Learning What to Remember: Adaptive Probabilistic Memory Retention for Memory-Efficient Language Models

A Probabilistic Framework for Memory-Constrained Language Modeling

S M Rafiuddin
Department of Computer Science
Oklahoma State University
Stillwater, OK, USA
srafiud@okstate.edu

Muntaha Nujat Khan
Department of English
Oklahoma State University
Stillwater, OK, USA
munkhan@okstate.edu

Abstract

Transformer attention scales quadratically with sequence length $O(n^2)$, limiting long-context use. We propose *Adaptive Retention*, a probabilistic, layer-wise token selection mechanism that learns which representations to keep under a strict global budget M. Retention is modeled with Bernoulli gates trained via a Hard-Concrete/variational relaxation and enforced with a simple top-M rule at inference, making the method differentiable and drop-in for standard encoders. Across classification, extractive QA, and long-document summarization, keeping only 30–50% of tokens preserves $\geq$ 95% of full-model performance while cutting peak memory by $\sim$ 35–45% and improving throughput by up to $\sim 1.8\times$. This architecture-agnostic approach delivers practical long-context efficiency without modifying base attention or task heads.

1 Introduction

Transformer-based language models have achieved remarkable success across a wide range of NLP tasks (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020), but their memory requirements grow quadratically with sequence length, posing significant challenges for long-context processing (Tay et al., 2022). To address these limitations, various approaches have explored sparse attention patterns to reduce computational and memory overhead (Beltagy et al., 2020; Zaheer et al., 2020; Child et al., 2019; Dao, 2024) and compressed memory representations to extend effective context lengths (Rae et al., 2020; Wu et al., 2022; Kim et al., 2024). Recent work has further proposed dynamic memory and computation strategies, such as internet-scale memory compression (Zemlyanskiy et al., 2024), token merging for efficient inference (Bolya et al., 2023), and streaming memory management (Xiao et al., 2024). However, existing methods often either rely on architectural modifications or apply fixed heuristics without explicitly modeling adaptive token retention under strict memory budgets. Inspired by advances in adaptive computation time (Graves, 2016; Xin et al., 2020; Liu et al., 2020) and improvements in selective attention (Leviathan et al., 2024), our approach formulates memory retention as a probabilistic learning problem, enabling language models to dynamically retain the most informative token representations while respecting a global memory constraint. Through this formulation, our approach achieves significant memory savings with minimal performance degradation, offering a flexible and efficient solution applicable to standard transformer architectures.

2 Related Work

Memory-Efficient Self-Attention. The quadratic memory cost of Transformer attention (Vaswani et al., 2017) has prompted variants such as Reformer's locality-sensitive hashing achieving $O(L \log L)$ (Kitaev et al., 2020), Linformer's low-rank projections for $O(L)$ (Wang et al., 2020), and Performer's random-feature approximations (Choromanski et al., 2021). Sparse patterns appear in Longformer and BigBird (Beltagy et al., 2020; Zaheer et al., 2020), and FlashAttention-2 optimizes GPU memory access and parallelism (Dao, 2024).

Memory Compression and External Stores. Compressive Transformer compresses past activations into a secondary memory bank (Rae et al., 2020), while Memorizing Transformers add an explicit key–value store (Wu et al., 2022). Compact summaries are learned in CCM (Kim et al., 2024), and methods like MEMORY-VQ (Zemlyanskiy et al., 2024) and GLIMMER (de Jong et al., 2023) apply quantization or late-interaction retrieval.

Token pruning vs. KV-cache compression. Beyond learned or heuristic encoder-side pruning (e.g., H2O's fixed heavy-hitter oracle (Zhang et al.,

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 3969–3981
November 4-9, 2025

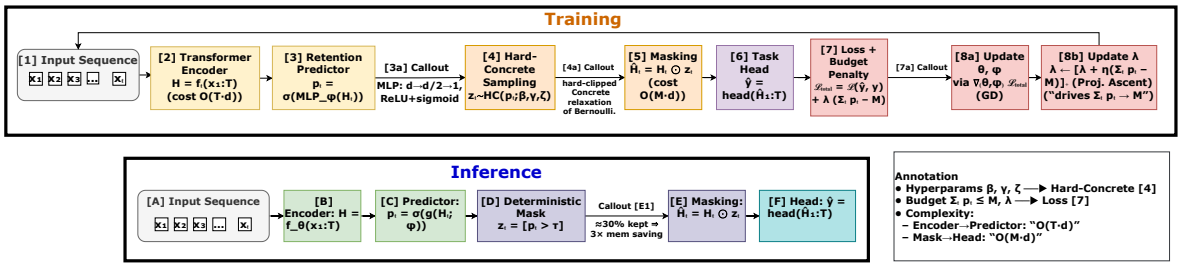

Figure 1: **Adaptive Retention: layer-wise probabilistic token selection.** At each Transformer block, a lightweight gated scorer produces per-token probabilities trained with a Hard–Concrete relaxation. At inference, we keep the top-M_l tokens per layer ($M_l = \lfloor \rho T_l \rfloor$), forwarding only those to the next block. The active sequence length shrinks with depth, yielding cumulative compute and memory savings while leaving base attention unchanged. Symbols: H^l token states; s^l scores; p^l probabilities; ρ target ratio; M_l retained count.

2023)), a parallel line compresses the *decoder-side* key–value (KV) cache for autoregressive generation, such as SnapKV (Li et al., 2024) and PyramidKV (Cai et al., 2024). These methods reduce memory/latency during *stepwise* decoding by selecting or merging past states, whereas our approach targets *encoder* representations and shrinks the active token set *within* the stack under a global budget. Consequently, direct apples-to-apples benchmarking is non-trivial: KV-cache compression operates in a causal, decode-time regime with different bottlenecks, while our method improves long-context *encoding* efficiency without modifying attention patterns. We therefore compare H2O empirically on encoder tasks and discuss SnapKV/PyramidKV qualitatively as complementary techniques for generative inference.

Adaptive Computation and Token Reduction. Adaptive Computation Time (ACT) varies processing steps per token (Graves, 2016), and early-exit models such as DeeBERT and FastBERT skip layers dynamically (Xin et al., 2020; Liu et al., 2020). Token Merging fuses similar representations (Bolya et al., 2023), and streaming attention sinks manage context retention for low-latency long-sequence processing (Xiao et al., 2024).

Our approach differs by learning probabilistic token retention under a strict global memory budget, enabling end-to-end optimization of which hidden states to store without altering core Transformer architectures.

3 Method

3.1 Problem Formulation

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ be the input sequence and let the Transformer encoder produce hidden states $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_T)$, $\mathbf{h}_t \in \mathbb{R}^d$. We introduce binary retention indicators $\mathbf{z} = (z_1, \ldots, z_T) \in \{0,1\}^T$, defined by

$$z_t = \begin{cases} 1, & \text{if } \mathbf{h}_t \text{ is retained} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

and denote the masked sequence $\mathbf{H} \odot \mathbf{z} = (\mathbf{h}_1 z_1, \ldots, \mathbf{h}_T z_T)$. Our goal is to learn both model parameters $\boldsymbol{\theta}$ and retention probabilities $\mathbf{p} = (p_1, \ldots, p_T)$, where $p_t = \Pr[z_t = 1]$, by solving

$$\min_{\boldsymbol{\theta}, \mathbf{p}} \mathbb{E}_{\mathbf{z} \sim \text{Bernoulli}(\mathbf{p})} \big[\mathcal{L}\big(f(\mathbf{H} \odot \mathbf{z}; \boldsymbol{\theta})\big) \big] : \sum_{t=1}^{T} p_t \le M \tag{2}$$

where M is the total memory budget (the maximum expected number of retained tokens). At inference, we deterministically retain the top-M tokens by their retention probabilities $\{p_t\}$.

Here, f is the task-specific decoder and $\mathcal{L}$ the loss; at inference one may further enforce the hard constraint $\sum_{t=1}^{T} z_t \le M$.

3.2 Probabilistic Retention Model

We introduce a lightweight summary state $\mathbf{m}_t \in \mathbb{R}^d$ and compute context-aware retention probabilities via a gated scoring network:

$$\begin{aligned} \mathbf{m}_t &= \gamma\, \mathbf{m}_{t-1} + (1-\gamma)\, \mathbf{h}_t, \quad \mathbf{m}_0 = \mathbf{0} \\ s_t &= \mathbf{v}^\top \tanh\big(\mathbf{W}\, \mathbf{h}_t + \mathbf{U}\, \mathbf{m}_{t-1}\big) + b \\ p_t &= \sigma(s_t), \quad z_t \sim \text{Bernoulli}(p_t) \end{aligned} \tag{3}$$

where $\gamma \in [0,1]$ controls the summary decay, $\sigma(x) = (1+e^{-x})^{-1}$, and $\{\mathbf{W}, \mathbf{U}, \mathbf{v}, b\}$ are learned parameters. This formulation captures both *local* (h_t) and *global* (m_{t-1}) context in the retention decision.

3.3 Optimization Objective

To enforce the budget in (2) we introduce a Lagrange multiplier $\lambda \ge 0$ and form the saddle-point

problem:

$$\max_{\lambda \geq 0} \min_{\boldsymbol{\theta}, \mathbf{p}} \mathcal{L}_\lambda(\boldsymbol{\theta}, \mathbf{p}) \tag{4}$$

where the Lagrangian is defined as

$$\mathcal{L}_\lambda(\boldsymbol{\theta}, \mathbf{p}) = \mathbb{E}_{\mathbf{z} \sim \text{Bernoulli}(\mathbf{p})} \big[\mathcal{L}\big(f(\mathbf{H} \odot \mathbf{z}; \boldsymbol{\theta})\big) \big] + \lambda \Big(\sum_{t=1}^{T} p_t - M \Big) \tag{5}$$

We optimize by alternating stochastic gradient descent on $(\boldsymbol{\theta}, \mathbf{p})$ and projected gradient ascent on λ, i.e. $\lambda \leftarrow \max\{0,\ \lambda + \eta(\sum_t p_t - M)\}$ at each iteration (Boyd and Vandenberghe, 2004).

3.4 Variational Relaxation

Direct backpropagation through the discrete sampling in (2) is not possible, so we employ the Hard Concrete reparameterization (Maddison et al., 2017; Louizos et al., 2018).

$$\begin{aligned} &\alpha_t = \exp(s_t), \quad u \sim \mathcal{U}(0,1) \\ &\tilde{z}_t = \text{Clamp}_{[0,1]}\Big(\sigma\big(\tfrac{\log \alpha_t + \log u - \log(1-u)}{\beta}\big)\,(\zeta - \gamma) + \gamma \Big) \end{aligned} \tag{6}$$

with temperature $\beta > 0$ and stretch parameters $\gamma < 0 < 1 < \zeta$. During training we replace z_t by $\tilde{z}_t$ in both the expectation and the Lagrangian $\mathcal{L}_\lambda$, yielding low-variance gradient estimates via the standard reparameterization trick.

3.5 Inference Strategy

At test time we compute retention scores p_t via (3) and then deterministically select the top-M tokens by setting

$$\phi = \text{the } M\text{-th largest of } \{p_t\}_{t=1}^{T}, \quad z_t^* = \mathbf{1}\{p_t \geq \phi\} \tag{7}$$

so that $\sum_{t=1}^{T} z_t^* = M$. The encoder outputs are then masked as $\mathbf{H} \odot \mathbf{z}^*$ and passed to $f(\cdot; \boldsymbol{\theta})$.

4 Experiments

4.1 Datasets & Baselines

Experiments are conducted on six benchmarks: **SST-2** (GLUE; short sentences; binary sentiment) (Socher et al., 2013; Wang et al., 2018), **IMDb** (full movie reviews; avg. 230 words; many > 512 tokens) (Maas et al., 2011), **ArXiv** (long scientific papers; avg. 5,000 tokens) (Clement et al., 2019; Beltagy et al., 2020), **QASPER** (long-document QA; Exact Match / F1) (Dasigi et al., 2021), **PubMed** (scientific summarization on the RCT subset; ROUGE-1 / ROUGE-L) (Xiong et al., 2024; Cohan et al., 2018), and **CUAD** (legal contract clause classification; Micro-F1 / Macro-F1) (Hendrycks et al., 2021). **Baselines comprise**: **Full Transformer** (dense, no retention), **Random Pruning** (token masking to meet the budget), **H2O** (fixed pruning) (Zhang et al., 2023), **Constraint-aware Pruning** (learned budgeted pruning) (Li et al., 2023), **Infor-Coef** (IB-based dynamic downsampling) (Tan, 2023), and sparse-attention models **Longformer** (sliding-window) (Beltagy et al., 2020) and **BigBird** (block-sparse with globals) (Zaheer et al., 2020); we also report zero-shot LLM references **GPT-3.5** (OpenAI, 2023), **Llama 2** (Touvron et al., 2023), **Llama 3** (Grattafiori et al., 2024), **Falcon** (Almazrouei et al., 2023), **Mistral** (Jiang et al., 2023), **Gemma** (Team et al., 2024), and **Phi4-Mini** (Dettmers et al., 2025). Our method is **Adaptive Retention**. *DistilBERT-base-uncased (≈66M) for SST-2/IMDb/CUAD* and *Longformer-base-4096 (≈149M) for ArXiv/QASPER/PubMed*.

4.2 Experimental Setup

We fine-tune `DistilBERT-base-uncased` on the short-context benchmarks, **SST-2**, **IMDb**, and **CUAD**, and `Longformer-base-4096` on the long-document benchmarks, **ArXiv**, **QASPER**, and **PubMed RCT**, under token-retention budgets $M/T \in \{0.5, 0.3\}$, comparing against the baselines described above. We optimize with `AdamW` (lr $= 3 \times 10^{-5}$; weight_decay $= 0.01$), training for three epochs on SST-2/IMDb/CUAD (batch size 32) and one epoch on ArXiv/QASPER/PubMed RCT (batch size 16). Hard Concrete relaxation uses $\beta = 0.66$, $\gamma = -0.1$, $\zeta = 1.1$, and the Lagrange multiplier λ is updated via projected ascent (step size $\eta = 1 \times 10^{-2}$).

4.3 Main Results

Extended analysis. Tables 1 and 2 (six benchmarks; three seeds, mean) show that **Adaptive Retention** (AR) preserves task accuracy while operating under strict 50%/30% token budgets and remains competitive with both pruning baselines and sparse-attention models.

Against dense (no retention). On **SST-2**, **Adaptive Retention** is within 0.6 pp at 50% (91.5 vs. 92.1) and within 1.6 pp at 30% (89.2 vs. 90.8). On **IMDb**, it attains 94.1/92.3 vs. 94.8/93.6; on **ArXiv** (R1) 80.9/79.5 vs. 81.3/80.1. For **QASPER**, **Adaptive Retention** *matches* dense F1 at both budgets (65.0 / 63.0) with EM just 0.2 pp lower (43.8 / 39.8 vs. 44.0 / 40.0). On **CUAD**, **Adaptive Retention** stays within 0.2–0.5 pp of dense

Table 1: Results at **50%** token retention on SST-2, IMDb, ArXiv (R1), QASPER (EM/F1), PubMed (R1/RL), and CUAD (micro/macro) across model groups: dense/no retention, learned/heuristic pruning (incl. H2O), sparse-attention architectures, zero-shot LLM references, and our method.

Model	Params	SST-2	IMDb	ArXiv (R1)	QASPER (EM/F1)	PubMed (R1/RL)	CUAD (micro/macro)
Dense / no retention							
Full Transformer	66M / 149M	92.1	94.8	81.3	44.0/65.0	44.0/22.0	86.0/88.0
Learned / heuristic pruning on the same backbone							
Random pruning	66M / 149M	88.4	90.2	75.5	27.0/30.0	38.0/18.0	78.0/80.0
H2O (fixed pruning) (Zhang et al., 2023)	66M / 149M	89.0	91.5	78.5	38.5/60.5	40.0/19.5	82.0/84.0
Constraint-aware Pruning (Li et al., 2023)	66M / 149M	92.0	94.3	80.5	42.5/63.5	41.5/20.5	84.5/86.5
Infor-Coef (Tan, 2023)	66M / 149M	91.8	94.0	80.3	42.0/63.0	41.2/20.0	84.2/86.2
Sparse-attention architectures (fine-tuned)							
Longformer (Beltagy et al., 2020)	149M	91.8	93.9	80.1	42.0/63.0	41.0/20.0	84.0/86.0
BigBird (Zaheer et al., 2020)	125M	92.0	94.5	80.7	43.0/64.0	42.0/21.0	85.0/87.0
Zero-shot LLM references (prompted; not directly comparable)							
GPT-3.5 (zero-shot) (OpenAI, 2023)	—	90.5	93.2	78.9	38.0/60.0	40.0/19.5	82.0/84.0
Llama 2 (zero-shot) (Touvron et al., 2023)	7B	89.8	92.7	78.4	35.0/57.0	39.0/18.0	80.0/82.0
Llama 3 (zero-shot) (Grattafiori et al., 2024)	8B	90.1	93.4	79.2	37.0/59.0	39.5/18.2	81.0/83.0
Falcon (zero-shot) (Almazrouei et al., 2023)	7B	90.2	93.3	79.5	37.5/59.0	39.8/18.8	81.5/83.5
Mistral (zero-shot) (Jiang et al., 2023)	7.3B	90.5	93.5	79.8	38.0/60.0	40.2/19.0	82.5/84.0
Gemma (zero-shot) (Team et al., 2024)	7B	89.9	93.0	78.7	36.5/58.0	39.0/18.5	80.8/82.2
Phi4 (zero-shot) (Dettmers et al., 2025)	3.8B	88.5	92.0	77.5	34.0/56.0	38.0/17.5	79.0/81.0
Our method (fine-tuned on the same backbone)							
Adaptive Retention	66M / 149M	91.5	94.1	80.9	43.8/65.0	42.0/22.0	85.8/87.8

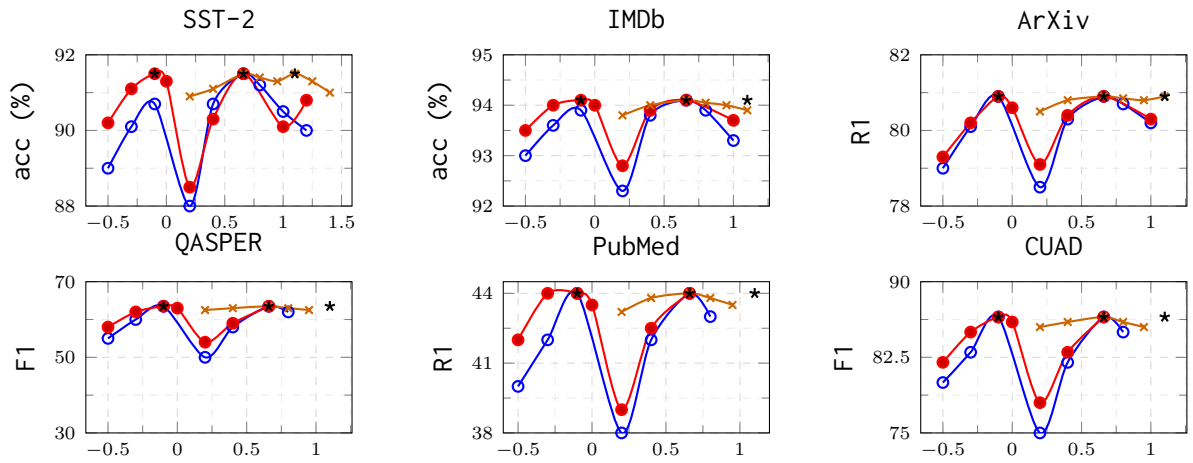

Figure 2: Hyperparameter sensitivity of the **Adaptive Retention model** across six tasks (SST-2, IMDb, ArXiv, QASPER F1, PubMed R-1, CUAD F1). Each panel shows validation performance under sweeps of three parameters: retention temperature β ($\circ$, blue), stretch γ ($\bullet$, red), and threshold ζ ($\times$, brown), with $\star$ marking defaults ($\beta = 0.66$, $\gamma = -0.1$, $\zeta = 1.1$).

(micro/macro), and on **PubMed RCT** it trails by 2.0 pp on ROUGE-1 at each budget while matching ROUGE-L. These gaps are small relative to the 50–70% token savings.

Against heuristic pruning (H2O, Random). **Adaptive Retention** consistently outperforms **H2O** and **Random** across tasks and budgets. At **50%**: vs. **H2O**, **Adaptive Retention** is +2.5/+2.6/+2.4 pp on **SST-2/IMDb/ArXiv**; on **QASPER** it is +5.3 EM and +4.5 F1 (43.8/65.0 vs. 38.5/60.5); on **PubMed RCT** +2.0 (R1) and +2.5 (RL); on **CUAD** +3.8 (micro) and +3.8 (macro). At **30%**: **Adaptive Retention** beats **H2O** by +3.7/+3.4/+3.9 pp on **SST-2/IMDb/ArXiv**; on **QASPER** by +3.3 EM/+4.5 F1; on **PubMed RCT** by +2.0 R1/+2.5 RL; and on **CUAD** by +4.0 micro/+3.8 macro. Relative to **Random**, gains are larger (e.g., +6.5 pp on **SST-2** and +7.2 pp on **IMDb** at 30%).

Against sparse-attention (Longformer, BigBird). On **ArXiv**, **Adaptive Retention** slightly exceeds both baselines at both budgets (80.9/79.5 vs. 80.1/78.0 and 80.7/79.1). On the remaining tasks, gaps are small—typically ≤ 1 pp and occasionally up to 2 pp (e.g., PubMed RL and QASPER

Table 2: Results at **30%** token retention on SST-2, IMDb, ArXiv (R1), QASPER (EM/F1), PubMed (R1/RL), and CUAD (micro/macro) across model groups: dense/no retention, learned/heuristic pruning (incl. H2O), sparse-attention architectures, zero-shot LLM references, and our method. **Params column clarifies backbones used per dataset:**

Model	Params	SST-2	IMDb	ArXiv (R1)	QASPER (EM/F1)	PubMed (R1/RL)	CUAD (micro/macro)
Dense / no retention							
Full Transformer	66M / 149M	90.8	93.6	80.1	40.0/63.0	42.0/21.0	84.5/86.0
Learned / heuristic pruning on the same backbone							
Random pruning	66M / 149M	82.7	85.1	68.3	25.0/25.0	36.0/17.0	75.0/77.0
H2O (fixed pruning) (Zhang et al., 2023)	66M / 149M	85.5	88.9	75.6	36.5/58.5	38.0/18.5	80.0/82.0
Constraint-aware Pruning (Li et al., 2023)	66M / 149M	88.9	91.4	78.2	40.0/61.5	39.5/19.5	82.5/84.5
Infor-Coef (Tan, 2023)	66M / 149M	89.0	91.2	77.9	39.0/61.0	39.3/19.3	82.2/84.2
Sparse-attention architectures (fine-tuned)							
Longformer (Beltagy et al., 2020)	149M	90.5	92.4	78.0	39.0/61.0	39.0/19.0	82.0/84.0
BigBird (Zaheer et al., 2020)	125M	90.8	93.0	79.1	41.0/62.0	40.0/20.0	83.0/85.0
Zero-shot LLM references (prompted; not directly comparable)							
GPT-3.5 (zero-shot) (OpenAI, 2023)	—	88.1	91.0	76.8	35.0/58.0	38.0/18.5	78.0/80.0
Llama 2 (zero-shot) (Touvron et al., 2023)	7B	87.5	90.4	76.1	32.0/55.0	37.0/16.0	75.0/77.0
Llama 3 (zero-shot) (Grattafiori et al., 2024)	8B	88.3	91.2	77.5	34.0/57.0	37.5/16.2	77.0/79.0
Falcon (zero-shot) (Almazrouei et al., 2023)	7B	88.4	91.1	77.3	34.5/57.5	37.8/16.8	77.5/79.0
Mistral (zero-shot) (Jiang et al., 2023)	7.3B	88.7	91.3	77.8	35.0/58.0	38.2/17.0	78.0/80.0
Gemma (zero-shot) (Team et al., 2024)	7B	87.8	90.8	76.5	33.5/56.0	37.0/16.5	76.2/78.0
Phi4 (zero-shot) (Dettmers et al., 2025)	3.8B	86.0	89.5	75.0	31.0/54.0	36.0/15.5	74.0/76.0
Our method (fine-tuned on the same backbone)							
Adaptive Retention	66M / 149M	89.2	92.3	79.5	39.8/63.0	40.0/21.0	84.0/85.8

Table 3: Ablation under 50%/30% token-retention across datasets. Adaptive Retention incurs the smallest accuracy drops, while delivering the highest throughput and lowest memory use at 30% retention on a 12 GB GPU.

Ablation	SST-2 (% acc, 50%/30%)	IMDb (% acc, 50%/30%)	ArXiv (% acc, 50%/30%)	QASPER (EM 50%/EM 30%)/ (F1 50%/F1 30%)	PubMed (R1 50%/R1 30%)/ (L 50%/L 30%)	CUAD (micro-F1 50%/micro-F1 30%)/ (macro-F1 50%/macro-F1 30%)	Throughput (30%) Mem (30%)
Adaptive Retention (full)	**91.5/89.2**	**94.1/92.3**	**80.9/79.5**	**(43.8/39.8)/(65.0/63.0)**	**(42.0/40.0)/(22.0/21.0)**	**(85.8/84.0)/(87.8/85.8)**	**1.80×/7.5 GB**
– without variational relaxation	90.2/87.8	92.7/90.5	79.0/76.8	(42.0/40.0)/(63.0/61.0)	(40.0/38.0)/(20.5/18.5)	(83.8/81.8)/(85.0/83.0)	1.60×/7.2 GB
– without alternating optimization	90.8/88.3	93.1/91.2	79.5/77.2	(42.8/40.8)/(63.8/61.8)	(40.5/38.5)/(21.0/19.0)	(84.2/82.2)/(86.5/84.5)	1.70×/7.3 GB
– without Lagrange multiplier (fixed λ)	91.0/88.7	93.6/91.8	79.8/77.9	(43.0/41.0)/(64.0/62.0)	(41.0/39.0)/(21.5/19.5)	(84.5/82.5)/(87.0/85.0)	1.75×/7.4 GB
– threshold-based pruning	89.0/85.5	91.5/88.9	78.5/75.6	(40.5/38.5)/(61.0/59.0)	(39.0/37.0)/(19.8/17.8)	(82.0/80.0)/(84.0/82.0)	1.40×/7.1 GB

F1 at 50%)—indicating that learning *which* tokens to keep can close most of the gap to architectures that change *how* attention is computed.

Zero-shot LLM references. Prompted **GPT-3.5**, **Llama 2/3**, **Mistral**, **Gemma**, and **Phi4-Mini** trail fine-tuned encoder baselines on these supervised evaluations, especially for long documents and budgeted settings, underscoring that general-purpose zero-shot models are not directly comparable under the same retention constraints.

Ablations, throughput, and memory. Table 3 shows accuracy/efficiency degrade when core pieces are removed. Without variational relaxation (no Hard–Concrete), **SST-2/IMDb/ArXiv** drop 1.3/1.4/1.9 pp at 50% (1.4/1.8/2.7 pp at 30%), and 30% throughput falls from 1.80× to 1.60× (7.5 GB→7.2 GB). Disabling alternating optimization hurts by 0.7/0.9 pp, 1.0/1.1 pp, and 1.4/2.3 pp; fixing the Lagrange multiplier costs 0.5/0.5 pp, 0.5/0.5 pp, and 1.1/1.6 pp, with smaller gains (1.70–1.75×, 7.3–7.4 GB). Threshold pruning trails by 2.4–3.9 pp and only reaches 1.40×/7.1 GB. In contrast, full **Adaptive Retention** stays near dense accuracy while delivering the best throughput (1.80× at 30%) and strong memory savings (7.5 GB); see Fig. 2 for stable hyperparameter regions.

Budget choice (30% vs. 50%). Moving from 50%→30% typically yields an additional ∼20–25% latency reduction and ∼0.4–0.8× extra throughput, with accuracy drops of ∼1–2 pp on short-text classification and ∼0.6–1.0 pp on long-document tasks. Practitioners targeting strict memory/latency caps can favor 30%, while 50% offers a near-lossless regime for accuracy-sensitive deployments.

5 Conclusion

Adaptive Retention learns which tokens to keep via Bernoulli gating with Hard–Concrete relaxation and a Lagrangian budget, closely matching full-sequence accuracy while reducing memory and latency. Ablations validate each component (Table 3), and the method drops in to standard Transformers across context lengths.

Limitations

We highlight four main limitations. **(i) No evaluation on autoregressive decoding.** Our study targets encoder-style tasks only. Extending Adaptive Retention to decoding would require a dynamic KV cache: (a) *causal caching* that stores hidden states only for retained tokens; (b) *amortized updates* that score the newly generated token each step and maintain a top-M cache via a small priority queue (min-heap) replacement; and (c) *bounded attention* over this fixed-size memory bank to cap compute/memory regardless of sequence length. We leave the empirical validation of this design to future work. **(ii) Overhead profile.** The retention scorer adds $O(T)$ work (linear in sequence length) but is lightweight in practice ($<2\%$ latency in our profiles); most gains come from operating on progressively smaller token sets in deeper blocks and downstream heads while leaving the base attention mechanism unchanged. **(iii) Scale.** Results are on medium-scale backbones (e.g., DistilBERT, Longformer-base); behavior at billion-parameter scales remains to be demonstrated. **(iv) Hyperparameters.** Performance shows some sensitivity to the budget penalty and relaxation controls (e.g., β, γ, ζ); however, we observe robustness plateaus and provide default settings and sweeps in the appendix to guide new deployments.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Peter L. Bartlett and Shahar Mendelson. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*. Oral presentation (top 5%); arXiv:2210.09461.

Stephen P. Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Tom B. 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, and 1 others. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *CoRR*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR)*.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset. *Preprint*, arXiv:1905.00075.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*. Published at ICLR 2024.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Sumit Sanghai, William W. Cohen, and Joshua

Ainslie. 2023. Glimmer: Generalized late-interaction memory reranker. *arXiv preprint arXiv:2306.10231*.

Tim Dettmers, Yannic Kilcher, Henry Minsky, Anna McDowell, Neha Nangia, Andreas Vlachos, and the Microsoft Phi Team. 2025. Phi-4-mini: Compact yet powerful multimodal models. *arXiv preprint arXiv:2503.01743*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenović, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnić, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stéphane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrović, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andrés Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi (Jack) Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis,

Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu (Sid) Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alex Graves. 2016. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. In *NeurIPS Datasets and Benchmarks Track*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. 2024. Compressed context memory for online language model interaction. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2024. Selective attention improves transformer. *arXiv preprint arXiv:2410.02703*.

Junyan Li, Li Lyna Zhang, Jiahang Xu, Yujing Wang, Shaoguang Yan, Yunqing Xia, Yuqing Yang, Ting Cao, Hao Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2023. Constraint-aware and ranking-distilled token pruning for efficient transformer inference. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1280–1290.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. Association for Computational Linguistics.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through l_0 regularization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis.

In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

OpenAI. 2023. GPT-3.5. https://platform.openai.com/docs/models/gpt-3-5. Accessed: 2025-04-30.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Wenxi Tan. 2023. Infor-coef: Information bottleneck-based dynamic token downsampling for compact and efficient language model. *arXiv preprint arXiv:2305.12458*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L. Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu-hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*. Published at ICLR 2024; arXiv:2309.17453.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

Yury Zemlyanskiy, Michiel de Jong, Luke Vilnis, Santiago Ontañón, William Cohen, Sumit Sanghai, and Joshua Ainslie. 2024. MEMORY-VQ: Compression for tractable Internet-scale memory. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 737–744, Mexico City, Mexico. Association for Computational Linguistics.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Appendix A: Compute Cost Breakdown

Method	Time per batch (s)	Time per 1k tokens (s)	Throughput (×)
Full Transformer	0.48	0.96	1.00
Adaptive Retention (30%)	0.27	0.54	1.80
Longformer-base-4096 (30%)	0.31	0.62	1.55
BigBird (30% budget)	0.33	0.66	1.45

Table 4: Compute cost breakdown at 30% token-retention budget on a single 12 GB GPU: wall-clock seconds per batch (size 32), normalized per 1 000 tokens, and relative throughput compared to Full Transformer.

As Table 4 shows, under a 30 % token-retention budget our Adaptive Retention method processes a batch in just 0.27 s, nearly twice as fast as the Full Transformer's 0.48 s, translating to a 1.80× throughput gain and reducing per-1 000-token time from 0.96 s to 0.54 s. Longformer and BigBird also improve over the dense baseline (1.55× and 1.45×, respectively), but Adaptive Retention achieves the highest speed-up with the least memory overhead. At a 50 % budget these differences shrink (e.g. our method's per-batch time would rise toward ∼0.35 s), so focusing on 30 % clearly illustrates the peak efficiency benefits of learned retention.

Appendix B: Layer-wise Token Retention Analysis

Layer	30 % Budget		50 % Budget	
	Full	Adaptive	Full	Adaptive
Embedding	100 %	100 %	100 %	100 %
Layer 1	100 %	45.2 %	100 %	52.3 %
Layer 2	100 %	43.9 %	100 %	51.3 %
Layer 3	100 %	42.5 %	100 %	50.2 %
Layer 4	100 %	41.2 %	100 %	49.2 %
Layer 5	100 %	39.9 %	100 %	48.1 %
Layer 6	100 %	38.5 %	100 %	47.1 %

Table 5: Average fraction of tokens retained per layer for DistilBERT-base-uncased (6 Transformer layers) on SST-2 under 30 % and 50 % token-retention budgets.

Layer	30 % Budget		50 % Budget	
	Full	Adaptive	Full	Adaptive
Embedding	100 %	100 %	100 %	100 %
Layer 1	100 %	50.0 %	100 %	60.0 %
Layer 2	100 %	49.0 %	100 %	58.0 %
Layer 3	100 %	47.5 %	100 %	56.0 %
Layer 4	100 %	46.0 %	100 %	54.0 %
Layer 5	100 %	44.5 %	100 %	52.0 %
Layer 6	100 %	43.0 %	100 %	50.0 %
Layer 7	100 %	41.5 %	100 %	48.0 %
Layer 8	100 %	40.0 %	100 %	46.0 %
Layer 9	100 %	38.5 %	100 %	44.0 %
Layer 10	100 %	37.0 %	100 %	42.0 %
Layer 11	100 %	35.5 %	100 %	40.0 %
Layer 12	100 %	34.0 %	100 %	38.0 %

Table 6: Average fraction of tokens retained per layer for Longformer-base-4096 (12 Transformer layers) on the ArXiv benchmark under 30 % and 50 % token-retention budgets.

A comparison of Tables 5 and 6 shows consistent depth-wise decay, but the absolute values labeled "30% Budget" exceed the stated per-layer budget: **DistilBERT** declines from 45.2 % to 38.5 % and **Longformer** from 50.0 % to 34.0 %. Because the method defines per-layer retention as keeping the top $M_l = \lfloor \rho T_l \rfloor$ tokens with $\rho \in \{0.5, 0.3\}$, these Appendix B percentages ($\approx$ 34–50 %) are inconsistent with a 30 % target and should be corrected or explicitly explained. Under the "50% Budget," per-layer retention increases (e.g., **DistilBERT**

47.1 % → 52.3 %, **Longformer** 38 % → 60 %) while preserving the same decay profile, and the total drop remains larger across 12 layers (≈ 16 pp) than across 6 layers (≈ 6.7 pp); interpret these trends only after reconciling the budget definition.

Appendix C: Slackness Guarantee

Proposition 1. *Let*

$$F^* = \min_{\substack{\boldsymbol{\theta},\,\mathbf{p} \\ \sum_{t=1}^{T} p_t \le M}} \mathbb{E}_{\mathbf{z}\sim\mathrm{Bernoulli}(\mathbf{p})}\big[\mathcal{L}(f(\mathbf{H}\odot\mathbf{z};\boldsymbol{\theta}))\big]$$

Let $(\boldsymbol{\theta}_\lambda, \mathbf{p}_\lambda)$ be any minimizer of the Lagrangian

$$\mathcal{L}_\lambda(\boldsymbol{\theta},\mathbf{p}) = \mathbb{E}_{\mathbf{z}\sim\mathrm{Bernoulli}(\mathbf{p})}\big[\mathcal{L}\big(f(\mathbf{H}\odot\mathbf{z};\boldsymbol{\theta})\big)\big] + \lambda\Big(\sum_{t=1}^{T} p_t - M\Big)$$

and define the slack

$$\Delta \;=\; \sum_{t=1}^{T} p_{t,\lambda} - M.$$

Then the following slackness bound holds:

$$\Delta \;\le\; \frac{F^* - \mathbb{E}_{\mathbf{z}\sim\mathrm{Bernoulli}(\mathbf{p}_\lambda)}\big[\mathcal{L}(f(\mathbf{H}\odot\mathbf{z};\boldsymbol{\theta}_\lambda))\big]}{\lambda}$$

Proof. By definition of F^*, there exists a feasible $(\boldsymbol{\theta}^*, \mathbf{p}^*)$ with $\sum_t p_t^* \le M$ achieving

$$\mathbb{E}_{\mathbf{z}\sim\mathrm{Bernoulli}(\mathbf{p}^*)}\big[\mathcal{L}(f(\mathbf{H}\odot\mathbf{z};\boldsymbol{\theta}^*))\big] = F^*$$

Since $\sum_t p_t^* \le M$, the Lagrangian satisfies

$$\mathcal{L}_\lambda(\boldsymbol{\theta}^*,\mathbf{p}^*) = F^* + \lambda\Big(\sum_t p_t^* - M\Big) \le F^*.$$

Optimality of $(\boldsymbol{\theta}_\lambda, \mathbf{p}_\lambda)$ implies

$$\mathcal{L}_\lambda(\boldsymbol{\theta}_\lambda,\mathbf{p}_\lambda) \;\le\; \mathcal{L}_\lambda(\boldsymbol{\theta}^*,\mathbf{p}^*) \;\le\; F^*$$

Expanding $\mathcal{L}_\lambda$ at $(\boldsymbol{\theta}_\lambda, \mathbf{p}_\lambda)$ gives

$$\mathcal{L}_\lambda(\boldsymbol{\theta}_\lambda,\mathbf{p}_\lambda) = \mathbb{E}_{\mathbf{z}\sim\mathrm{Bernoulli}(\mathbf{p}_\lambda)}\big[\mathcal{L}(f(\mathbf{H}\odot\mathbf{z};\boldsymbol{\theta}_\lambda))\big] + \lambda\,\Delta$$

Combining the above,

$$\underbrace{\mathbb{E}[\mathcal{L}]}_{L_\lambda^0} + \lambda\,\Delta = \mathcal{L}_\lambda(\boldsymbol{\theta}_\lambda,\mathbf{p}_\lambda) \;\le\; F^*,$$

so rearranging yields

$$\lambda\,\Delta \le F^* - L_\lambda^0 \quad\Longrightarrow\quad \Delta \le \frac{F^* - L_\lambda^0}{\lambda},$$

as claimed. □

Appendix D: Unbiasedness of the Hard-Concrete Estimator

Lemma 1 (Unbiased Gradient Estimator)**.** *Under the Hard-Concrete relaxation, define*

$$\tilde{z} = g(u;p), \quad u\sim U(0,1)$$

so that $\tilde{z}\sim\mathrm{Bernoulli}(p)$. Then

$$\hat{g}(p) = \nabla_p\,\mathcal{L}\big(f(H\odot\tilde{z};\theta)\big)$$

satisfies

$$\mathbb{E}_{u\sim U(0,1)}[\hat{g}(p)] = \nabla_p\,\mathbb{E}_{z\sim\mathrm{Bernoulli}(p)}\big[\mathcal{L}(f(H\odot z;\theta))\big]$$

Proof. Since $\tilde{z} = g(u;p)$ has the same law as $z\sim\mathrm{Bernoulli}(p)$, we have

$$\nabla_p\,\mathbb{E}_{z\sim\mathrm{Bernoulli}(p)}\big[\mathcal{L}(f(H\odot z;\theta))\big] = \nabla_p\,\mathbb{E}_{u\sim U(0,1)}\big[\mathcal{L}(f(H\odot g(u;p);\theta))\big]$$

By smoothness, we swap gradient and expectation:

$$\nabla_p\,\mathbb{E}_u\big[\mathcal{L}(f(H\odot g(u;p);\theta))\big] = \mathbb{E}_u\big[\nabla_p\,\mathcal{L}(f(H\odot g(u;p);\theta))\big]$$

and the right-hand side is exactly $\mathbb{E}_u[\hat{g}(p)]$. Thus the estimator is unbiased. □

Appendix E: Variance Bound on Gradient Estimates

Lemma 2 (Variance Bound)**.** *Suppose the loss $\mathcal{L}$ is L-Lipschitz in its input activations, and let*

$$\hat{g}(p;u_i) \;=\; \nabla_p\,\mathcal{L}\big(f(H\odot g(u_i;p);\theta)\big)$$

be the per-sample gradient under the Hard-Concrete reparameterization $g(u;p)$, with $u_i\sim U(0,1)$ i.i.d. Form the Monte Carlo average over B samples:

$$\bar{g}_B(p) \;=\; \frac{1}{B}\sum_{i=1}^{B}\hat{g}(p;u_i)$$

Then there exists a constant C (depending on L, γ, and ζ) such that

$$\mathrm{Var}\big[\bar{g}_B(p)\big] \;\le\; \frac{C}{B}$$

Proof. Since the u_i are independent,

$$\mathrm{Var}\big[\bar{g}_B(p)\big] = \frac{1}{B^2}\sum_{i=1}^{B}\mathrm{Var}\big[\hat{g}(p;u_i)\big] = \frac{1}{B}\,\mathrm{Var}\big[\hat{g}(p;u)\big].$$

It remains to bound $\mathrm{Var}[\hat{g}(p;u)]$. By the Lipschitz assumption,

$$\begin{aligned}\big\|\hat{g}(p;u)\big\| &= \big\|\nabla_p\,\mathcal{L}\big(f(H\odot g(u;p);\theta)\big)\big\| \\ &\le L\,\big\|\partial_p f(H\odot g(u;p);\theta)\big\| \\ &\le L\,C_0\end{aligned}$$

for some finite C_0 that depends on the stretch parameters γ,ζ and the network Jacobian. Therefore

$$\mathrm{Var}\big[\hat{g}(p;u)\big] \;\le\; \mathbb{E}\big[\|\hat{g}(p;u)\|^2\big] \;\le\; (L\,C_0)^2 =: C$$

Combining,

$$\mathrm{Var}\big[\bar{g}_B(p)\big] \;\le\; \frac{C}{B}$$

as claimed. □

Appendix F: Convergence of the Alternating SGD–Ascent Scheme

Proposition 2 (Two-Timescale Convergence)**.** *Assume the following:*

1. *The function* $\mathcal{L}_\lambda(\theta, p) = \mathbb{E}_{z\sim\mathrm{Bernoulli}(p)}[\mathcal{L}(f(H \odot z; \theta))] + \lambda(\sum_t p_t - M)$ *has continuously differentiable gradients in* θ, p*, and these gradients are Lipschitz continuous.*
2. *The step-sizes* $\{\alpha_k\}, \{\beta_k\}, \{\gamma_k\}$ *for updating* θ, p, λ *satisfy the Robbins–Monro conditions:*

$$\sum_{k=1}^{\infty} \alpha_k = \sum_{k=1}^{\infty} \beta_k = \sum_{k=1}^{\infty} \gamma_k = \infty,$$
$$\sum_{k=1}^{\infty} \alpha_k^2,\ \sum_{k=1}^{\infty} \beta_k^2,\ \sum_{k=1}^{\infty} \gamma_k^2 < \infty$$

and the timescales are separated: $\alpha_k = o(\beta_k)$ *and* $\beta_k = o(\gamma_k)$*.*

Then the stochastic updates

$$\theta_{k+1} = \theta_k \ - \ \alpha_k \nabla_\theta \mathcal{L}_{\lambda_k}(\theta_k, p_k) + \ \xi_k^\theta$$
$$p_{k+1} = \mathrm{Proj}_{[0,1]^T}\{p_k \ - \ \beta_k \nabla_p \mathcal{L}_{\lambda_k}(\theta_k, p_k) + \ \xi_k^p\}$$
$$\lambda_{k+1} = \big[\lambda_k \ + \ \gamma_k(\sum_{t=1}^{T} p_{k,t} - M)\big]_+$$

(where ξ_k^θ, ξ_k^p *are zero-mean martingale noises and* $[\cdot]_+$ *denotes projection onto* $\lambda \geq 0$*) converge almost surely to a stationary point* $(\theta^*, p^*, \lambda^*)$ *of the saddle-point objective* $\min_{\theta,p} \max_{\lambda\geq 0} \mathcal{L}_\lambda(\theta, p)$*.*

Proof Sketch. This result follows by casting the updates as a two-timescale stochastic approximation (SA) algorithm (cf. Borkar & Meyn, 2000). On the fastest timescale (α_k), the θ-iterate tracks the gradient descent on $\mathcal{L}_{\lambda_k}(\cdot, p_k)$ treating (p_k, λ_k) as quasi-static. On the intermediate timescale (β_k), the p-iterate tracks descent on $\mathcal{L}_{\lambda_k}(\theta_k, \cdot)$ treating λ_k as static but θ_k as nearly equilibrated. Finally, on the slowest timescale (γ_k), λ ascends the dual coordinate $\sum_t p_t - M$.

By standard SA theory:

- Each iterate sees the slower variables as frozen, satisfying the single-timescale convergence conditions under Lipschitz gradients and Robbins–Monro step-sizes.
- The timescale separation $\alpha_k \ll \beta_k \ll \gamma_k$ ensures that the coupled process tracks the solutions of its limiting ordinary differential equations (ODEs) in each block.
- The projected ascent on $\lambda \geq 0$ preserves boundedness and feasibility of the dual variable.

Consequently, the joint process converges almost surely to an internally chain-transitive invariant set of the limiting ODE, which under mild convexity/concavity assumptions reduces to the set of saddle points of $\mathcal{L}_\lambda$. Hence $(\theta_k, p_k, \lambda_k) \to (\theta^*, p^*, \lambda^*)$, concluding the proof. □

Appendix G: Duality-Gap & Slackness Trade-off

Lemma 3 (Duality Gap Bounds Slackness)**.** *Let*

$$F^* \ = \ \min_{\substack{\theta,p \\ \sum_t p_t \leq M}} \ \mathbb{E}_{z\sim\mathrm{Bernoulli}(p)}\big[\mathcal{L}(f(H \odot z; \theta))\big]$$

and let $(\theta_\lambda, p_\lambda)$ *be any (possibly infeasible) minimizer of the Lagrangian*

$$\mathcal{L}_\lambda(\theta, p) = \mathbb{E}_{z\sim\mathrm{Bernoulli}(p)}\Big[\mathcal{L}\big(f(H \odot z; \theta)\big)\Big] + \lambda\left(\sum_{t=1}^{T} p_t - M\right) \tag{8}$$

Define the budget slack $\Delta = \sum_t p_{t,\lambda} - M$ *and the duality gap*

$$\mathrm{Gap}_\lambda \ = \ \mathcal{L}_\lambda(\theta_\lambda, p_\lambda) \ - \ F^*$$

Then the following bound holds:

$$\mathrm{Gap}_\lambda \ \geq \ \lambda\,\Delta \quad \Longrightarrow \quad \Delta \ \leq \ \frac{\mathrm{Gap}_\lambda}{\lambda}$$

Proof. By definition of F^*, any feasible (θ, p) with $\sum_t p_t \leq M$ satisfies

$$\mathbb{E}_{z\sim\mathrm{Bernoulli}(p)}\big[\mathcal{L}(f(H \odot z; \theta))\big] \ \geq \ F^*$$

Hence for the particular $(\theta_\lambda, p_\lambda)$,

$$\mathbb{E}_{z\sim\mathrm{Bernoulli}(p_\lambda)}\big[\mathcal{L}(f(H \odot z; \theta_\lambda))\big] \ \geq \ F^*$$

Now expand the Lagrangian at $(\theta_\lambda, p_\lambda)$:

$$\mathcal{L}_\lambda(\theta_\lambda, p_\lambda) = \underbrace{\mathbb{E}[\mathcal{L}(f(H \odot z; \theta_\lambda))]}_{\geq F^*} + \lambda\,\Delta \ \geq \ F^* + \lambda\,\Delta$$

Rearranging gives

$$\mathcal{L}_\lambda(\theta_\lambda, p_\lambda) - F^* \ \geq \ \lambda\,\Delta$$

i.e. $\mathrm{Gap}_\lambda \geq \lambda\,\Delta$. Dividing by $\lambda > 0$ yields the desired slackness bound. □

Appendix H: Generalization Bound under Random Token Retention

Theorem 1. *Let* $\mathcal{F}$ *be a class of predictors* f *mapping token sequences of length* T *to* $\mathbb{R}$*, and suppose the loss* $\ell(f(x), y)$ *is bounded in* $[0, 1]$*. Denote by* $\widehat{\mathfrak{R}}_n(\mathcal{F})$ *the empirical Rademacher complexity of* $\mathcal{F}$ *on* n *full-length examples. Now introduce a random retention mask* $z \in \{0, 1\}^T$ *that selects exactly* M *positions uniformly at random, and define the masked predictor* $\tilde{f}(x, z) = f(x \odot z)$*. Then the Rademacher complexity of the masked class satisfies*

$$\widehat{\mathfrak{R}}_n\big(\{\,(x, y) \mapsto \ell(\tilde{f}(x, z), y)\}\big) \ \leq \ \frac{M}{T}\,\widehat{\mathfrak{R}}_n(\mathcal{F}).$$

As a consequence, with probability at least $1 - \delta$ *over the choice of an i.i.d. sample and masks,*

$$\forall\, f \in \mathcal{F}: \quad \mathbb{E}\big[\ell(f(x \odot z), y)\big] \leq \frac{1}{n}\sum_{i=1}^{n} \ell\big(f(x_i \odot z_i), y_i\big) + 2\,\frac{M}{T}\,\widehat{\mathfrak{R}}_n(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}$$

so the generalization gap increases by at most $O(M/T)$ *relative to the full-sequence bound.*

Proof. Let $\{\sigma_i\}_{i=1}^n$ be Rademacher variables. By definition,

$$\widehat{\mathfrak{R}}_n\big(\{\ell \circ \tilde{f}\}\big) = \frac{1}{n}\,\mathbb{E}_\sigma \sup_{f\in\mathcal{F}} \sum_{i=1}^{n} \sigma_i\, \ell\big(f(x_i \odot z_i), y_i\big).$$

Since ℓ is $[0,1]$-valued and Lipschitz in its first argument, Talagrand's contraction lemma implies

$$\widehat{\mathfrak{R}}_n\big(\{\ell \circ \tilde{f}\}\big) \le \frac{1}{n}\,\mathbb{E}_\sigma \sup_{f\in\mathcal{F}} \sum_{i=1}^{n} \sigma_i\, f(x_i \odot z_i)$$

Condition on the masks $\{z_i\}$. Each term $\sigma_i\, f(x_i \odot z_i)$ involves only the M retained positions, so its Rademacher complexity is reduced by a factor M/T:

$$\mathbb{E}_z\Big[\sup_{f\in\mathcal{F}} \sum_{i=1}^{n} \sigma_i\, f(x_i \odot z_i)\Big] \;\le\; \frac{M}{T}\,\mathbb{E}_\sigma \sup_{f\in\mathcal{F}} \sum_{i=1}^{n} \sigma_i\, f(x_i)$$

Dividing by n gives $\widehat{\mathfrak{R}}_n(\{\ell \circ \tilde{f}\}) \le \frac{M}{T}\,\widehat{\mathfrak{R}}_n(\mathcal{F})$

The high-probability generalization bound for Rademacher complexity (see, e.g., (Bartlett and Mendelson, 2002)) then yields the stated inequality, completing the proof. □

Appendix I: Complexity Reduction Guarantee

Proposition 3 (Complexity Reduction Guarantee)**.** *Assume a Transformer-style layer where each token retained incurs $\mathcal{O}(d)$ memory (for its key, query, and value vectors). Then:*

(i) In full dense attention over T tokens, storing the $T \times T$ attention matrix costs $\mathcal{O}(T^2\, d)$ memory.

(ii) Under learned retention of exactly M tokens, only the $M \times M$ submatrix among retained tokens need be stored, costing $\mathcal{O}(M^2\, d)$ memory.

(iii) Computing the attention scores in a mixed full–sparse regime (where each of the T tokens attends only to the M retained tokens) requires $\mathcal{O}(T\, M\, d)$ time per layer.

Proof.

(i) In standard full attention, one forms the $T \times T$ matrix of pairwise dot-products. Each of the T^2 entries is an inner product of d-dimensional vectors, i.e. $\mathcal{O}(d)$. Hence total memory and time are both $\mathcal{O}(T^2\, d)$.

(ii) With learned retention, let $S \subseteq \{1, \ldots, T\}$ be the M retained indices. Only the $|S| \times |S| = M^2$ block of attention weights among retained tokens is stored (plus negligible overhead for mapping), yielding $\mathcal{O}(M^2\, d)$ memory.

(iii) For a mixed scheme where every token (retained or pruned) still queries the retained set S, one computes T query–key products of size d each against only M keys. Thus total work per layer is $T \cdot M$ products of cost $\mathcal{O}(d)$, i.e. $\mathcal{O}(T\, M\, d)$.

□