HLA: Hadamard Linear Attention

Anonymous CVPR submission

Paper ID *****

Abstract

The attention mechanism is an important reason for the success of transformers. To reduce the high computational cost of standard quadratic attention, linear attention has been proposed. It applies kernel functions to the inputs before the pairwise similarities are calculated. Although that allows for an efficient computational procedure, it reduces the expressive power of linear attention, leading to worse results than softmax-based attention.

We propose Hadamard Linear Attention (HLA). In contrast to others works on linear attention, the nonlinearity in HLA is applied after the pairwise similarities have been computed, analogously to standard softmax attention. An efficient computation scheme for the proposed method is derived that is similar to that of standard linear attention. The effectiveness of the approach is demonstrated by applying it to a large diffusion transformer model for video generation, an application that involves very large amounts of tokens.

1. Introduction

The transformer architecture has demonstrated remarkable success across a wide range of domains, for instance, question answering [24], reasoning [28], and even tasks such as 3D reconstruction [27]. An important limitation of the attention mechanism used in transformers is its quadratic complexity with respect to the length of the input sequence. Although recent advances [2, 3, 22] have mitigated the quadratic memory complexity, the computational complexity remains quadratic. Linear attention mechanisms have been proposed as scalable alternatives. However, despite their improved efficiency, linear attention-based transformers [13, 20] exhibit reduced performance compared to those that use standard attention.

In standard linear attention, nonlinear transformations are applied *prior* to computing pairwise interactions, resulting in a low-rank approximation of the full attention matrix. This contrasts with softmax-based attention, where the nonlinearity is applied *after* the pairwise interactions have been computed. In this work, we introduce a novel nonlinearity for linear attention, *Hadamard Linear Attention (HLA)*. Unlike existing approaches, the proposed nonlinearity can also be applied *after* the pairwise interactions have been computed. We derive an efficient computation scheme for the proposed attention mechanism, preserving the scalability benefits of linear attention.

An important advantage of the proposed algorithm is that it can be directly applied to the sequence, unlike other algorithms [14, 32] which first require the reshape the tensor. While this reshaping operation is not measured by metrics such as the number of floating point operations (FLOPs), it may cause a higher latency especially on memory-constrained platforms. Furthermore, Hadamard Linear Attention considers the entire sequence through its attention mechanism whereas [14] only has a limited view via its receptive field.

2. Related Works

Linear attention has been proposed to solve the quadratic memory requirement and computation complexity that standard softmax-based attention has [13, 23]. It has been noted that algorithms relying on linear attention often perform poorly compared to algorithms using standard attention. The authors of [20] point out that linear attention is equivalent to fast weight programmers [21]. The same authors also propose to use a decay factor to hinder retention of irrelevant information over a sequence via the Delta rule.

[1, 16, 17, 19] propose how to better approximate the exponential function in the context of linear attention. The low-rank constraint inherent to linear attention is a likely cause of the poor performance of linear attention. A solution has been proposed in [5]. Other approaches have been proposed as well, for instance based on maximal coding rate reduction [30] or locality sensitive hashing [8].

Recently, [32] and [6] combined softmax attention with linear attention for video diffusion. This idea is to model large values in the attention matrix by softmax, yet smaller ones by linear attention. While these hybrid methods offer good performance, they require dynamical indexing[1] and are

[1]Directly selecting one or multiple individual values in a tensor is not

Figure 1. Results generated using the proposed video diffusion transformer (**HLA-3F-R1-10**, see sec. 5.1 for more details) using Hadamard Linear Attention.

not softmax-free

3. Attention

3.1. Softmax Attention

Let there be N tokens $\boldsymbol{X} = \left[\boldsymbol{x}_0^\top, \ldots, \boldsymbol{x}_N^\top\right] \in \mathbb{R}^{N \times d}$ where the scalar d indicates the number of elements each of the vectors $\boldsymbol{x}_i$ has. These input tokens are mapped to queries Q, keys K and values V by right-multiplying with learnable matrices W_q, W_k, W_v. The attention scores are the product QK^T normalized by *softmax*. The attention mechanism used in [25] uses them to compute a convex combination of the values $\mathrm{softmax}\left(\frac{1}{\sqrt{d}} QK^\top\right) V$.

3.2. Linear Attention

Linear attention has been proposed [13, 20, 20] as a solution to both problems. Its idea is to approximate the exp function using a separable kernel

$$\exp\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \approx K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle. \qquad (1)$$

Due to the separability of the kernels, the order of matrix multiplication is changed for linear attention

$$\phi(Q)\left(\phi(K)^\top \cdot V\right) = \phi(Q) \cdot C \qquad (2)$$

supported by some low-power compute units.

which avoids explicitly computing the attention scores whereby the complexity is reduced to $\mathcal{O}(N)$ since $\phi(K)^\top \cdot V$ forms the $d \times d$ *context* matrix. Matrix C in equation 2 is often called the context.

4. Hadamard Linear Attention

4.1. Definition

Separable kernel functions are the basis of linear attention and enable compute and memory-efficient attention procedures. Since the nonlinearities are applied before the pairwise interactions are computed, linear attention models a low-rank approximation of pairwise interactions [32]. This is unlike *softmax*-based attention that applies the nonlinearity to the pairwise interactions.

The contribution made in this paper is to propose both a novel type of linear attention mechanism and a nonlinearity that is novel in linear attention. In contrast to existing works, this novel nonlinearity can be applied *after* the pairwise interactions have been computed. The proposed attention mechanism is therefore more similar to standard *softmax* attention. Although it would appear that such a model prohibits an efficient computational procedure to calculate attention, we will prove that our proposed attention mechanism allows for that, similar to linear attention.

Let $\phi_q : d \to d_\phi$ and $\phi_{k_f} : d \to d_\phi$ be nonlinear transformations with $f = 1, \ldots, F$. We define our attention

operator as follows

$$A = \left(\phi_q(Q) \cdot \phi_{k_1}(K)^\top\right) \odot \left(\phi_q(Q) \cdot \phi_{k_2}(K)^\top\right) \odot \cdots \tag{3}$$

where the symbol $\odot$ indicates the element-wise product, also called Hadamard product. Instead of *softmax* as nonlinearity, equation 3 uses the element-wise products $\prod_{f=1}^{F} \phi_q(q_i)^\top \cdot \phi_{k_f}(k_j)$. The attention update can be computed by right-multiplying A with the value matrix V.

4.2. Efficient Attention

Naïvely evaluating $A \cdot V$ by equation 3 has quadratic computational complexity. We now derive an efficient strategy for computing the attention, which will reveal a procedure analogous to that of standard linear attention. In the following, indicate by a summation symbol without any indices $\sum$ a summation over all elements of the argument.

Lemma 4.1. *Given $F = 2$ factors for the Hadamard product in equation 3, we may express the product involving the d-dimensional vectors q, r^1 and r^2 as*

$$\left(q^\top \cdot r^1\right) \cdot \left(q^\top \cdot r^2\right) = \sum \underbrace{\left(q \cdot q^\top\right)}_{d\times d} \odot \underbrace{\left(r^1 \cdot r^{2\top}\right)}_{d\times d} \tag{4}$$

where the sum is over all the $\text{len}(q)^2$ elements.

For $F > 2$, the relation involving the d-dimensional vectors q and r^f generalizes to

$$\prod_{f=1}^{F} \left(q^\top \cdot r^f\right) = \sum \underbrace{\left(q \otimes \cdots \otimes q\right)}_{d\times\cdots\times d} \odot \underbrace{\left(r^1 \otimes \cdots \otimes r^F\right)}_{d\times\cdots\times d} \tag{5}$$

where the symbol $\otimes$ denotes the outer tensorial product and the summation is over all the elements.

Let $\mathcal{T}_q$ be the $N \times d_\phi \times \cdots d_\phi$ dimensional tensor that contains the N F-fold tensorial outer products of the vectors $\phi_q(q_i)$ with themselves, ie the ith slice, $i = 1, \ldots, N$, is equivalent to $[\mathcal{T}_q]_i = \phi_q(q_i) \otimes \cdots \otimes \phi_q(q_i)$. Analogously, define by $\mathcal{T}_k$ the $N \times d_\phi \times \cdots d_\phi$ tensor that contains the N F-fold tensorial outer products of the vectors $\phi_{k_f}(k_j)$.

Theorem 4.2. *The attention defined by equation 3 can be equivalently expressed by the product between $\mathcal{T}_q$ and a $d_\phi \times \cdots \times d_\phi \times d$ dimensional context tensor $\mathcal{C}$*

$$\mathcal{T}_q \otimes \mathcal{C}. \tag{6}$$

Please note that although equation 6 appears similar to the formulation of plain linear attention, it contains higher-order terms of the keys in $\mathcal{C}$ via $\mathcal{T}_k$. In other words, linear attention computes attention scores linear in the keys, whereas HLA computes attention scores F-fold in the keys.

Lastly, we need to ensure that each row of matrix A sums up to 1. We may perform this similarly to standard linear attention if we require that $\phi_{q,k_1,k_2,k_3}(\cdot) \geq 0$. The factors η_i that normalize each row of the matrix in equation 3 to sum 1 are given by the contraction of the tensor $\mathcal{T}_q \odot \mathcal{T}_k$ over all dimensions except the first

$$\eta = \sum_{l,m,\ldots} [\mathcal{T}_q \odot \mathcal{T}_k]_{l,m,\ldots}. \tag{7}$$

4.3. Causal Attention, Decay Factors and Sequential Updates

Since we require that $\phi_{q,k_1,k_2,k_3}(\cdot) \geq 0$ for normalization, we may easily consider, for instance, causal attention by left-multiplying an $n \times n$ matrix M to the keys $\phi_{k_f}(K)$. For causal attention, the entries of M are chosen to be $\{0, 1\}$. The mask can include decay factors as proposed in [20].

5. Experiments

5.1. Implementation Details

Data We fine-tune our model variants using a 350K subset of the data used by OpenSoraPlan [15]. We also use about 100K synthetic videos generated by Wan2.1 14B.

Model The proposed attention mechanism is integrated into the Wan2.1 1.3B model for video diffusion [4]. Two resolutions are used: first, the smaller of two resolutions used by the original model (81x480x832); second, a lower solution more suitable for fast video generation (81x320x480). The sequence lengths are 32760 and 12600 tokens, respectively. We indicate the two resolutions by **R1** and **R2**.

We evaluate with two variants of Hadamard Linear Attention. The first one employs three factors in equation 3, whereas the second only uses two factors. For the first, we use small MLPs ϕ_q, $\phi_{k_j}, j = \{1, 2, 3\}$ and $\phi_{v_l}, l = \{1, 2\}$. For the second, we use slightly larger MLPs.

The two HLA variants are included into four variants of Wan: **HLA-2F-R1-21** uses two factors in equation 3, a resolution of 81x480x832, and applies Hadmard Linear Attention to 21 out of 30 transformer blocks. **HLA-3F-R1-21** is identical to **HLA-2F-R1-21** except that it uses three factors in equation 3. **HLA-3F-R1-10** is identical to **HLA-3F-R1-21** except that it applies HLA to 10 transformer blocks. Lastly, **HLA-3F-R2-15** uses three factors for the Hadamard product in 15 out of 30 transformer blocks and resolution 81x320x480. Please see the supplementary material for more information about the model definitions.

Evaluation *VBench* [10] scores are reported to assess the visual quality and report the floating-point operations (FLOPs) of each model variant to measure the computational complexity. We evaluate against two checkpoints of Wan2.1 [4], one at 480p provided by the original authors, the second at 320p trained by us. We do not compare with recent

	Total↑	Quality↑	Semantic↑
Models with at most 2B parameters			
Open-Sora1.2 [18]	79.76	81.35	73.39
SnapGenV [29]	81.14	83.47	71.84
LTX-Video [7]	80.00	82.30	70.79
CogVideoX [31]	81.55	82.48	77.81
Hummingbird [11]	81.35	83.73	71.84
M4V [9]	81.91	83.36	76.10
PyramidFlow [12]	81.72	84.74	69.62
Wan2.1-1.3B [26]	83.31	85.23	75.65
Wan2.1-1.3B-HLA			
$\sim 23\%$ less total compute			
HLA-3F-R1-10	81.42	83.22	74.20

Table 1. Comparison on VBench with other models for video diffusion. We report the numbers published in the respective papers. While the proposed method has slightly lower scores, it requires less than 80% of the TFLOPs used by Wan2.1-1.3B.

sophisticated variants of linear attention that use windowing, delta rule or hybrid methods that combine standard linear attention with softmax-attention. These techniques can be integrated into HLA, or HLA can replace standard linear attention in hybrid methods.

5.2. Main Results

We compare the performance of **HLA-3F-R1-10** with several state-of-the-art models of comparable number of parameters in table 1. Higher numbers indicate better results. Compared with Wan2.1-1.3B, the proposed model has about 23% lower computational complexity (cf. table 2) but only slightly lower vbench scores.

5.3. Ablations

Table 2 shows a comparison of different model variants. They differ in whether they use two or three factors in the Hadamard product in equation 3 (indicated by **2F** or **3F**), the video resolution they were trained with (**R1** or **R2**) and the number of Hadamard Linear Attention Layers they use (**10**, **15** or **21**). It can be seen that the VBench scores slightly reduce while the number of HLA layers increases. Conversely, the computational complexity reduces. The time necessary to generate a single video is perceivably lower for **HLA-3F-R1-21** compared to the baseline. **HLA-3F-R1-21** requires almost half the number of floating-point operations (FLOPs) as the baseline. We also see that 3-factor HLA provides better performance than 2-factor, despite that the 2-factor version has similar FLOPs (by using larger MLPs). The reason that the computation time does not reduce as much as the number of FLOPs lies in the number of compute units that allow massive parallelization. Less capable accelerators, for instance those widely used in mobile applications, can be expected to benefit even more from the proposed algorithm. This shows that using higher-degree HLA is a more effective way to scale up model capacity.

5.4. Complexity

Table 3 compares the computational complexities of the four different HLA variants compared with standard quadratic attention. As in Sec. 5.1, we use slightly larger MLPs for HLA with 2 factors. Depending on the resolution and the algorithmic variant, HLA requires between 20% and 90% less floating-point operations (FLOPs) than quadratic attention.

Method	Tokens	Score	TFLOPs	Time
Wan2.1 (quad.)	32760	83.3	283.03	1:36
Wan2.1 (quad.)	12600	81.0	62.13	0:45
HLA-3F-R1-10	32760	81.42	218.59	1:26
HLA-3F-R2-15	12600	79.78	48.37	0:26
HLA-2F-R1-21	32760	79.02	147.16	1:13
HLA-3F-R1-21	32760	80.54	147.71	1:16

Table 2. VBench scores and times to generate a single video of several model variants. TFLOPs are for a single forward pass through the model. All measurement were done on an H100 GPU.

Attn Type	Resolution	Tokens	TFLOPs
Wan2.1 (quad.)	320x480	12600	1.21
Wan2.1 (quad.)	480x832	32760	7.21
HLA 2 factors	320x480	12600	0.97
HLA 2 factors	480x832	32760	2.52
HLA 3 factors	320x480	12600	0.30
HLA 3 factors	480x832	32760	0.77

Table 3. Computational complexity of models with 2 or 3 factors in the Hadamard product in equation 3 for 12600 or 32760 tokens.

6. Conclusions

We proposed Hadamard Linear Attention (HLA), a novel algorithm to compute attention. Unlike traditional linear attention algorithms, HLA applies a nonlinearity after the pairwise similarities have been computed. We derived an efficient formulation that reduces the complexity to $\mathcal{O}(N)$. HLA relies on standard tensor operations and can directly operate on the input sequence without requiring any time-consuming tensor reshaping. We demonstrated good results on the challenging application of video diffusion which involves extremely long sequences. For typical sequence lengths, our proposed HLA requires only about $1/10$ of the FLOPs that baselines need. HLA can replace standard linear attention in hybrid algorithms like [32] and [6] to yield improved performance. Likewise, HLA can be extended with windowing or the delta-rule to improve the algorithm.

References

[1] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2022. 1

[2] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 1

[3] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 1

[4] Team Wan et al. Open and advanced large-scale video generative models, 2025. 3

[5] Qihang Fan, Huaibo Huang, and Ran He. Breaking the low-rank dilemma of linear attention, 2025. 1

[6] Mohsen Ghafoorian, Denis Korzhenkov, and Amirhossein Habibian. Attention surgery: An efficient recipe to linearize your video diffusion transformer, 2025. 1, 4

[7] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion, 2024. 4

[8] Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David P. Woodruff, and Amir Zandieh. Hyperattention: Long-context attention in near-linear time, 2023. 1

[9] Jiancheng Huang, Gengwei Zhang, Zequn Jie, Siyu Jiao, Yinlong Qian, Ling Chen, Yunchao Wei, and Lin Ma. M4v: Multi-modal mamba for text-to-video generation, 2025. 4

[10] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[11] Takashi Isobe, He Cui, Dong Zhou, Mengmeng Ge, Dong Li, and Emad Barsoum. Amd-hummingbird: Towards an efficient text-to-video model, 2025. 4

[12] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling, 2025. 4

[13] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 5156–5165, 2020. 1, 2

[14] Pierre-David Letourneau, Manish Kumar Singh, Hsin-Pai Cheng, Shizhong Han, Yunxiao Shi, Dalton Jones, Matthew Harper Langston, Hong Cai, and Fatih Porikli. Padre: A unifying polynomial attention drop-in replacement for efficient vision transformer, 2024. 1

[15] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, Tanghui Jia, Junwu Zhang, Zhenyu Tang, Yatian Pang, Bin She, Cen Yan, Zhiheng Hu, Xiaoyi Dong, Lin Chen, Zhang Pan, Xing Zhou, Shaoling Dong, Yonghong Tian, and Li Yuan. Open-sora plan: Open-source large video generation model, 2024. 3

[16] Weikang Meng, Yadan Luo, Xin Li, Dongmei Jiang, and Zheng Zhang. Polaformer: Polarity-aware linear attention for vision transformers. In *International Conference on Learning Representations (ICLR)*, 2025. 1

[17] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[18] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, Yuhui Wang, Anbang Ye, Gang Ren, Qianran Ma, Wanying Liang, Xiang Lian, Xiwen Wu, Yuting Zhong, Zhuangyan Li, Chaoyu Gong, Guojun Lei, Leijun Cheng, Limin Zhang, Minghao Li, Ruijie Zhang, Silan Hu, Shijie Huang, Xiaokang Wang, Yuanheng Zhao, Yuqi Wang, Ziang Wei, and Yang You. Open-sora 2.0: Training a commercial-level video generation model in $200k, 2025. 4

[19] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In *International Conference on Learning Representations (ICLR)*, 2022. 1

[20] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2021. 1, 2, 3

[21] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992. 1

[22] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision, 2024. 1

[23] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Winter Conference on Applications of Computer Vision (WACV)*, 2024. 1

[24] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2

[26] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng

Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. 4

[27] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2024. 1

[28] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[29] Yushu Wu, Zhixing Zhang, Yanyu Li, Yanwu Xu, Anil Kag, Yang Sui, Huseyin Coskun, Ke Ma, Aleksei Lebedev, Ju Hu, Dimitris Metaxas, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapgen-v: Generating a five-second video within five seconds on a mobile device, 2025. 4

[30] Ziyang Wu, Tianjiao Ding, Yifu Lu, Druv Pai, Jingyuan Zhang, Weida Wang, Yaodong Yu, Yi Ma, and Benjamin D. Haeffele. Token statistics transformer: Linear-time attention via variational rate reduction, 2024. 1

[31] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025. 4

[32] Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E. Gonzalez, Jun Zhu, and Jianfei Chen. Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention, 2025. 1, 2, 4