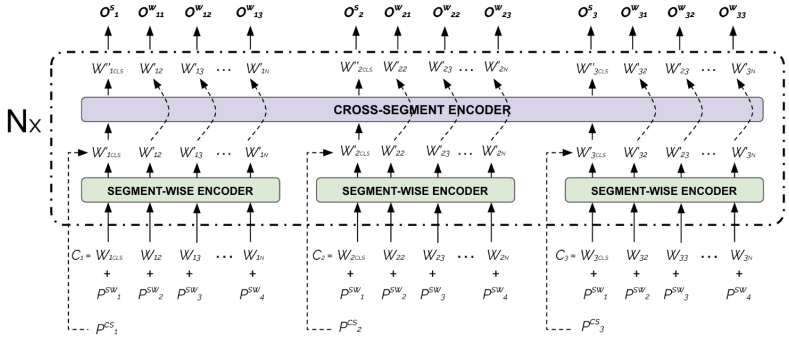


Figure: HAT block [Chalkidis et al, 2022]

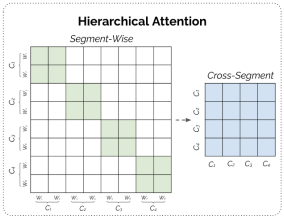


Figure: HAT Attention

Method: HDT Overview

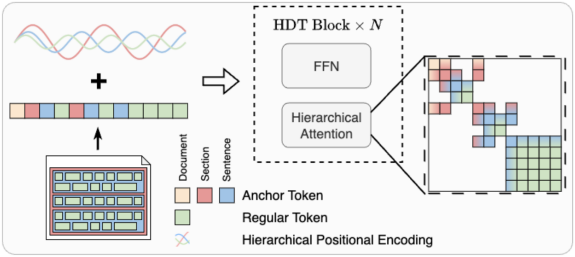


Figure: HDT Overview

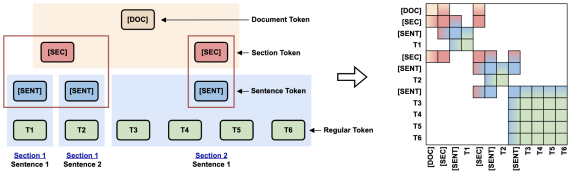


Figure: HDT structure & decomposition. Attention is restricted to parent–child pairs and sibling nodes.

Method: Hierarchical Positional Encoding

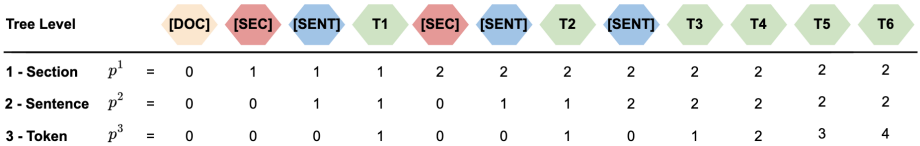


Figure 3: Hierarchical Positional Encoding. We represent the position of each token in the hierarchy with one linear index p^l per hierarchy level l yielding an index vector $\mathbf{p} = (p^1, \ldots, p^L)^T$. Above, we show an example with $L = 3$ levels. Note that level 0 (document) does not require an index. Each index in $\mathbf{p}$ is passed through sinusoidal encoding functions which are summed over all levels to form the final encoding vector according to Eq. (2).

Let $\boldsymbol{p} = (p^1, ..., p^L)^\top$ be the hierarchical position vector and i be the feature index.

$$\text{HPE}(\boldsymbol{p}, i) = \sum_{l=1}^{L} \begin{cases} \sin(\omega_k p^l) & \text{if } i = 2k \\ \cos(\omega_k p^l) & \text{if } i = 2k+1 \end{cases} \qquad \text{where } \omega_k = \frac{1}{10000^{2k/d_{\text{model}}}} \tag{2}$$

Borrowing the idea from RoPE-ViT, we can also apply a hierarchical version of RoPE.

Method: Hierarchical Attention Kernel

Different from the original MHA, the attention masking pattern $\boldsymbol{M}$ is more complex. For a 3-level document structure, we can compute the attention mask at each level separately:

- **within DOC & across SEC:** $M_{ij}^{\text{DOC}} = [p_i^2 = 0][p_j^2 = 0]$
- **within SEC & across SENT:** $M_{ij}^{\text{SEC}} = [p_i^3 = 0][p_j^3 = 0][p_i^1 = p_j^1]$
- **within SENT:** $M_{ij}^{\text{SENT}} = [p_i^1 = p_j^1][p_i^2 = p_j^2]$

Method: Hierarchical Attention Kernel (cont'd)

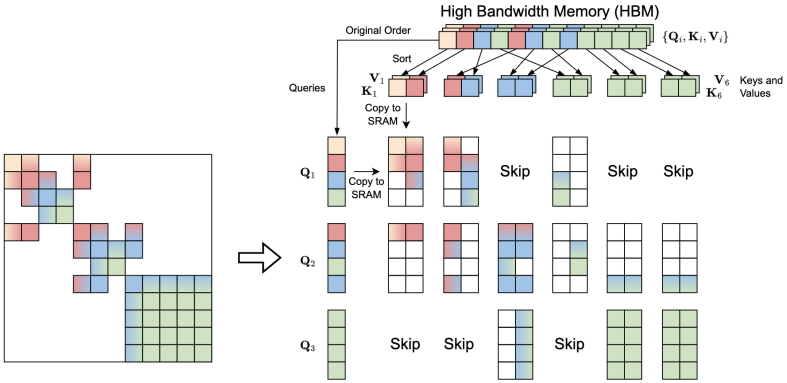


Figure: FlashAttention does not natively support this sparse attention masking pattern. The PyTorch `SDPA_backend` for FA only expects regular attention masks. The authors implemented their own Block-Sparse FA, but it remains less efficient than the HDT kernel.

The new HDT kernel involves two major steps:

1. KV-sort (convert BFT to DFT)
2. FlashAttention on Q and sorted KV

The order of the columns in $\boldsymbol{A}$ will be reverted in the end.

Experiment: Math Reasoning via ListOps

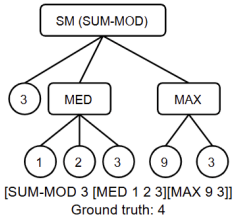


Figure 5: ListOps Sample

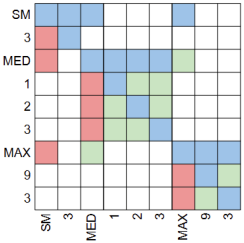


Figure 6: ListOps Attention

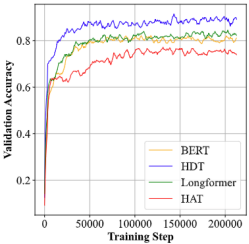


Figure 7: ListOps Convergence

Figure: ListOps [Nikita & R., 2018] is a proof-of-concept experiment demonstrating that HDT can capture structural relationships more effectively than other transformer encoder-only models.

The special structure of the problem admits of further increasing sparsity by using a "causal" mask:

1. operands do not attend to their operators (green+blue)
2. operands do not attend to each other (blue only)

HDT acc. performance: 79 $\rightarrow$ 85 $\rightarrow$ 86, which means HDT can better understand directional graph.

Experiment: Encoder-only Language Tasks

Model	Full Text	SciDocs				Feeds-M	High. Infl.	Avg.
		Cite	CoCite	CoView	CoRead			
Pretrained Only								
SciBERT$_{\text{base}}$		53.75	66.73	66.37	53.20	63.18	40.80	57.34
Longformer	✓	56.64	71.92	71.51	61.57	63.66	43.82	61.52
HAT	✓	60.14	75.55	73.62	67.65	65.02	**45.81**	64.63
HDT-E	✓	**62.50**	**78.51**	**75.69**	**72.12**	65.19	43.60	**66.27**
HDT-E (-arXiv)	✓	59.03	76.05	72.85	71.71	**65.41**	43.69	64.79
Pretrained + Finetuned with Contrastive Learning								
SciNCL @684k		64.77	81.67	78.55	77.48	70.22	48.66	70.23
SciNCL @19k		62.56	82.29	77.84	75.84	67.11	46.23	68.65
Longformer @19k	✓	61.75	79.87	78.20	74.25	67.80	43.85	67.62
HAT @19k	✓	63.46	81.24	**79.43**	75.76	69.31	47.37	69.42
HDT-E @19k	✓	**64.23**	**82.44**	78.95	**77.09**	**71.22**	**49.37**	**70.55**
HDT-E @19k (-arXiv)	✓	63.34	82.18	79.06	76.78	70.64	48.95	70.16

Table 2: Results on SciRepEval Proximity Tasks. Top: Models pre-trained with MLM without fine-tuning. Bottom: Models pre-trained with MLM and fine-tuned using SciNCL's contrastive learning objective. Full text documents are available only for a subset of 19k training triplets. For reference, we also report the results of the original SciNCL model which is trained on all 684k title+abstract triplets. We also report HDT-E pre-trained without arXiv data (-arXiv) to study the impact of scientific documents as pre-training data to our model's performance on the SciRepEval tasks which are in the scientific domain. All numbers are mean average precision. SciBERT and SciNCL use only title and abstract as input.

Experiment: Encoder-Decoder Language Tasks

Model	Purpose	Method	Findings	Value	Purpose-ZS	Method-ZS
LED_{base}	39.51	19.31	19.22	**24.26**	18.54	14.12
HDT-ED-[SEC]	30.70	17.65	17.15	18.90	15.01	12.67
+[SENT]	34.29	19.72	18.38	20.53	19.67	13.94
+ tokens	**40.60**	**22.21**	**22.11**	22.11	**21.75**	**15.43**

Table 3: Results on FacetSum Summarization Task. Following the original paper, we report ROUGE-L as the metric here. For HDT-ED-[SEC], the decoder cross-attends only to the section anchor token [SEC]. We observe that even when attending only to the anchor tokens (+[SENT]), our model is on par with LED, where the decoder attends to all tokens of the section, demonstrating the expressiveness of the learned intermediate representation of anchor tokens. When attending to additional regular tokens (+tokens), our model outperforms LED. We also report zero-shot (ZS) performance for "Purpose" and "Method", training only on "Findings" and "Value".

Model	GovRep ROUGE-1/2/L	SumScr ROUGE-1/2/L	QMSum ROUGE-1/2/L	Qspr F1	Nrtv F1	QALT EM-T/H	CNLI EM	Avg Score
LED_{base}	**56.2/26.6/28.8**	24.2/4.5/15.4	25.1/**6.7**/**18.8**	26.6	**18.5**	25.8/25.4	71.5	29.16
HDT-ED	49.8/22.2/25.8	**30.8**/**7.1**/**18.6**	**28.3**/**6.7**/18.7	**33.1**	14.2	**29.4**/**26.4**	**81.4**	**31.41**

Table 4: Results on the SCROLLS Summarization, QA and NLI Benchmark. We compare HDT-ED (pre-trained for 12 GPU days) to Longformer-Encoder-Decoder (LED) on the official SCROLLS benchmark *without document structure*. We choose LED as baseline as it has a comparable number of parameters (162M) to HDT-ED (124M). We remark that neither model is competitive with state-of-the-art billion-parameter models such as CoLT5 XL (score 43.5) which are trained on large GPU clusters.

Recommendation

Is it worth reading? Yes.

- the framework is thorough and elegant; the experimental interpretation is straightforward
- the paper gives clear illustration of how the sparse attention kernel is designed & developed using the tiling algorithm

Is it worth implementing? Yes (if there is enough computing power).

- they have a public github repo which includes the code to reproduce all the results; the models are uploaded to Hugging Face
- I'd be very interested to see how the HDT attention kernel could enhance my EHR foundational model